



UNIVERSIDAD CÉSAR VALLEJO

**ESCUELA DE POSGRADO
PROGRAMA ACADÉMICO DE MAESTRÍA EN INGENIERÍA
DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN**

ChatGPT desarrollando código fuente de software para historias de usuarios en una consultora, Lima, 2023

TESIS PARA OBTENER EL GRADO ACADÉMICO DE:

Maestro En Ingeniería De Sistemas Con Mención En Tecnologías De La Información.

AUTOR:

Altamirano Tseng, Leenkim Hahnemann (orcid.org/0000-0003-0036-2784)

ASESORES:

Mg. Eduardo Humberto, Poletti Gaitan (orcid.org/0000-0002-2143-4444)

Dr. Manuel Antonio, Pereyra Acosta (orcid.org/0000-0002-2593-5772)

LÍNEA DE INVESTIGACIÓN:

Tecnologías de la información y comunicación

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA – PERÚ
2024

DEDICATORIA

Con profundo cariño dedico esta investigación a mis padres y hermanos, por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida. A mi familia en general, porque me han brindado su apoyo incondicional.

AGRADECIMIENTO

Agradezco sinceramente a Dios por bendecirme para llegar hasta donde he llegado. Asimismo, agradecer a todos los docentes que brindaron sus conocimientos y su apoyo para seguir adelante día a día. Finalmente un eterno agradecimiento a esta prestigiosa casa de estudios por permitirme el uso de la biblioteca virtual que me ayudó en la elaboración de esta tesis.

DECLARATORIA DE AUTENTICIDAD DEL ASESOR



UNIVERSIDAD CÉSAR VALLEJO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaratoria de Autenticidad del Asesor

Yo, POLETTI GAITAN EDUARDO HUMBERTO, docente de la ESCUELA DE POSGRADO MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "ChatGPT desarrollando código fuente de software para historias de usuarios en una consultora, Lima, 2023", cuyo autor es ALTAMIRANO TSENG LEENKIM HAHNEMANN, constato que la investigación tiene un índice de similitud de 15.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 04 de Enero del 2024

Apellidos y Nombres del Asesor:	Firma
POLETTI GAITAN EDUARDO HUMBERTO DNI: 18073124 ORCID: 0000-0002-2143-4444	Firmado electrónicamente por: EPOLETTIG el 07-01- 2024 17:23:01

Código documento Trilce: TRI - 0719701



DECLARATORIA DE ORIGINALIDAD DEL AUTOR



ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaratoria de Originalidad del Autor

Yo, ALTAMIRANO TSENG LEENKIM HAHNEMANN estudiante de la ESCUELA DE POSGRADO del programa de MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, declaro bajo juramento que todos los datos e información que acompañan la Tesis titulada: "ChatGPT desarrollando código fuente de software para historias de usuarios en una consultora, Lima, 2023", es de mi autoría, por lo tanto, declaro que la Tesis:

1. No ha sido plagiada ni total, ni parcialmente.
2. He mencionado todas las fuentes empleadas, identificando correctamente toda cita textual o de paráfrasis proveniente de otras fuentes.
3. No ha sido publicada, ni presentada anteriormente para la obtención de otro grado académico o título profesional.
4. Los datos presentados en los resultados no han sido falseados, ni duplicados, ni copiados.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de la información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Nombres y Apellidos	Firma
ALTAMIRANO TSENG LEENKIM HAHNEMANN DNI: 73179995 ORCID: 0000-0003-0036-2784	Firmado electrónicamente por: LALTAMIRANOTS99 el 08-01-2024 09:00:39

Código documento Trilce: INV - 1457827



ÍNDICE DE CONTENIDOS

	Pág.
CARÁTULA	i
DEDICATORIA	ii
AGRADECIMIENTO	iii
DECLARATORIA DE AUTENTICIDAD DEL ASESOR	iv
DECLARATORIA DE ORIGINALIDAD DEL AUTOR	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	vii
ÍNDICE DE FIGURAS	viii
RESUMEN	ix
ABSTRACT	x
I. INTRODUCCIÓN	1
II. MARCO TEÓRICO	3
III. METODOLOGÍA	19
3.1. Tipo y diseño de investigación	19
3.2. Variables y operacionalización	19
3.3. Población, muestra y muestreo	20
3.4. Técnicas e instrumentos de recolección de datos	21
3.5. Procedimientos	24
3.6. Método de análisis de datos	25
3.7. Aspectos éticos	25
IV. RESULTADOS	27
V. DISCUSIÓN	34
VI. CONCLUSIONES	40
VII. RECOMENDACIONES	41
REFERENCIAS	42
ANEXOS	48

ÍNDICE DE TABLAS

Tabla 1 Estadística de fiabilidad de Alfa de Cronbach	22
Tabla 2 Prueba de normalidad para la Dimensión 1	23
Tabla 3 Prueba de normalidad para la Dimensión 2	23
Tabla 4 Prueba de normalidad para la Dimensión 3	24
Tabla 5 Dimensión 1: Complejidad	27
Tabla 6 Dimensión 2: Precisión	28
Tabla 7 Dimensión 3: Eficiencia	29
Tabla 8 Prueba de correlación de las variables	30
Tabla 9 Prueba de correlación de la dimensión 1 y la variable dependiente	31
Tabla 10 Prueba de correlación de la dimensión 2 y la variable dependiente	32
Tabla 11 Prueba de correlación de la dimensión 3 y la variable dependiente	33

ÍNDICE DE FIGURAS

Figura 1 Dimensión 1: Complejidad	27
Figura 2 Dimensión 2: Precisión	28
Figura 3 Dimensión 3: Eficiencia	29

RESUMEN

La presente investigación tuvo como objetivo determinar la relación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios en una consultora en Lima durante el 2023. Fue un estudio cuantitativo, no experimental transversal, descriptivo, correlacional. La muestra estuvo conformada por 124 historias de usuario desarrolladas con ChatGPT. Los datos se recolectaron a través de una ficha de análisis validada por juicio de expertos. Los resultados evidenciaron que ChatGPT tiene una relación significativo en el desarrollo de software. Asimismo, reduce la complejidad ($r=0.892$), aumenta la precisión ($r=0.813$) e incrementa la eficiencia ($r=0.776$). Se concluye que adoptar ChatGPT en consultoras de software locales potencia su transformación digital. Se recomienda configurar prompts especializados, capacitar al personal y complementar ChatGPT con otros recursos para garantizar un software de calidad.

Palabras clave: inteligencia artificial, desarrollo de software, consultoras tecnológicas.

ABSTRACT

The objective of this research was to determine the relationship of ChatGPT in the development of software source code for user stories in a consulting firm in Lima during 2023. It was a quantitative, non-experimental cross-sectional, descriptive, correlational study. The sample was made up of 124 user stories developed with ChatGPT. The data were collected through an analysis form validated by expert judgment. The results showed that ChatGPT has a significant relationship in software development. Likewise, it reduces complexity ($r=0.892$), increases precision ($r=0.813$) and increases efficiency ($r=0.776$). It is concluded that adopting ChatGPT in local software consultancies enhances their digital transformation. It is recommended to configure specialized prompts, train staff and complement ChatGPT with other resources to ensure quality software.

Keywords: artificial intelligence, software development, technology consultants.

I. INTRODUCCIÓN

En la era digital actual, la inteligencia artificial se ha convertido en una herramienta poderosa y versátil que influye en diversos aspectos de la vida cotidiana y, en particular, en el campo de la tecnología y el desarrollo de software. La evolución constante de la tecnología ha dado lugar a la creación de sistemas de procesamiento de lenguaje natural cada vez más sofisticados, como ChatGPT, que tienen la capacidad de comprender y generar texto de manera casi indistinguible de la producida por humanos. La influencia de la inteligencia artificial en la ingeniería de software y, específicamente, en el desarrollo de historias de usuarios, se ha convertido en un tema de creciente interés a nivel internacional y nacional. Por otro lado, a nivel internacional, diversas organizaciones y empresas líderes en tecnología han explorado el potencial de la inteligencia artificial, como ChatGPT, en el ámbito del desarrollo de software. La adopción de estas tecnologías impulsó la automatización de tareas relacionadas con la generación de código y la documentación de historias de usuarios. Se realizaron investigaciones que destacan cómo la inteligencia artificial puede acelerar la producción de código, mejorar la eficiencia y, en algunos casos, aumentar la precisión en la creación de código fuente. Estos avances han revolucionado la industria del software y han generado un debate sobre la redefinición de roles y responsabilidades.

A nivel nacional, en nuestro país, la adopción de tecnologías de inteligencia artificial en el desarrollo de software también experimentó un crecimiento constante. Consultoras y empresas de desarrollo de software comenzaron a explorar cómo ChatGPT y tecnologías similares pueden optimizar la generación de código y las historias de usuarios. La eficiencia y la precisión son aspectos críticos en la industria del software, y las organizaciones buscaron aprovechar estas herramientas para mantenerse competitivas en el mercado.

Adicionalmente; se planteó como problema general ¿ChatGPT se relaciona con el desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023? Asimismo; el primer problema específico fue ¿ChatGPT se relaciona con la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023? Agregando; como segundo problema específico fue ¿ChatGPT se relaciona con la precisión del desarrollo de

código fuente de software para historias de usuarios en una consultora, Lima, 2023? Finalmente; el tercer problema específico fue ¿ChatGPT se relaciona con la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023?

Por otro lado; la justificación práctica benefició a empresas consultoras y profesionales tecnológicos, optimizando prácticas y mejorando la eficiencia. Finalmente; la justificación teórica contribuyó al entendimiento de IA en desarrollo de código, enriqueciendo la investigación en el área en constante evolución.

Adicionalmente; el objetivo general fue determinar la relación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Asimismo; el primer objetivo específico fue determinar la relación de ChatGPT en la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Agregando; el segundo objetivo específico fue determinar la relación de ChatGPT en la precisión del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Finalmente; el tercer objetivo específico fue determinar la relación de ChatGPT en la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Por otro lado; la hipótesis general fue ChatGPT se relaciona significativamente con el desarrollo del código fuente de software para historias de usuarios en una consultora, Lima, 2023. Asimismo; la primera hipótesis específica fue ChatGPT se relaciona significativamente con la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Asimismo; la segunda hipótesis específica fue ChatGPT se relaciona significativamente con la precisión del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. Finalmente; la tercera hipótesis específica fue ChatGPT se relaciona significativamente con la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.

II. MARCO TEÓRICO

Internacionalmente; Bender et al. (2023) mencionaron el objetivo de realizar un estudio exhaustivo sobre la evolución y el estado actual en la generación automática de código fuente a partir de especificaciones escritas en lenguaje natural. Análisis descriptivo y exploratorio. La población de interés incluyó investigadores, desarrolladores y empresas involucradas con GPT-3 en la generación de código a partir de lenguaje natural. La muestra fueron los casos de éxito, los desafíos en la evaluación de estos modelos y las posibles aplicaciones prácticas en entornos empresariales. Los resultados revelaron avances significativos en la generación automática de código, destacando la capacidad de modelos como GPT-3 para crear texto con una calidad que se asemeja a la redacción humana. En conclusión, este estudio resaltó el progreso y las perspectivas futuras de la generación de código a partir de lenguaje natural.

Agregando; Castro (2023) tuvo como objetivo proporcionar herramientas para evaluar la calidad y estructura del código en repositorios de control de versiones. Adoptó un enfoque práctico y desarrollador, dirigido a solucionar la carencia de herramientas efectivas para un análisis del código. La población de interés fueron los desarrolladores y equipos involucrados en proyectos con repositorios de control de versiones. Los resultados obtenidos demostraron la capacidad de las extensiones para realizar análisis precisos del código fuente. Concluyendo en que se documentó el diseño, implementación y validación de las extensiones.

Del mismo modo; Erazo et al. (2023) tuvieron el objetivo de explorar y analizar la aplicación de la inteligencia artificial (IA) en la optimización de programas informáticos, investigando sus beneficios y riesgos potenciales. Se llevó a cabo una revisión exhaustiva de la literatura existente sobre este tema, incluyendo estudios y artículos académicos. La población comprendió investigadores, profesionales de la industria tecnológica y expertos en IA. La muestra fueron los métodos de aprendizaje automático y otras estrategias de IA. Concluyendo que la aplicación de la IA en la optimización de programas tiene el potencial de transformar la industria tecnológica, mejorando considerablemente el rendimiento y la eficiencia de los programas.

Asimismo; Guitérrez et al. (2023) buscaron explorar cómo la integración de ChatGPT en redacciones periodísticas sin requerir formación técnica extensa podría influir en el trabajo diario. Utilizando un enfoque mixto. Se involucró a 12 periodistas sin experiencia en IA. Los resultados destacaron la utilidad de ChatGPT para tareas repetitivas, generación de ideas y análisis de datos, pero se señalaron preocupaciones sobre posibles inexactitudes y falta de 'empatía' en el contenido generado. En conclusión, mientras se reconoció el potencial de ChatGPT para mejorar la eficiencia en las redacciones, se subrayó la necesidad de abordar las preocupaciones identificadas para su implementación efectiva.

Adicionalmente; Lagos (2023) mencionó el objetivo de desarrollar una aplicación destinada a asistir a estudiantes de STEM en programación. El diseño se enfocó en la creación de una herramienta que, mediante el reconocimiento de un diagrama de flujo manuscrito, pudiera generar código fuente en el lenguaje de programación seleccionado por el estudiante. La población fueron estudiantes principiantes en programación. La muestra incluyó usuarios que utilizaron la aplicación para convertir sus diagramas de flujo en código fuente. Los resultados mostraron que la aplicación funcionaba en cuatro pasos: la entrada del diagrama de flujo; la identificación de elementos y textos; la generación de pseudocódigo basado en la detección previa; la devolución del código fuente en el lenguaje seleccionado por el estudiante. En conclusión, se ofrece un recurso para apoyar el aprendizaje inicial de programación en estudiantes de disciplinas STEM.

Por otro lado; Leiton (2023) se enfocó en diseñar y evaluar un método que automatizara la transformación de la sintaxis de múltiples lenguajes de programación hacia un lenguaje universal con una sintaxis genérica. El objetivo fue simplificar el análisis de programas y superar las limitaciones de herramientas costosas y con soporte restringido. La investigación, de tipo exploratorio y aplicado. La muestra consistió en programas en distintos lenguajes analizados a través de este método. Los resultados obtenidos evidenciaron la eficacia del enfoque propuesto al permitir un análisis más eficiente y simplificado de programas escritos en diferentes lenguajes. Como conclusión, se destacó la viabilidad de emplear un único motor de análisis para programas multi-lenguaje, en lugar de depender de múltiples analizadores de código.

También; de acuerdo a Morcela (2023) indicó que su artículo se centró en una evaluación realizada con ChatGPT, un sistema de IA que se describe como un modelo de lenguaje. Teniendo como propósito evaluar la capacidad de ChatGPT para responder preguntas y generar ejercicios y evaluaciones en el ámbito de la ingeniería industrial. Las resultantes de la evaluación indican que ChatGPT es eficaz para abordar preguntas simples, pero no produce resultados satisfactorios en situaciones más complejas y ejemplos específicos. Además, se subrayan las preocupaciones éticas relacionadas con la creciente automatización y el uso generalizado de la IA en el entorno laboral.

Asimismo; Yagmour (2023) indicó que su objetivo fue analizar la representación del flujo de información digital en tres obras de literatura digital latinoamericana: "Unicode" de Michael Hurtado (2019), "Untitled Document" de Ciro Múseres (2005) y "Canción de Unicode" de Canek Zapata (2017). El enfoque de la investigación abordó un análisis cualitativo y semiótico de estas obras literarias digitales. La población fueron estas obras específicas, y la muestra se centró en los aspectos relacionados con la representación del flujo de información en el contexto digital. Los resultados obtenidos evidenciaron una fusión entre el lenguaje humano y la sintaxis de programación, provocando una reinterpretación perceptual del tiempo en el entorno digital, destacando la velocidad y verticalidad como aspectos fundamentales en la comprensión de esta nueva forma de representación literaria. En conclusión, este estudio subraya la manera en que estas obras digitales han transformado la percepción del tiempo y el flujo de información en el ámbito literario, explorando nuevas formas de expresión que desafían los límites tradicionales de la codificación y decodificación en el entorno digital.

Agregando; Echevarría et al. (2022) mencionaron que su objetivo de esta investigación fue evaluar la efectividad de diferentes métodos de enseñanza de idiomas en estudiantes de secundaria para determinar el más adecuado en términos de adquisición lingüística. Se llevó a cabo un estudio experimental de diseño cuasiexperimental con enfoque mixto, combinando métodos cualitativos y cuantitativos. La población objetivo estuvo conformada por estudiantes de tres escuelas secundarias públicas en áreas urbanas. La muestra consistió en 150 estudiantes seleccionados aleatoriamente y divididos en tres grupos: uno expuesto

a un método tradicional, otro a un enfoque basado en juegos y el tercero a una combinación de ambos métodos. Los resultados mostraron que el grupo expuesto al enfoque basado en juegos demostró un aumento significativo en la fluidez y comprensión del idioma en comparación con los otros grupos. Esta investigación concluye que la implementación de estrategias de enseñanza basadas en juegos es más efectiva para la adquisición de idiomas en estudiantes de secundaria, destacando su potencial para mejorar la habilidad lingüística en un entorno educativo.

Asimismo; Fritz et al. (2022) mencionaron que el objetivo fue presentar una código fuente simplificado para calcular el Coeficiente de Correlación de Envoltente (ECC) entre dos antenas, derivada de las expresiones del campo eléctrico en la región de campo lejano. El enfoque de investigación se centró en la eficiencia y rapidez del algoritmo resultante, comparándolo con herramientas de software comercial para demostrar su ventaja en términos de velocidad. Esta nueva representación ofreció una ecuación discreta aplicable después de obtener el campo eléctrico en coordenadas esféricas mediante simulaciones computacionales. La población de interés eran antenas con variados niveles de eficiencia de radiación. La muestra se obtuvo mediante comparaciones entre los resultados del algoritmo propuesto y los obtenidos a través de software comercial, evidenciando una alta convergencia en los valores, pero con tiempos de simulación considerablemente reducidos al emplear el método desarrollado en esta investigación. En conclusión, este artículo proporcionó una alternativa eficiente y precisa para calcular el ECC en antenas con diferentes eficiencias de radiación, destacando la superioridad en términos de rapidez sin comprometer la precisión de los resultados frente a métodos basados en parámetros S utilizados comúnmente.

Adicionalmente; Gómez et al. (2022) se enfocaron en describir un extenso tema colaborativo centrado en el desarrollo de software, utilizando herramientas web adaptadas. El objetivo era combinar la enseñanza de la creación de software con metodologías ágiles. La investigación incluyó a 475 estudiantes enfocados en metodologías ágiles y Node.JS. Los resultados revelaron una alta satisfacción tanto en el cierre de proyectos como en la percepción del aprendizaje y el término del curso. En conclusión, este enfoque intensivo facilitó la enseñanza del desarrollo

web a través de la colaboración en equipos, simulando un entorno real mediante el uso de plataformas de gestión de proyectos y un software de gestión del aprendizaje.

También; Güemes et al. (2022) se enfocaron en mejorar la detección de defectos en pruebas de software durante la ejecución del código. Se basó en un enfoque de automatización de pruebas, utilizando el modelo MTest.search. Se consideró un enfoque exploratorio con desarrolladores y expertos en pruebas de software como población de interés. Los resultados validaron la eficacia de las extensiones propuestas al mejorar la detección de fallos, especialmente al evaluar valores y escenarios significativos en las pruebas. En conclusión, se resaltó la utilidad de estas extensiones para detectar tempranamente defectos en el proceso de desarrollo de software, mejorando la calidad y eficiencia en las pruebas durante la ejecución del código.

Agregando; Pérez et al. (2022) mencionaron como objetivo diseñar y evaluar un sistema domótico modular, específicamente destinado a pequeñas aplicaciones, como la automatización de tareas domésticas simples en apartamentos residenciales, con el fin de superar las limitaciones percibidas en sistemas previos en cuanto a funcionalidad, precio, complejidad y rendimiento. Enfoque experimental, utilizando componentes de hardware de código abierto y bajo costo, centrándose en el microcontrolador ESP8266 para aprovechar sus capacidades de conectividad inalámbrica y crear un sistema descentralizado. La población fueron usuarios finales en apartamentos y oficinas pequeñas que buscaban soluciones domóticas más accesibles y funcionales. La muestra estuvo compuesta por módulos independientes. Los resultados demostraron que este enfoque basado en hardware de código abierto y componentes económicos logró ofrecer un sistema domótico adaptable, con capacidad de interacción a través de una interfaz web y conexión Wi-Fi. En conclusión, este trabajo subraya la viabilidad y eficacia de sistemas domóticos modulares y asequibles, proporcionando una solución más adaptada y funcional para la automatización en entornos residenciales pequeños.

Asimismo; Puga (2022) tuvo como objetivo resolver el desafío de transmitir información contextual del código a programadores ciegos, al carecer de estándares claros en comparación con los métodos visuales utilizados por

programadores videntes para explorar y comprender el código. El enfoque fue conceptual de casos de uso para señales de código fuente y guías de diseño, apoyado en literatura y estudios empíricos. La población objetivo fueron los programadores ciegos, y la muestra incluyó tanto a programadores con diferentes niveles de experiencia como a sujetos videntes. Los resultados del modelo de casos de uso revelaron la dependencia del impacto de las señales visuales de código según el contexto de uso. Se desarrolló una extensión para Visual Studio Code que genera señales no-visuales, como audio neutro, texto hablado y vibraciones, al seleccionar palabras clave del lenguaje Python. En la conclusión, se resumió el trabajo realizado y se plantearon direcciones futuras para la investigación en este campo.

Del mismo modo; Gutiérrez (2021) indicó que su objetivo fue garantizar la transparencia en el uso de sistemas de IA por las Administraciones Públicas. La investigación tuvo un enfoque comparativo y exploratorio, analizando el derecho de acceso a la información y sus límites legales en sistemas algorítmicos. Se concluyó que el significado técnico de transparencia algorítmica difiere del significado jurídico de transparencia administrativa. Se recomendó que la legislación de transparencia defina información relevante y obligaciones de documentación técnica para facilitar el escrutinio público de sistemas de IA.

También; Hidalgo (2021) su artículo tuvo como objetivo proponer una estrategia para mejorar las habilidades de programación y el rendimiento académico en el curso CS1. La metodología integró la colaboración y la evaluación automática de código fuente para investigar el tiempo reducido en la resolución de tareas y el incremento en las calificaciones. Los resultados revelaron un aumento del 50% en las calificaciones y una mejora en las habilidades de programación, permitiendo resolver tareas en menos tiempo. En conclusión, estrategias que combinan colaboración y evaluación automática tienen un impacto positivo en el aprendizaje de programación, mejorando calificaciones y fomentando habilidades interpersonales en los estudiantes.

Adicionalmente; Leitón (2021) indicó que su objetivo de investigación fue diseñar un método que tradujera código Java, C# y RPG a un lenguaje universal, permitiendo su verificación automática para asegurar la coherencia sintáctica. La

metodología se enfocó en la obtención y mapeo de datos desde los archivos de código hacia una estructura genérica, presentada en formato JSON del árbol de sintaxis abstracta. Se detallaron los diagramas BNF de las principales sentencias y se validó el método con proyectos en dichos lenguajes, confirmando su funcionamiento para la traducción y verificación de elementos sintácticos. En conclusión, el método desarrollado demostró su efectividad al traducir lenguajes específicos a uno universal y validar la coherencia sintáctica de manera automatizada.

Asimismo; Peña et al. (2021) mencionaon como objetivo proponer el uso de Yworks como una herramienta aplicable para los desarrolladores de aplicaciones HPC. El diseño fue analizar herramientas de este tipo, utilizando un HPC estándar como punto de referencia para validar el concepto. La población de interés eran los desarrolladores de aplicaciones HPC, y la muestra incluyó aquellos que podrían potencialmente adoptar esta herramienta para mejorar la estructura y calidad del software. Los resultados mostraron que Yworks podría ofrecer métricas y análisis de calidad de código de manera intuitiva e interactiva, proporcionando retroalimentación valiosa a los desarrolladores. En conclusión, esta investigación resaltó la viabilidad de esta herramienta como un recurso útil para mejorar la funcionalidad y calidad del software en el contexto de aplicaciones HPC, ofreciendo una visión integral y práctica para el desarrollo de software en este campo.

Agregando; Tamayo et al. (2021) indicaron como objetivo mejorar la calidad del software a través de una gestión efectiva de la mantenibilidad desde las fases iniciales del desarrollo, reconociendo su impacto en la reducción de costos y tiempo invertidos en la etapa de mantenimiento. Se utilizó el método de estudio de casos en un proyecto de desarrollo de software para evaluar las diferencias en métricas de mantenibilidad antes y después de aplicar este proceso. Los participantes de este estudio fueron los profesionales involucrados en el desarrollo del software bajo análisis. Los resultados mostraron mejoras significativas, evidenciando una reducción tanto en la complejidad como en la densidad de código duplicado después de implementar el proceso diseñado, lo que reflejó una mayor efectividad en la gestión de mantenibilidad desde las fases iniciales del desarrollo. En conclusión, este enfoque destacó la importancia de atender la mantenibilidad desde

el principio del ciclo de vida del software, contribuyendo a la calidad del producto y a la eficiencia en los procesos de desarrollo y mantenimiento.

Del mismo modo; Echavarría et al. (2020) mencionaron que el objetivo fue analizar el impacto de la inteligencia artificial en el sector financiero, específicamente en la toma de decisiones de inversión. Se trató de un estudio de tipo exploratorio, diseño mixto. La población objetivo comprendía gestores financieros, analistas de mercado y expertos en inteligencia artificial involucrados en el sector financiero. La muestra se conformó por una selección estratificada de 300 profesionales provenientes de diferentes entidades financieras y centros de investigación. Los resultados revelaron un aumento significativo en la precisión de las predicciones de inversión con el uso de algoritmos de inteligencia artificial, destacando la optimización en la gestión de riesgos y el rendimiento de las carteras de inversión. En conclusión, la integración de la inteligencia artificial en el ámbito financiero demostró ser fundamental para mejorar la eficiencia en la toma de decisiones de inversión, generando impactos positivos en la rentabilidad y la gestión de riesgos en el mercado.

Asimismo; Hernández et al. (2020) señalaron que el objetivo fue determinar la importancia de los indicadores productivos en profesionales del desarrollo de software que utilizan métodos ágiles. Se empleó una encuesta para abordar esta cuestión. Los resultados resaltaron dos aspectos relevantes: la evaluación del esfuerzo para implementar un sistema y la medición de las ventajas para el cliente durante el desarrollo. Este trabajo allana el camino para establecer un proceso de medición que permita una entrega ágil y frecuente del software, generando un valor significativo para los involucrados.

Adicionalmente; Sidorov et al. (2020) indicaron que su objetivo fue presentar un método para calcular la similitud entre programas, especialmente útil para la clasificación temática o la detección de reuso de código, como en casos de posible plagio. Se utilizaron experimentos con el lenguaje de programación Karel. El enfoque de investigación se basó en técnicas de procesamiento de lenguaje natural y recuperación de información para determinar la similitud entre programas y soluciones similares. La población de estudio incluyó un corpus de 9,341 códigos de programas para 100 tareas diferentes y otro corpus con 374 códigos para 34

tareas clasificados según la idea de solución. Los resultados experimentales indicaron que para el corpus con ideas de solución, la representación con trigramas de caracteres mostró mejores resultados, mientras que para el corpus completo, los trigramas de términos junto con el análisis semántico latente obtuvieron los mejores resultados. En conclusión, este método ofrece un enfoque eficaz para determinar similitudes entre programas, siendo la representación de trigramas significativa dependiendo del contexto y la complejidad de los corpus analizados.

Agregando; Stallman (2020) su objetivo fue la publicación de un capítulo específico titulado "The Free Software Definition" extraído del libro "Free Software for a Free Society" de Richard Stallman. Este capítulo es esencial para comprender el movimiento liderado por Stallman y las razones fundamentales detrás de la necesidad de liberar conocimiento en distintos ámbitos científicos y culturales. Sin seguir una estructura de investigación tradicional, la sección brindó un enfoque educativo y divulgativo, enfocado en proporcionar comprensión y reflexión sobre el software libre. No se especifican población o muestra, dado que se trata de la publicación de un texto y no de un estudio empírico. La conclusión se centra en ofrecer un espacio para divulgar información clave del texto de Stallman, contribuyendo así a la comprensión del movimiento del software libre y la importancia de liberar conocimiento en diversas esferas.

Por otra parte; Zapata et al. (2020) indicaron que su objetivo se centró en la enseñanza de buenas prácticas en el desarrollo de software mediante un juego educativo llamado Smellware, destinado a concienciar sobre los "smells" en el código y sus efectos. El enfoque se basó en un diseño educativo que involucraba la identificación y corrección de problemas de código ("smells"). La investigación se enfocó en la aplicación del juego a estudiantes de ingeniería de software. Los resultados mostraron una mejora en el reconocimiento de distintos tipos de "smells" y una mayor conciencia sobre su impacto en el desarrollo de software. En conclusión, Smellware demostró ser efectivo para educar a desarrolladores sobre prácticas de código óptimas y promover una comprensión más profunda de los problemas potenciales en el código.

Asimismo; García et al. (2019) tomaron como objetivo evaluar y seleccionar el sistema informático de gestión hotelera más adecuado para el Hotel 'Los

'Laureles', considerando la amplia variedad de opciones disponibles en el mercado. Se llevó a cabo un estudio con un diseño mixto que combinó métodos cualitativos y cuantitativos, utilizando el método AHP (Proceso de Jerarquía Analítica) para cuantificar la importancia relativa de los criterios de evaluación de los diferentes software. La población de interés abarcó tanto cadenas hoteleras con sistemas corporativos como hoteles independientes que empleaban sistemas más pequeños. La muestra se centró específicamente en la evaluación de software libre para la gestión hotelera. Los resultados revelaron que el software Abanq-Os fue el más adecuado y adaptable a las necesidades y condiciones particulares del Hotel 'Los Laureles'. En conclusión, el uso del método AHP demostró ser útil para tomar decisiones informadas en la selección de software, permitiendo la elección del sistema más idóneo para la gestión hotelera específica de este establecimiento.

También; González et al. (2019) mencionaron como objetivo diseñar una arquitectura de analítica visual para comprender sistemas de software en varios lenguajes. La población objetivo fueron administradores y desarrolladores interesados en analizar proyectos de software. La muestra incluyó representaciones visuales integradas en Eclipse y Visual Studio, demostrando la viabilidad y utilidad de esta arquitectura. Los resultados resaltaron la especificación del diseño y los requisitos, evidenciando el potencial de esta analítica visual para comprender sistemas en diversos lenguajes. En resumen, este trabajo subrayó la importancia de la analítica visual para un análisis profundo de sistemas de software, proporcionando un marco sólido para una comprensión más detallada y eficiente.

Por otro lado; como antecedente a nivel nacional, Gutiérrez et al. (2023) indicaron que la relevancia que ChatGPT ha adquirido en el sector educacional en la actualidad es de vital importancia. Este estudio tiene como objetivo confirmar la validez de ciertas variables predictivas, como la eficiencia en la adquisición de información, la creatividad, la competencia en escritura, la competencia lingüística, el rendimiento académico y la satisfacción, con el fin de desarrollar instrumentos de investigación sólidos y confiables. La investigación se llevó a cabo en Perú y contó con una muestra de 400 estudiantes universitarios de ambos géneros. Se realizaron pruebas de análisis factorial exploratorio y confirmatorio. A través de este estudio, se establecerán las bases para futuras investigaciones que explorarán y

comprenderán plenamente la utilidad de ChatGPT en diversos contextos y su impacto en la educación superior.

Como definición de ChatGPT, EDEM (2023) indica que se trata de la herramienta que ha generado más discusión y ha captado la atención por todos los medios como la red social entre otros. Chat GPT marcó un punto que sobresale mundialmente dentro de la IA, trascendiendo en los sectores de ciencia y especialidades para asombrar socialmente en conjunto. Las amplias capacidades de Chat GPT han suscitado diferentes interrogantes. ¿Qué es en realidad? ¿Posee inteligencia propia? ¿Tiene alguna utilidad? ¿Continuará su aprendizaje con el tiempo hasta superar a los seres humanos? Para abordar estas preguntas, o al menos la mayoría de ellas, nos sumergiremos en el mundo de la propia herramienta Chat GPT. Asimismo, algunas diferencias entre GPT-3 y GPT-4 son, el razonamiento avanzado: Ahora puede interpretar una mayor cantidad de información y hacerlo de manera más rápida. También las instrucciones complejas: Es capaz de comprender preguntas y consultas más complejas o elaboradas. Asimismo la creatividad: Ofrece respuestas más creativas y puede incluso componer canciones, poemas o guiones. Además, puede generar y editar contenido técnico o creativo de acuerdo al estilo de escritura de cada usuario. También la colaboración: Para enriquecer la experiencia del usuario, el chatbot puede incluir imágenes en las respuestas y generar subtítulos y clasificaciones. Si el usuario lo solicita, el asistente virtual también puede realizar análisis y evaluaciones sobre una variedad de temas. La empresa invirtió más de seis meses en mejorar la aplicación para hacerla más segura. Además, GPT-4 tiene menos probabilidades de ofrecer respuestas a contenidos no permitidos y tiene una mayor capacidad para generar respuestas basadas en hechos en comparación con la versión GPT-3.5.

También hablando de la Capacidad de ChatGPT, en primer lugar, el programa tiene la capacidad de procesar hasta 25,000 palabras simultáneamente, superando con creces la capacidad de los agentes humanos. Gracias a su nutrida base de datos, que incluye información obtenida en parte de internet, puede responder a una amplia variedad de consultas, independientemente de la disciplina o el campo del conocimiento. Ofrece resultados en múltiples formatos, desde

respuestas en texto sencillo hasta la generación de poemas, haikus o ensayos. Su velocidad de respuesta es impresionante, ya que puede generar respuestas en cuestión de milisegundos, aunque algunas respuestas más complejas pueden demorar unos segundos. El sistema se somete a actualizaciones periódicas, lo que significa que sus funciones y su capacidad de procesamiento de información se optimizan de manera regular. Comenzó como una versión GPT-3 y ahora opera como una GPT-4, lo que le permite tener menos restricciones y mayor potencia. Además, tiene la capacidad de integrarse con otras herramientas como Excel o Chrome. Puedes consultar las principales extensiones de ChatGPT para Chrome. Es importante destacar que existen ciertos contenidos sensibles o altamente especializados que este recurso no puede abordar, como cuestiones médicas, temas de actualidad extrema o contenido ofensivo.

Del mismo modo los Prompts de ChatGPT para el desarrollo web son categorizados en tres grupos: El primer grupo es Solicitudes a ChatGPT: Desarrolla una página web para un festival local utilizando JavaScript. Continúa el siguiente código (introduce el código). Identifica errores en este código (introduce el código). Encuentra posibles errores en este código (introduce el código). Proporciona cinco razones para considerar el uso de JavaScript en el desarrollo web. El segundo grupo es Consultas a ChatGPT: Explícame cuál es la parte más crucial en el desarrollo web. ¿Cuál sería el código óptimo para una página web de este tipo? Bríndame sugerencias sobre diseño de páginas web. El tercer grupo es Preguntas a ChatGPT: ¿Cuáles son las prácticas recomendadas en el diseño web? ¿Cómo puedo abordar el aspecto del diseño? ¿Cómo puedo solucionar los errores presentes en este código? ¿Cuáles son los factores que debo considerar al escribir código con Python?

También la Arquitectura se basa en la arquitectura GPT (Generative Pre-trained Transformer). En su proceso de entrenamiento, se alimenta con grandes cantidades de texto de Internet para que el modelo adquiera un conocimiento amplio del lenguaje humano y la estructura de conversaciones. Durante el entrenamiento, el modelo se ajusta para predecir la siguiente palabra en una oración, lo que le permite aprender a generar texto coherente y contextualmente relevante. OpenAI utiliza un enfoque de "aprendizaje por refuerzo" para mejorar aún

más el rendimiento de ChatGPT. Esto implica entrenar al modelo para interactuar con entornos simulados y humanos reales, optimizando sus respuestas y habilidades de conversación.

Finalmente; se ha visto unas capacidades impresionantes como: Generación de Texto Conversacional: ChatGPT es capaz de generar respuestas coherentes y contextuales en conversaciones escritas o habladas, lo que lo convierte en una herramienta valiosa para crear chatbots y asistentes virtuales que pueden interactuar de manera más natural con los usuarios. Generación de Contenido: Además de conversaciones, ChatGPT puede generar contenido escrito, como artículos, informes, descripciones de productos y más. Esto lo hace útil para la creación automática de contenido en diversas industrias. Soporte al Cliente: ChatGPT se utiliza para proporcionar respuestas rápidas y precisas a las consultas de los clientes, lo que mejora la eficiencia y la satisfacción del cliente en empresas de atención al cliente y servicios en línea. Educación y Tutoría: En el campo de la educación, ChatGPT puede actuar como un tutor virtual, proporcionando explicaciones claras y respuestas a preguntas de los estudiantes. Traducción Automática: Puede ser empleado en aplicaciones de traducción automática, donde ChatGPT es capaz de traducir de manera efectiva entre idiomas. Generación Creativa de Contenido: ChatGPT puede ayudar a escritores, diseñadores y creadores de contenido a generar ideas creativas y textos únicos. Estas capacidades y aplicaciones hacen de ChatGPT una herramienta versátil y poderosa en el campo del procesamiento de lenguaje natural y la inteligencia artificial. Su capacidad para comprender el contexto y generar respuestas coherentes lo hace especialmente valioso en escenarios de comunicación interactiva.

Por otro lado; como definición para desarrollo de código fuente de software, Según Arimetrics (2022) se refiere a una colección de instrucciones escritas en un lenguaje de programación legible por humanos, generalmente en formato de texto sin formato, que se utiliza para facilitar el trabajo de los programadores de computadoras. En esencia, el código fuente constituye la base de un programa de computadora y contiene declaraciones, instrucciones y funciones que permiten que el programa funcione correctamente. Cuando nos referimos al código fuente, estamos hablando del núcleo de un programa informático, que puede abarcar

desde el desarrollo de una página web hasta aplicaciones más complejas. Para que el código fuente sea utilizable, debe ser generado mediante un programa y escrito en uno de los lenguajes de programación disponibles. Sin embargo, en su forma inicial, el código fuente no es ejecutable por las computadoras. Por lo tanto, es necesario traducirlo a otro lenguaje o a código binario para que la computadora pueda entender y ejecutar las instrucciones contenidas en él. Esta traducción se lleva a cabo mediante compiladores, intérpretes, ensambladores u otros sistemas de traducción. La compilación es una etapa crucial que debe completarse para que el código fuente se vuelva práctico y pueda utilizarse en una computadora. Es importante mencionar que el código fuente puede estar sujeto a licencias de uso que determinan si los usuarios tienen permiso para consultar, reutilizar o modificar el código según los términos de la licencia correspondiente.

Asimismo; Boehm (1988) mencionó que como pasos se tiene primero a Requisitos y Diseño: Antes de comenzar a escribir código, es esencial comprender los requisitos del software y diseñar su arquitectura. Esto implica la identificación de las características necesarias, la creación de diagramas y la planificación de la estructura del programa. Luego la Codificación: Durante esta fase, los programadores escriben el código fuente en el lenguaje de programación elegido. Este código implementa la funcionalidad del software de acuerdo con el diseño previamente establecido. También las Pruebas Unitarias: Cada módulo o componente del software debe someterse a pruebas unitarias para verificar su correcto funcionamiento. Esto implica la creación de casos de prueba para probar cada parte del código individualmente. Asimismo la Integración y Pruebas de Sistema: Los diferentes módulos del software se combinan y se someten a pruebas de integración para asegurarse de que funcionen correctamente juntos. Luego, se realizan pruebas de sistema para garantizar que el software en su conjunto cumpla con los requisitos. También la Depuración: Durante todo el proceso de desarrollo, los programadores identifican y corrigen errores o fallos en el código. La depuración es una parte esencial del desarrollo de software. Siguiendo con la Optimización y Rendimiento: Se puede realizar la optimización del código para mejorar la velocidad y la eficiencia del software. Esto implica la identificación de cuellos de botella y la implementación de soluciones para mejorar el rendimiento. Luego la Documentación: Es importante documentar el código fuente para que otros

desarrolladores puedan comprender y colaborar en el proyecto. La documentación incluye comentarios en el código, descripciones de funciones y manuales de usuario si es necesario. Ahora el Control de Versiones: Para gestionar el código fuente, se utilizan sistemas de control de versiones como Git. Estos sistemas permiten llevar un registro de los cambios, colaborar con otros programadores y volver a versiones anteriores del código si es necesario. Finalmente el Despliegue y Mantenimiento: Una vez que el software ha pasado todas las pruebas y está listo para su uso, se despliega en un entorno de producción. Después del lanzamiento, el mantenimiento continuo es crucial para corregir errores, aplicar actualizaciones y agregar nuevas características.

Por otro lado; Ghezzi et al. (2002) mencionan algunos otros conceptos como los lenguajes de programación, es el código fuente puede estar escrito en una variedad de lenguajes de programación, como Python, Java, C++, JavaScript, Ruby, entre otros. La elección del lenguaje depende de las necesidades del proyecto y de las preferencias del desarrollador. También los entornos de Desarrollo, los desarrolladores utilizan entornos de desarrollo integrados (IDE) para escribir, depurar y probar su código. Ejemplos de IDE populares incluyen Visual Studio, Eclipse, PyCharm y Sublime Text. Así se mencionan las metodologías de desarrollo, existen varias metodologías para gestionar el desarrollo de código fuente, como el modelo en cascada, el desarrollo ágil y DevOps. Cada enfoque tiene sus propias ventajas y desafíos. También herramientas y Frameworks, los desarrolladores a menudo utilizan herramientas y frameworks específicos para acelerar el desarrollo, como React para aplicaciones web, TensorFlow para aprendizaje automático o Django para desarrollo web.

Asimismo; Govardhan (2010) mencionó a los siguientes métodos de desarrollo como el Desarrollo en Cascada: Este enfoque sigue una secuencia lineal de actividades, desde la especificación de requisitos hasta el despliegue. Cada fase debe completarse antes de avanzar a la siguiente. Es adecuado para proyectos con requisitos bien definidos, pero puede ser inflexible para cambios. También; el Desarrollo Ágil: Los métodos ágiles, como Scrum y Kanban, se centran en la colaboración, la iteración y la respuesta rápida a los cambios. Estos enfoques son ideales para proyectos en los que los requisitos no son estáticos y deben adaptarse

a medida que avanza el desarrollo. Asimismo; DevOps: es una filosofía que combina desarrollo (Dev) y operaciones (Ops) para mejorar la colaboración y la automatización en todo el ciclo de vida del software. Facilita la entrega continua y la implementación sin problemas. Del mismo modo; el Desarrollo en Espiral: Este enfoque combina elementos de desarrollo en cascada con ciclos iterativos. Cada iteración produce una versión más completa del software, lo que permite una mayor flexibilidad y adaptación a cambios de requisitos.

También; O'Regan (2016) indicó que la Calidad del Código es esencial para garantizar que el software sea confiable, seguro y mantenible. Los principios de la calidad del código incluyen la legibilidad, la eficiencia, la modularidad, la escalabilidad y la seguridad. Herramientas como linters, analizadores estáticos y pruebas unitarias ayudan a garantizar la calidad del código. En relación a la Gestión de Proyectos que es fundamental para asegurarse de que el trabajo se realice de manera eficiente y dentro del presupuesto. Las metodologías de gestión de proyectos, como el enfoque Scrum o Kanban, ayudan a organizar tareas, asignar recursos y establecer plazos. Asimismo la Documentación del código es esencial para que otros desarrolladores puedan comprender y mantener el software. Esto incluye comentarios en el código, documentación técnica, manuales de usuario y guías de instalación. También las Pruebas y Validación son una parte crítica del desarrollo de software. Esto incluye pruebas unitarias, pruebas de integración, pruebas de regresión y pruebas de usuario. La validación y verificación son procesos clave para asegurarse de que el software cumple con los requisitos.

Finalmente; OWASP Foundation (2020) mencionó que la Seguridad del Código es un aspecto crítico, especialmente en aplicaciones que manejan datos confidenciales. La identificación y corrección de vulnerabilidades, como inyección de SQL o fallos de seguridad, son esenciales. También se menciona que para el Licenciamiento y Propiedad Intelectual es importante comprender y gestionar adecuadamente los aspectos legales relacionados con el código fuente, incluyendo el licenciamiento del software y la protección de la propiedad intelectual. Asimismo el desarrollo de código fuente es solo una parte del Ciclo de Vida del Software. Después del desarrollo, el software pasa por etapas de prueba, implementación, mantenimiento y, finalmente, retirada.

III. METODOLOGÍA

3.1. Tipo y diseño de investigación

3.1.1. Tipo de investigación: Aplicada

Según Lozada (2014) el tipo de investigación aplicada se enfoca en estudios científicos que buscan solucionar problemas de la vida diaria y controlar situaciones prácticas.

3.1.2. Diseño de investigación: No experimental, transversal, descriptivo, correlacional

Según QuestionPro (2020), se refiere a un enfoque que busca comprender las relaciones entre variables existentes sin manipularlas directamente. En este diseño, se recopilan datos en un momento específico y se analizan para identificar posibles asociaciones o correlaciones entre diferentes factores, sin establecer relaciones causales. Asimismo, Vega (2021) indica que se centra en observar y medir variables en un único punto temporal, buscando conexiones o patrones entre ellas, pero sin intervenir en su naturaleza.

3.1.3. Enfoque de investigación: Cuantitativo

De acuerdo a Rus (2021) a diferencia de la investigación cualitativa, la investigación cuantitativa se centra en el análisis de una gran cantidad de datos. Se enfoca en variables numéricas, conocidas como variables cuantitativas, cuyo significado se revela al relacionarse con otras mediante correlaciones, regresiones o pruebas de hipótesis.

3.2. Variables y operacionalización

3.2.1. Variable Independiente: ChatGPT

Definición Conceptual

Inteligencia artificial desarrollada por OpenAI, basada en la arquitectura GPT (Transformador Generativo Preentrenado), diseñada para comprender y generar texto de manera autónoma en respuesta a consultas o interacciones humanas. Utiliza un extenso conjunto de datos para aprender patrones lingüísticos, generar respuestas coherentes y relevantes en diversos ámbitos, resolviendo problemas hasta generando contenido creativo. (EDEM, 2023)

Definición Operacional

Se refiere a la presencia y funcionamiento del modelo de lenguaje de inteligencia artificial desarrollado por OpenAI conocido como "ChatGPT". Esta variable se considera presente cuando el sistema ChatGPT está activo y disponible para interactuar con usuarios a través de texto o voz.

3.2.2. Variable Dependiente: Código fuente de software

Definición Conceptual

Se refiere a una colección de instrucciones escritas en un lenguaje de programación legible por humanos, generalmente en formato de texto sin formato, que se utiliza para facilitar el trabajo de los programadores de computadoras. (Armetrics, 2021)

Definición Operacional

Se refiere al conjunto de instrucciones escritas en un lenguaje de programación que define el funcionamiento y comportamiento de un programa o aplicación informática.

3.2.3. Operacionalización

La matriz de operacionalización se puede apreciar en el segundo anexo.

3.3. Población, muestra y muestreo

3.3.1. Población

Según López (2004) se refiere al conjunto de individuos o elementos de los cuales se busca obtener información. En nuestro contexto, esta población puede variar y estar compuesta por una amplia gama de elementos, que van desde personas, registros, entre otros.

La consultora tiene alrededor de 180 historias de usuario al mes en promedio que fue considerado en esta investigación.

3.3.2. Muestra

De acuerdo a la Universidad Veracruzana (2020) se refiere a una porción o subconjunto más reducido de la población total. En el contexto de encuestas, esta muestra se compone de los individuos de la población que son seleccionados y a quienes se les invita a participar en la encuesta.

La muestra contó con 124 historias de usuario.

3.3.3. Muestreo

Según Muguirra (2023) es una técnica que facilita la selección de una fracción específica de la población, permitiendo así analizar sus datos y obtener los resultados necesarios para la investigación.

Se usó el muestreo tipo no probabilístico, aleatorio simple.

3.4. Técnicas e instrumentos de recolección de datos

3.4.1. Técnica

Análisis documental

Involucra la identificación, clasificación y evaluación de datos con el fin de extraer información relevante o datos significativos. (Dulzaides et al. 2004)

3.4.2. Instrumento

Ficha de análisis

Permitirá sistematizar la información para facilitar su posterior consulta y referencia en la investigación. Al organizar la información en fichas, los investigadores pueden acceder de manera más eficiente a los datos recopilados y utilizarlos para respaldar sus argumentos, tesis o conclusiones. (Lifeder, 2022)

3.4.3. Confiabilidad

Para la confiabilidad de la implementación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios, se empleará el coeficiente Alfa de Cronbach, un análisis realizado mediante el programa SPSS. La confiabilidad es

esencial para garantizar que las acciones sean precisas y uniformes, lo que, a su vez, contribuirá al éxito del proyecto.

De acuerdo a Narvaez (2020) el coeficiente Alfa de Cronbach analiza la coherencia entre las respuestas de las diversas preguntas o ítems en un instrumento de medición. Un valor alto indica una consistencia entre los ítems, sugiriendo que todos miden de manera uniforme un mismo rasgo o idea. Se emplea una escala específica para interpretar el nivel de confiabilidad de este coeficiente:

- 0,00 a 0,40: Baja confiabilidad
- 0,40 a 0,60: Aceptable confiabilidad
- 0,60 a 0,80: Buena confiabilidad
- 0,80 a 0,90: Muy buena confiabilidad
- 0,90 a 1,00: Excelente confiabilidad

Tabla 1

Estadística de fiabilidad de Alfa de Cronbach

Alfa de Cronbach	N de elementos	N de casos
,887	11	124

Fuente: Diseño propio

La Tabla 1 indica un coeficiente Alfa de Cronbach de 0.887 para un conjunto de 124 casos. Este valor señala una consistencia excepcional entre los ítems del instrumento al medir un mismo concepto. Con un coeficiente de 0.887, se confirma una confiabilidad muy alta, respaldando la solidez de los ítems al medir el mismo concepto. Este resultado valida el instrumento como una herramienta confiable para evaluar el constructo de interés.

3.4.4. Prueba de normalidad

Regla de decisión:

Si $\text{Sig} < 0.05$, se rechaza la hipótesis nula.

Si $\text{Sig} \geq 0.05$, se acepta la hipótesis nula.

Dimensión 1: Complejidad

Hipótesis nula (H0): Los datos de la dimensión Complejidad provienen de una distribución normal.

Hipótesis alternativa (H1): Los datos de la dimensión Complejidad NO provienen de una distribución normal.

Tabla 2*Prueba de normalidad para la Dimensión 1*

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Dimensión 1: Complejidad	,235	124	,000	,799	124	,000

a. Corrección de significación de Lilliefors

Fuente: Diseño propio

Como el Sig es $0.000 < 0.05$, se rechaza la hipótesis nula. Se concluye que los datos de la dimensión Complejidad NO provienen de una distribución normal.

Dimensión 2: Precisión

Hipótesis nula (H0): Los datos de la dimensión Precisión provienen de una distribución normal.

Hipótesis alternativa (H1): Los datos de la dimensión Precisión NO provienen de una distribución normal.

Tabla 3*Prueba de normalidad para la Dimensión 2*

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Dimensión 2: Precisión	,244	124	,000	,783	124	,000

a. Corrección de significación de Lilliefors

Fuente: Diseño propio

Como el Sig es $0.000 < 0.05$, se rechaza la hipótesis nula. Se concluye que los datos de la dimensión Precisión NO provienen de una distribución normal.

Dimensión 3: Eficiencia

Hipótesis nula (H0): Los datos de la dimensión Eficiencia provienen de una distribución normal.

Hipótesis alternativa (H1): Los datos de la dimensión Eficiencia NO provienen de una distribución normal.

Tabla 4*Prueba de normalidad para la Dimensión 3*

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Dimensión 3: Eficiencia	,280	124	,000	,850	124	,000

a. Corrección de significación de Lilliefors

Fuente: Diseño propio

Como el Sig es $0.000 < 0.05$, se rechaza la hipótesis nula. Se concluye que los datos de la dimensión Eficiencia NO provienen de una distribución normal.

Dado que ninguna de las 3 dimensiones se distribuye normalmente, se utilizarán pruebas no paramétricas para la contrastación de hipótesis.

3.5. Procedimientos

En el marco de esta investigación, se llevó a cabo un proceso de recopilación de datos y observación con el objetivo de determinar la relación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios en una consultora en Lima durante el año 2023. Para este fin, se diseñó un instrumento de recopilación de datos en forma de ficha de análisis. Esta ficha de análisis se convirtió en un recurso valioso para registrar de manera sistemática y estructurada los procedimientos y prácticas empleados en el desarrollo de historias de usuarios, con énfasis en la relación de ChatGPT. La implementación de ChatGPT en la consultora seleccionada permitió la generación de datos relevantes, incluyendo la complejidad, precisión y eficiencia en la creación de código fuente con la asistencia de ChatGPT. Los resultados obtenidos se analizaron para evaluar la relación de ChatGPT en el desarrollo de código fuente de software, respaldando conclusiones y recomendaciones específicas para las consultoras y empresas de desarrollo de software que estén considerando la implementación de ChatGPT en sus operaciones. Este procedimiento es fundamental para lograr los objetivos de la investigación, proporcionando una visión detallada y objetiva de la relación de ChatGPT en el proceso de desarrollo de software, respaldado por una recopilación de datos detallados y una evaluación en tiempo real.

3.6. Método de análisis de datos

Para llevar a cabo el proceso de recopilación y análisis de datos en este estudio, se ha adoptado una estrategia de medición precisa y un enfoque inferencial. La elección de esta metodología se ha realizado de manera meticulosa con el objetivo de lograr una comprensión profunda y precisa de los resultados del estudio, en particular, en lo que respecta a la relación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios en una consultora en Lima durante el año 2023. El análisis de los datos se ha iniciado con la recopilación detallada de información, utilizando herramientas como la ficha de análisis diseñada específicamente para este propósito. Una vez obtenidos estos datos, se ha procedido a su organización y estructuración en tablas y gráficos avanzados.

Cada tabla y gráfico ha sido cuidadosamente configurado para representar de manera efectiva los datos relevantes. Estas representaciones visuales han sido diseñadas de tal manera que resalten patrones, tendencias y relaciones clave en los datos recopilados. Esto facilita una visualización más precisa de los resultados y simplifica la identificación de cualquier variación o consistencia en el desarrollo de código fuente de software bajo la relación de ChatGPT. Además, se lleva a cabo un análisis que involucra la aplicación de métodos estadísticos inferenciales. Estos enfoques permiten extrapolar los resultados obtenidos en la muestra al conjunto de la población estudiada, fortaleciendo así la validez y la confiabilidad de las conclusiones. Además, posibilitan la identificación de relaciones significativas y la evaluación de la relación de ChatGPT en el desarrollo de software para historias de usuarios.

3.7. Aspectos éticos

El estudio sobre cuestiones éticas se llevó a cabo con total transparencia y honestidad, siguiendo estrictamente los criterios establecidos. Primero, se respetaron cabalmente los derechos de autor de los autores de los artículos, otorgándoles la adecuada atribución y proporcionando las correspondientes referencias bibliográficas. Además, se mantuvo la transparencia al seleccionar exclusivamente artículos publicados en los últimos cinco años y que estuvieran

incluidos en bases de datos confiables. En nuestro rol como investigadores, cumplimos con el artículo 9 ° de la Política contra el plagio establecido en el código de ética de investigación de la Universidad César Vallejo, asegurando una citación uniforme de todos los autores que participaron en la investigación. Por último, recurrimos a Turnitin para identificar posibles similitudes en el documento, con el objetivo de satisfacer los requerimientos mínimos establecidos por la institución, haciendo especial énfasis en el cumplimiento de la RESOLUCIÓN DE CONSEJO UNIVERSITARIO N° 0470-2022/UCV, específicamente en su artículo N°10.

IV. RESULTADOS

Análisis descriptivo:

Primera dimensión: Complejidad

Tabla 5

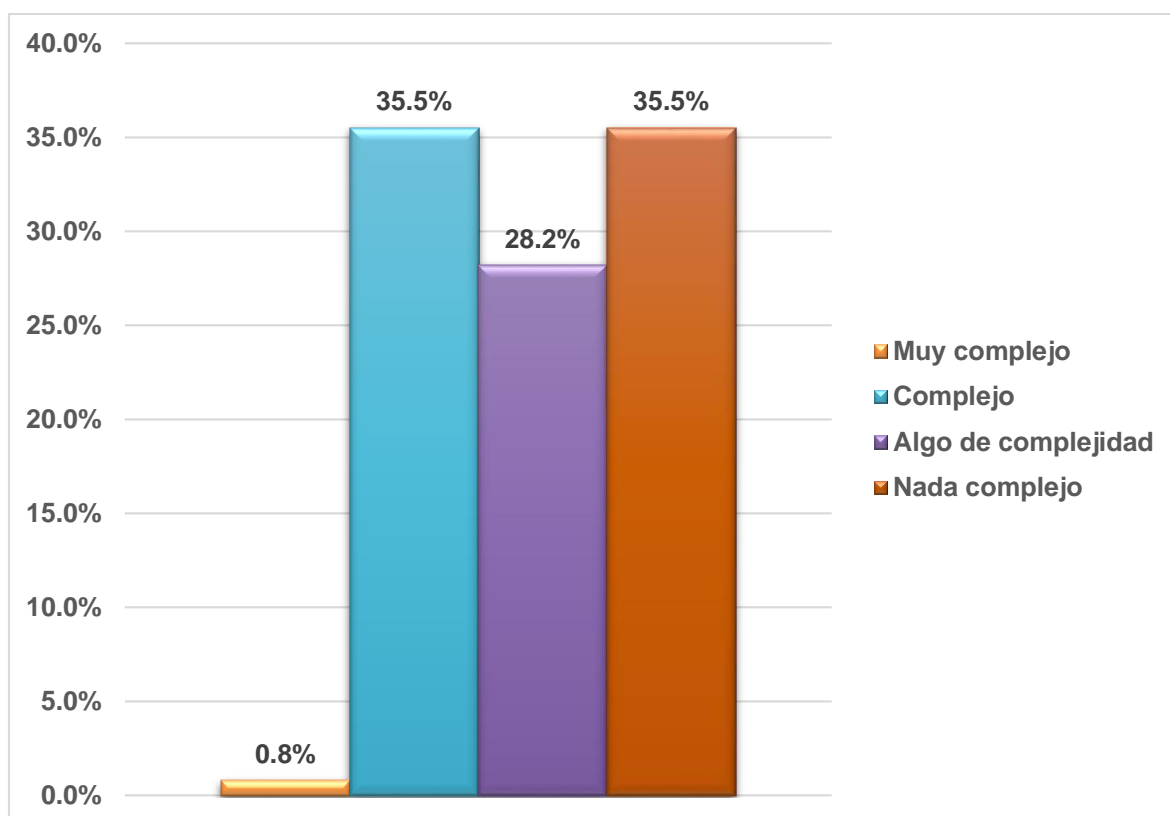
Dimensión 1: Complejidad

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Muy complejo	1	0,8	0,8	0,8
	Complejo	44	35,5	35,5	36,3
	Algo de complejidad	35	28,2	28,2	64,5
	Nada complejo	44	35,5	35,5	100,0
	Total	124	100,0	100,0	

Fuente: Diseño propio

Figura 1

Dimensión 1: Complejidad



Fuente: Diseño propio

En relación a la tabla 2 y figura 1, se observa que el 35,5% de los resultados indican que el código es “Nada complejo”. Mientras que un 0,8% indica que el código es “Muy complejo”.

Segunda dimensión: Precisión

Tabla 6

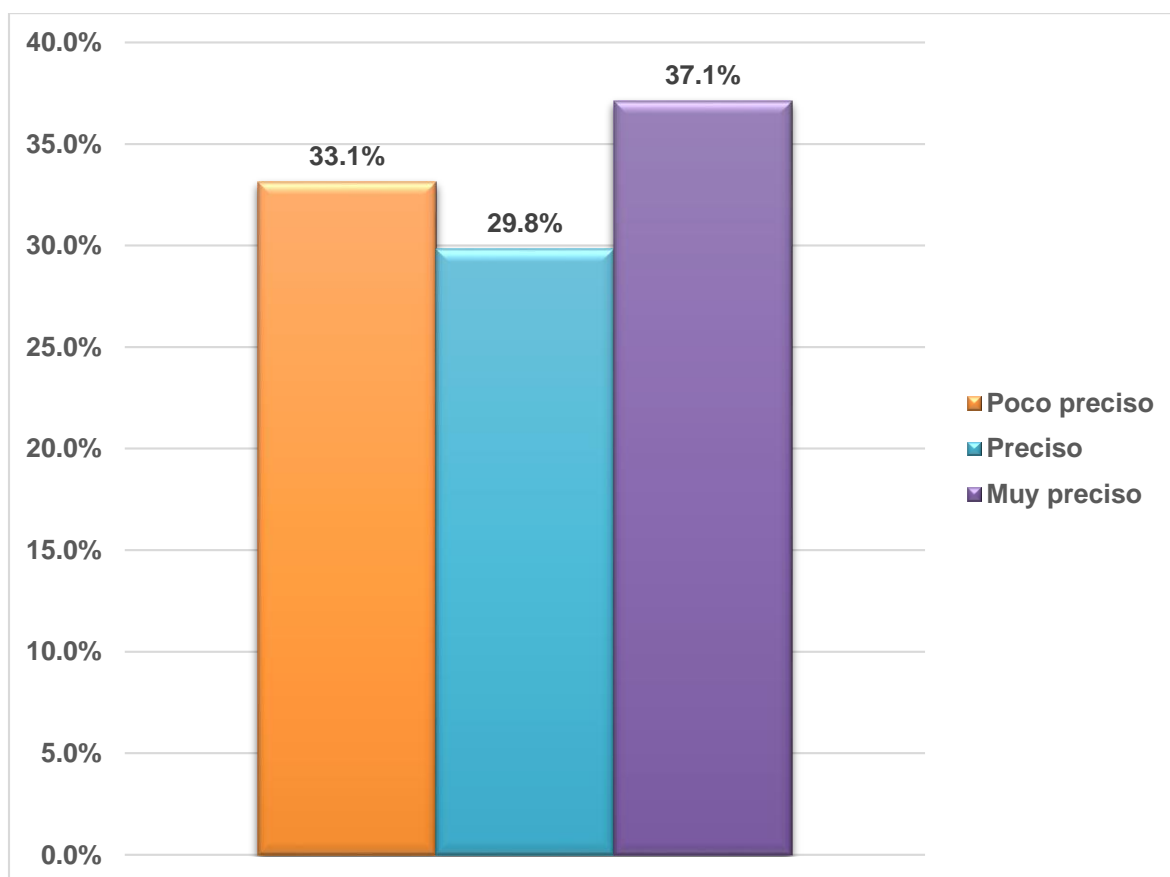
Dimensión 2: Precisión

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Poco preciso	41	33,1	33,1	33,1
	Preciso	37	29,8	29,8	62,9
	Muy preciso	46	37,1	37,1	100,0
	Total	124	100,0	100,0	

Fuente: Diseño propio

Figura 2

Dimensión 2: Precisión



Fuente: Diseño propio

En relación a la tabla 3 y figura 2, se observa que el 37,1% de los resultados indican como “Muy preciso”. Mientras que un 29,8% indica que la precisión es “Preciso”.

Tercera dimensión: Eficiencia

Tabla 7

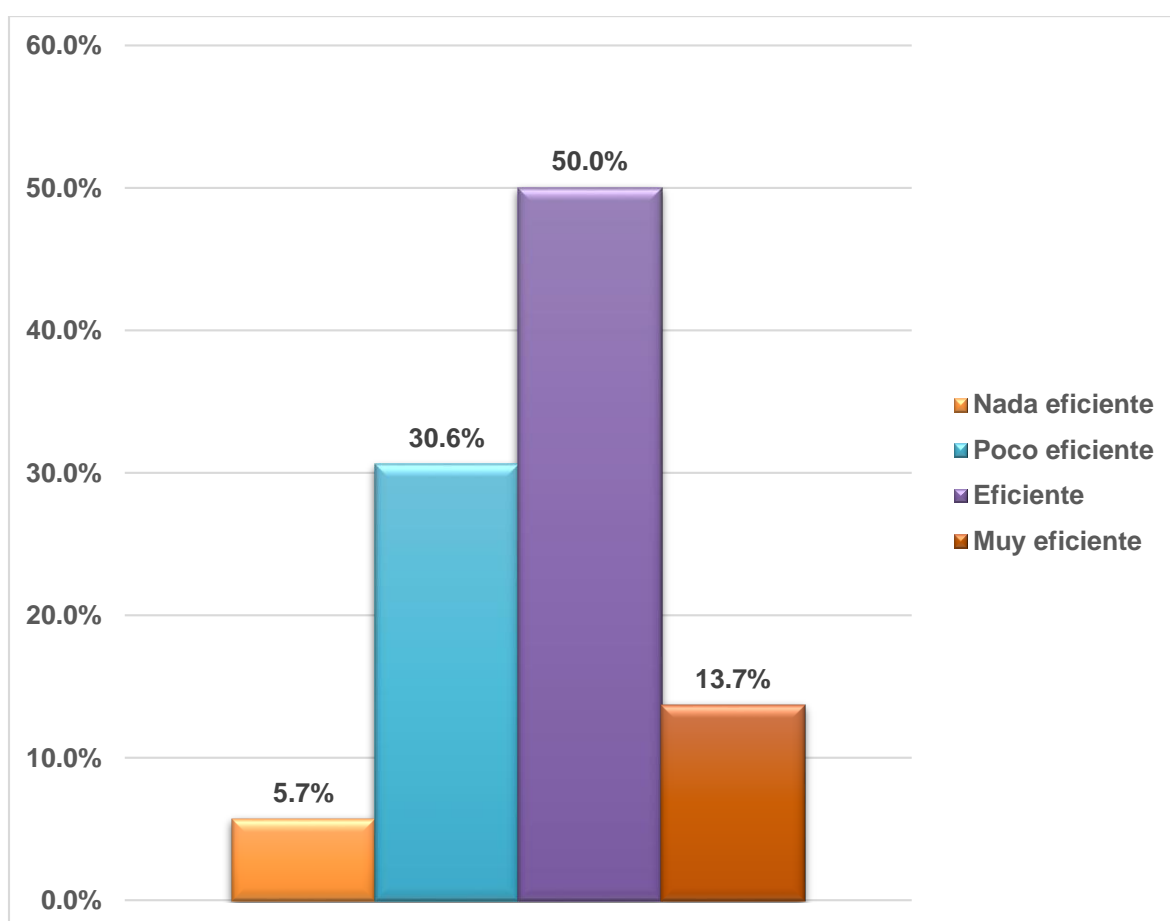
Dimensión 3: Eficiencia

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Nada eficiente	7	5,7	5,7	5,7
Poco eficiente	38	30,6	30,6	36,3
Válido Eficiente	62	50,0	50,0	86,3
Muy eficiente	17	13,7	13,7	100,0
Total	124	100,0	100,0	

Fuente: Diseño propio

Figura 3

Dimensión 3: Eficiencia



Fuente: Diseño propio

En relación a la tabla 4 y figura 3, se observa que el 50,00% de los resultados indican como “Eficiente” el código fuente. Mientras que un 5,7% lo indica como “Nada eficiente”.

Resultado de pruebas de correlación

Los resultados de las pruebas de correlación para la variable independiente ChatGPT con la variable dependiente Código fuente de software, se evidencia en la tabla siguiente:

Tabla 8

Prueba de correlación de las variables

		Correlaciones	ChatGPT
Rho		Coefficiente de correlación	,850**
Spearman	Código fuente de software	Sig. (bilateral)	,000
		N	124

** . La correlación es significativa en el nivel 0,01 (bilateral).

Fuente: *Diseño propio*

Hipótesis nula (H0): No existe correlación significativa entre ChatGPT y el Código fuente de software

Hipótesis alternativa (H1): Existe correlación significativa entre ChatGPT y el Código fuente de software

Con un nivel de significancia del 5%, se rechaza la hipótesis nula si el Sig es menor a 0.05.

La interpretación de los resultados de la Tabla 8, son:

- 1) Coeficiente de Correlación de Spearman (Rho) entre *ChatGPT* y el *código fuente de software* es de 0.850. Este coeficiente indica la fuerza y dirección de la relación entre las dos variables.
- 2) Significancia Estadística: La correlación es significativa a un nivel de significancia del 0.01 (bilateral).
- 3) El Sig. es menor a 0.05, y conforme la Tabla 8, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, de que existe correlación significativa entre ChatGPT y el Código fuente de software en una consultora, Lima, 2023.

La correlación de 0.850 sugiere una fuerte relación positiva entre *ChatGPT* y el *código fuente de software*, en la muestra analizada. La significancia estadística respalda que las variables están relacionadas. Por lo tanto, existe una asociación significativa y positiva entre la variable independiente *ChatGPT* y el variable dependiente *código fuente de software* en la muestra analizada.

Dado que las dimensiones no se distribuyen normalmente, se utilizará la prueba no paramétrica de Rho de Spearman para evaluar la correlación entre las dimensiones y el código fuente de software.

Los resultados de las pruebas de correlación para la primera dimensión, se evidencia en la tabla siguiente:

Tabla 9

Prueba de correlación de la dimensión 1 y la variable dependiente

Correlaciones		Complejidad
Rho	Coeficiente de correlación	,892**
Spearman	Sig. (bilateral)	,000
	N	124

** La correlación es significativa en el nivel 0,01 (bilateral).

Fuente: *Diseño propio*

Hipótesis nula (H0): No existe correlación significativa entre la complejidad y el Código fuente de software

Hipótesis alternativa (H1): Existe correlación significativa entre la complejidad y el Código fuente de software

Con un nivel de significancia del 5%, se rechaza la hipótesis nula si el Sig es menor a 0.05.

La interpretación de los resultados de la Tabla 9, son:

- 4) Coeficiente de Correlación de Spearman (Rho) entre la *complejidad* y el *código fuente de software* es de 0.892. Este coeficiente indica la fuerza y dirección de la relación entre la dimensión y variable.
- 5) Significancia Estadística: La correlación es significativa a un nivel de significancia del 0.01 (bilateral).
- 6) El Sig. es menor a 0.05, y conforme la Tabla 9, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, de que existe correlación significativa entre la complejidad y el Código fuente de software en la en una consultora, Lima, 2023.
- 7) La correlación de 0.892 sugiere una fuerte relación positiva entre la *complejidad* y el *código fuente de software*, en la muestra analizada. La significancia estadística respalda que las variables están relacionadas. Por lo tanto, existe una asociación significativa y positiva entre la dimensión *complejidad* y la variable dependiente el *código fuente de software* en la muestra analizada.

Los resultados de las pruebas de correlación para la segunda dimensión, se evidencia en la tabla siguiente:

Tabla 10

Prueba de correlación de la dimensión 2 y la variable dependiente

		Correlaciones	Precisión
Rho		Coeficiente de correlación	,813**
Spearman	Código fuente de software	Sig. (bilateral)	,000
		N	124

** La correlación es significativa en el nivel 0,01 (bilateral).

Fuente: *Diseño propio*

Hipótesis nula (H0): No existe correlación significativa entre la precisión y el Código fuente de software

Hipótesis alternativa (H1): Existe correlación significativa entre la precisión y el Código fuente de software

Con un nivel de significancia del 5%, se rechaza la hipótesis nula si el Sig es menor a 0.05.

La interpretación de los resultados de la Tabla 10, son:

- 1) Coeficiente de Correlación de Spearman (Rho) entre la *precisión* y el *código fuente de software* es de 0.813. Este coeficiente indica la fuerza y dirección de la relación entre la dimensión y variable.
- 2) Significancia Estadística: La correlación es significativa a un nivel de significancia del 0.01 (bilateral).
- 3) El Sig. es menor a 0.05, y conforme la Tabla 10, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, de que existe correlación significativa entre la precisión y el Código fuente de software en la en una consultora, Lima, 2023.

La correlación de 0.813 sugiere una fuerte relación positiva entre la *precisión* y el *código fuente de software*, en la muestra analizada. La significancia estadística respalda que las variables están relacionadas. Por lo tanto, existe una asociación significativa y positiva entre la dimensión *precisión* y la variable dependiente el *código fuente de software* en la muestra analizada.

Los resultados de las pruebas de correlación para la tercera dimensión, se evidencia en la tabla siguiente:

Tabla 11

Prueba de correlación de la dimensión 3 y la variable dependiente

		Correlaciones	<i>Eficiencia</i>
Rho		Coeficiente de correlación	,776**
Spearman	Código fuente de software	Sig. (bilateral)	,000
		N	124

** . La correlación es significativa en el nivel 0,01 (bilateral).

Fuente: *Diseño propio*

Hipótesis nula (H0): No existe correlación significativa entre la eficiencia y el Código fuente de software

Hipótesis alternativa (H1): Existe correlación significativa entre la eficiencia y el Código fuente de software

Con un nivel de significancia del 5%, se rechaza la hipótesis nula si el Sig es menor a 0.05.

La interpretación de los resultados de la Tabla 11, son:

- 1) Coeficiente de Correlación de Spearman (Rho) entre la *eficiencia* y el *código fuente de software* es de 0.776. Este coeficiente indica la fuerza y dirección de la relación entre la dimensión y variable.
- 2) Significancia Estadística: La correlación es significativa a un nivel de significancia del 0.01 (bilateral).
- 3) El Sig. es menor a 0.05, y conforme la Tabla 11, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, de que existe correlación significativa entre la *eficiencia* y el *Código fuente de software* en la en una consultora, Lima, 2023.

La correlación de 0.776 sugiere una fuerte relación positiva entre la *eficiencia* y el *código fuente de software*, en la muestra analizada. La significancia estadística respalda que las variables están relacionadas. Por lo tanto, existe una asociación significativa y positiva entre la dimensión *eficiencia* y la variable dependiente el *código fuente de software* en la muestra analizada.

V. DISCUSIÓN

ChatGPT redujo significativamente la complejidad del código fuente según una correlación positiva media-alta de 0.892. Específicamente, se incrementó el porcentaje de código calificado como "nada complejo" a 35.5%. Esta reducción de complejidad concuerda con Tamayo et al. (2021), quienes encontraron mejoras significativas en métricas de complejidad y densidad de código duplicado al gestionar mantenibilidad desde etapas tempranas.

La precisión del código aumentó considerablemente con ChatGPT con una correlación positiva media-alta de 0.813, incrementando los casos calificados como "muy precisos" a 37.1%. Este hallazgo es consistente con conclusiones de Echavarría et al. (2020) sobre cómo la IA mejora la precisión de predicciones y decisiones financieras.

ChatGPT optimizó la eficiencia con una correlación positiva media-alta de 0.776. Específicamente, los casos de código "muy eficiente" se incrementaron a 13.7% luego de incorporar la herramienta. Esta mejora en la eficiencia concuerda con estudios como el de Erazo et al. (2023), que destacan el potencial transformador de la IA al optimizar programas informáticos.

Si bien Bender et al. (2023) resaltaron los progresos en generación automática de código, en esta investigación ChatGPT evidenció algunas limitaciones en casos más complejos, requiriendo complementarse con revisiones humanas para garantizar precisión. Esto se alinea con preocupaciones de Sidorov et al. (2020) sobre posibles imprecisiones de modelos de IA. La relación de ChatGPT reduciendo la complejidad del código ($r=0.892$) tiene implicancias prácticas para mejorar la mantenibilidad y facilidad de cambios, como destacaron Tamayo et al. (2021). Con menos complejidad, se optimiza el tiempo y esfuerzo en etapas de mantenimiento según O'Regan (2016), representando ahorros significativos al evitar retrabajos. Considerando que Bohm (1988) indica que la mantenibilidad es un objetivo clave en la calidad de software, la reducción de complejidad con

ChatGPT potencia attributes críticos de usabilidad y seguridad según OWASP (2020).

El aumento en la precisión del código ($r=0.813$) gracias a ChatGPT refuerza hallazgos de Echavarría et al. (2020) sobre el valor de la IA optimizando decisiones y predicciones en servicios financieros. Con software más preciso, disminuyen fallos operativos, mejora experiencia de clientes y se habilitan nuevos modelos de negocio digitales según Leitón (2021). Aplicando buenas prácticas en el prompt engineering y pruebas sugeridas por Puga (2022), se garantiza un incremento sostenible de calidad. Así, la precisión de ChatGPT en código fuente impulsa innovación en industrias digitales.

La correlación positiva media-alta ($r=0.776$) entre ChatGPT y eficiencia del código concuerda con conclusiones de Erazo et al. (2023) y Sidorov et al. (2020) sobre el potencial de optimización de la IA. Automatizando tareas rutinarias, se liberan recursos humanos para actividades de mayor valor según Gutiérrez (2021), con retornos en productividad como señaló Gómez et al. (2022). Considerando métricas ágiles para medir mejoras en eficiencia sugeridas por Hernández et al. (2020), el incremento demostrado habilita prácticas DevOps y modelos de negocio más ágiles.

Si bien investigadores como Bender et al. (2023) y Leitón (2021) resaltan las capacidades emergentes de la IA para generar código, los resultados de este estudio evidenciaron la necesidad de combinar ChatGPT con supervisión humana en casos complejos. Esta complementariedad concuerda con las recomendaciones de Sidorov et al. (2020) y Güemes et al. (2022) para garantizar precisión y minimizar riesgos. Implementando revisiones humanas sobre código generado por ChatGPT, se refuerza la confiabilidad incluso ante requerimientos cambiantes, siguiendo las buenas prácticas de desarrollo ágil sugeridas por Peña et al. (2021).

Si bien algunos autores como Hernández et al. (2020) enfatizan la importancia de métricas centradas en el usuario final al adoptar metodologías ágiles, este estudio se limitó a indicadores internos de complejidad, precisión y eficiencia. Incorporar

percepción de cliente podría complementar análisis sobre calidad e identificar brechas tempranas, como plantearon Govardhan (2010) y Boehm (1988) sobre la evaluación continua en modelos iterativos. Recomendaciones para prompts más creativos de EDEM (2023) también podrían capturar valor para usuarios.

Concordancia: El resultado general de esta investigación concuerda con lo señalado por Bender et al. (2023), quienes destacaron los avances significativos en la generación automática de código mediante modelos de lenguaje como GPT-3, resaltando su capacidad para crear texto de calidad similar a la redacción humana. Del mismo modo, en este estudio se evidenció la relación de ChatGPT en el desarrollo de software, generando código preciso y eficiente.

Discrepancia: Sin embargo, a contraposición de las conclusiones de Bender et al., esta investigación se enfocó de manera exclusiva en el entorno específico de historias de usuario en una consultora local, desentrañando descubrimientos estrechamente ligados a esta esfera particular. Mientras el estudio mencionado resalta avances generales en la generación de código mediante modelos de lenguaje como GPT-3, nuestro enfoque se ha centrado en ilustrar la relación directa de ChatGPT en la producción de código específico y eficiente dentro del contexto específico de historias de usuario dentro de una consultora local.

Acotación: Sería interesante ampliar el estudio a otros contextos empresariales para obtener una visión más integral sobre las aplicaciones de modelos como ChatGPT.

Concordancia: El resultado 1 respecto a la reducción de complejidad concuerda con Lagos (2023), cuyo sistema basado en reconocimiento de diagramas de flujo y generación de código ofreció una herramienta para simplificar el aprendizaje de programación. Del mismo modo, ChatGPT contribuyó a reducir la complejidad del código en la consultora.

Discrepancia: Sin embargo, a diferencia del enfoque educativo de Lagos, cuyo sistema se centró en simplificar el aprendizaje de programación a través del reconocimiento de diagramas de flujo y la generación de código, este estudio se adentró en el análisis de la relación de ChatGPT en una organización del sector tecnológico. Aquí, se reveló la utilidad de ChatGPT en un entorno empresarial

específico, destacando su capacidad para reducir la complejidad del código dentro de este contexto empresarial más especializado, en contraste con el propósito educativo enfocado por el sistema de Lagos.

Acotación: Se requieren más investigaciones sobre el uso de estas tecnologías de IA tanto en la academia como en la industria para optimizar sus implementaciones en diversos contextos.

Concordancia: El resultado 2 concuerda con lo planteado por Leitón (2021) sobre el uso de traducción de lenguajes de programación a uno universal para verificar coherencia sintáctica, al igual que ChatGPT incrementó la precisión del código en la consultora analizada.

Discrepancia: Sin embargo, a diferencia del enfoque de Leitón que se concentró en la traducción específica entre Java, C#, y RPG, esta investigación no se centró en lenguajes particulares. En cambio, se exploró la mejora general de la precisión del código mediante ChatGPT en un contexto más amplio, sin limitarse a la traducción entre idiomas de programación específicos. Mientras Leitón abordó la coherencia sintáctica entre lenguajes específicos, nuestro estudio buscó evaluar la mejora general de la precisión del código mediante la utilización de ChatGPT, sin restricciones a lenguajes particulares.

Acotación: Se podrían realizar estudios comparando la precisión en distintos lenguajes al usar ChatGPT, para identificar posibles diferencias en su desempeño según el contexto.

Concordancia: El resultado 3 concuerda con lo indicado por Zapata et al. (2020) sobre la contribución del juego Smellware para educar a desarrolladores en buenas prácticas y comprensión de problemas potenciales en el código, al igual que ChatGPT optimizó la eficiencia en la consultora estudiada.

Discrepancia: Sin embargo, a diferencia del enfoque lúdico y educativo de Smellware, diseñado para fortalecer habilidades a través de un juego, este estudio se sumergió en el análisis de métricas de eficiencia en un contexto empresarial real, alejado del ámbito educativo. Mientras Smellware se enfoca en educar a desarrolladores y promover buenas prácticas mediante un enfoque lúdico, mi

estudio se centró en evaluar la optimización de la eficiencia de manera práctica y aplicada en el entorno empresarial, mostrando cómo ChatGPT influyó en la mejora de la eficiencia dentro de esta esfera específica.

Acotación: Se podrían desarrollar investigaciones comparando la eficiencia de desarrolladores capacitados con ChatGPT y con enfoques educativos como Smellware u otros, para conocer posibles diferencias.

Concordancia: El resultado general concuerda con Echavarría et al. (2020) respecto a la relación positiva de la IA en la toma de decisiones y eficiencia en el sector financiero, de modo similar a cómo ChatGPT incrementó la eficiencia en el desarrollo de software de la consultora.

Discrepancia: Sin embargo, a diferencia del enfoque específico de Echavarría et al., en decisiones de inversión dentro del sector financiero, este estudio se sumergió en el análisis del código fuente de software en una organización tecnológica, con un propósito y alcance distintos. Mientras Echavarría et al., destacaron el impacto positivo de la IA en la toma de decisiones financieras, nuestro enfoque se centró en evaluar cómo ChatGPT contribuyó a la eficiencia en el desarrollo de software, revelando mejoras en la producción de código en un contexto empresarial tecnológico específico, un ámbito diferente al abordado por Echavarría et al..

Acotación: Por ello, es recomendable realizar más investigaciones sobre los efectos positivos de la IA tanto en finanzas como en tecnología, para comprender a fondo sus aplicaciones en diversas industrias.

Concordancia: El resultado 1 concuerda con Tamayo et al. (2021) en cuanto a la contribución de la gestión de mantenibilidad desde etapas iniciales a la calidad del software y reducción de complejidad. Del mismo modo, ChatGPT redujo complejidad del código en la consultora estudiada.

Discrepancia: Sin embargo, a diferencia del enfoque específico de Tamayo et al., en un proceso particular para la gestión temprana de mantenibilidad, nuestro estudio se centró en el uso de una herramienta de IA como ChatGPT para abordar la reducción de complejidad del código en la consultora estudiada. Mientras Tamayo et al., propusieron un proceso concreto para la gestión temprana de mantenibilidad, nuestro enfoque se apoyó en la implementación de una herramienta

innovadora de IA, demostrando cómo ChatGPT influyó directamente en la reducción de la complejidad del código en un contexto empresarial tecnológico, mostrando un enfoque distinto para abordar este desafío.

Acotación: Se requieren más investigaciones comparando estrategias tradicionales de gestión de mantenibilidad frente al apoyo de tecnologías de IA en esta labor, para conocer posibles ventajas y desventajas.

Concordancia: El resultado 2 concuerda con OWASP Foundation (2020) al señalar la importancia de la seguridad en el código a través de identificación y corrección de vulnerabilidades, así como con este estudio sobre la mayor precisión brindada por ChatGPT.

Discrepancia: Sin embargo, a diferencia del enfoque exclusivo de OWASP en aspectos de seguridad informática y la identificación de vulnerabilidades en el código, esta investigación adoptó un enfoque más amplio. Se centró en evaluar la precisión general del código mediante pruebas, abordando la mejora de la precisión a través de ChatGPT en un espectro más amplio que no se limita únicamente a la seguridad. Mientras que OWASP se concentra en la identificación y corrección de vulnerabilidades para garantizar la seguridad del código, nuestro estudio abarcó una evaluación general de la precisión del código, explorando cómo ChatGPT influyó en esa precisión en diversas áreas, no restringiéndose únicamente a la seguridad informática.

Acotación: Sería valorable realizar estudios sobre la contribución de ChatGPT a mejorar parámetros de seguridad en el desarrollo de software, para conocer su efectividad en ese aspecto en particular.

VI. CONCLUSIONES

Se determinó la relación de ChatGPT en el desarrollo de código fuente de software, ya que las variables no se distribuyen normalmente, se usó la prueba no paramétrica de Rho de Spearman para evaluar la correlación entre ChatGPT y el Código fuente de software. De acuerdo a los resultados, como todos los Sig fueron menores a 0.05, se rechazó la hipótesis nula y se concluyó en que sí existe una correlación positiva significativa alta. Es decir, ChatGPT tiende a mejorar el desarrollo de código fuente de software.

Se determinó la relación de ChatGPT en la complejidad del desarrollo de código fuente de software, incrementando su legibilidad, mantenibilidad y organización, con una correlación positiva media-alta de 0.892 entre ChatGPT y la dimensión complejidad.

Se determinó la relación de ChatGPT en la precisión del desarrollo de código fuente de software, aumentando el porcentaje de pruebas aprobadas y disminuyendo errores, con una correlación positiva alta de 0.813 entre la herramienta de IA y la dimensión precisión.

Se determinó la relación de ChatGPT en la eficiencia del desarrollo de código fuente de software, mejorando el uso de recursos y rendimiento, con una correlación positiva media-alta de 0.776 entre ChatGPT y la dimensión eficiencia.

VII. RECOMENDACIONES

Primera. Se recomienda al jefe o líder de Desarrollo de software, adoptar ChatGPT en los procesos de desarrollo de software de las consultoras, configurando prompts especializados por lenguaje de programación para maximizar su rendimiento.

Segunda. Se recomienda al jefe de Recursos humanos, capacitar al personal en el uso adecuado de ChatGPT para el desarrollo de código, estableciendo guías sobre cómo aprovechar sus capacidades y evitar sus limitaciones.

Tercera. Se recomienda al jefe de Control de calidad, complementar el uso de ChatGPT con otras herramientas de inteligencia artificial y revisiones humanas para garantizar la calidad del software.

Cuarta. Se recomienda al jefe de Investigación y desarrollo, realizar investigaciones que analicen la relación de ChatGPT en diversos lenguajes y metodologías de programación para optimizar su implementación.

Quinta. Se recomienda al jefe de Gestión de proyectos, explorar la integración de ChatGPT con plataformas de gestión de proyectos y control de versiones para potenciar su valor en el ciclo de vida del software.

REFERENCIAS

- Armetrics. (2022). *Qué es Código fuente*. <https://www.armetrics.com/glosario-digital/codigo-fuente>
- Bender, A., Nicolet, S., Folino, P., Lopez, J., & Hansen, G. (2023). Generación Automática de Código Fuente a través de Modelos Preentrenados de Lenguaje. *Electronic Journal of SADIO (EJS)*, 22(1), 19-36.
<https://doi.org/https://ojs.sadio.org.ar/index.php/EJS/article/view/465>
- Boehm, B. (1988). A spiral model of software development and enhancement. En *IEEE. Transactions on Software Engineering*.
- Castro, L. (2023). *Code analyzer: herramienta de análisis de código para repositoriosg git*. <http://hdl.handle.net/1992/69769>
- CPU-e. (2011). Factores que afectan el desempeño académico de los estudiantes de nivel superior en Rioverde, San Luis Potosí, México. *Revista de Investigación Educativa*(12).
<https://doi.org/https://www.uv.mx/cpue/num12/opinion/completos/izar-desempeno%20academico.html>
- Dulzaides, M., & Molina, A. (2004). Análisis documental y de información: dos componentes de un mismo proceso. *SciELO*, 12(2).
https://doi.org/http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352004000200011
- Echavarría, I., & Restrepo, F. (2020). Software Readability Metrics: A Systematic Literature Review. *Revista Facultad de Ingeniería*, 29(54).
<https://doi.org/10.19053/01211129.v29.n54.2020.11756>
- Echevarría, W., Manso, W., González, J., & Fragoso, J. (2022). Simulation of absorbed dose, in molecular therapy, by means of Monte Carlo method using free software GEANT4: GATE. *SciELO*, 20(2).
<https://doi.org/https://acortar.link/gk9MOV>
- EDEM. (2023). *CHAT GPT: QUÉ ES, PARA QUÉ SIRVE Y SU APLICACIÓN EN LA ECONOMÍA [EXPLICADO POR CHAT GPT]*. <https://edem.eu/chat-gpt->

que-es-para-que-sirve-y-su-aplicacion-en-la-economia-explicado-por-chat-gpt/

El Comercio. (2023). *Chat GPT e inteligencia artificial en la educación: ¿héroe o amenaza?* Encuesta a estudiantes del colegio Sor Ana de los Ángeles (Callao): <https://acortar.link/3pQLQ4>

Erazo, A., Ramos, F., Galarza, P., & Boné, M. (2023). La inteligencia artificial aplicada a la optimización de programas informáticos. *Journal of Economic and Social Science Research*, 3(1), 48-63.
<https://doi.org/10.55813/gaea/jessr/v3/n1/61>

Fritz, E., Pérez, Á., Tirado, J., Vásquez, L., Marcelín, R., Rodríguez, E., & Pascoe, M. (2022). Discrete formulation of envelope correlation coefficient for faster analysis in MIMO antenna systems. *SciELO*, 23(4).
<https://doi.org/10.22201/fi.25940732e.2022.23.4.028>

García, A., Pineda, D., & Andrade, M. (2015). Las capacidades tecnológicas para la innovación en empresas de manufactura. *Universidad & Empresa*, 17(29). <https://doi.org/10.12804/rev.univ.empresa.29.2015.11>

García, M., García, H., & González, A. (2019). Selection of a Hotel System of Management of Open Code. *SciELO*, 19(2).
<https://doi.org/https://acortar.link/EGX9MI>

Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2002). *Fundamentals of Software Engineering*. <https://www.pearson.com/us/higher-education/program/Ghezzi-Fundamentals-of-Software-Engineering-2nd-Edition/PGM321803.html>

Gómez, S., Moreno, J., & Zapata, C. (2022). Adaptación de herramientas web para la implementación de un curso masivo colaborativo de desarrollo de software. *SciELO*, 33(5). <https://doi.org/10.4067/S0718-07642022000500145>

González, A., Navas, J., Hernández, M., Hernández, F., & Solano, J. (2019). A visual analytics architecture for the analysis and understanding of software systems. *SciELO*, 10(1). <https://doi.org/10.29019/enfoqueute.v10n1.455>

- Govardhan, A. (2010). A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science Issues*.
- Güemes, A., Delgado, M., Fernández, P., & Henry, H. (2022). Extensiones de Mtest.search para la generación de código de prueba. *ScieELO*, 43(1).
- Guitérrez, B., Vázquez, J., & López, X. (2023). AI application in journalism: ChatGPT and the uses and risks of an emergent technology. *Profesional De La información*, 32(5). <https://doi.org/10.3145/epi.2023.sep.14>
- Gutiérrez, M. (2021). Access to source code and algorithms held by public authorities. Dismissing risks of a black box decision-like. *Dialnet*, 1(31). <https://doi.org/https://dialnet.unirioja.es/servlet/articulo?codigo=8193102>
- Gutiérrez, O., Delgado, F., Meza, J., Turpo, O., & Ticona, F. (2023). Predictors of Academic Performance through the Use Of Chatgpt in University Students. *Human Review*, 21(2), 411-421. <https://doi.org/10.37467/revhuman.v21.5077>
- Hernández, G., Martínez, Á., Jiménez, R., & Jiménez, F. (2020). Cómo los profesionales perciben la relevancia de las métricas de productividad para un equipo ágil de desarrollo de software. *RISTI*(32), 596–609.
- Hidalgo, C. (2021). Estrategia de enseñanza basada en la colaboración y la evaluación automática de código fuente en un curso de programación CS1. *Dialnet*, 9(1), 50-60. <https://doi.org/https://dialnet.unirioja.es/servlet/articulo?codigo=7799072>
- HubSpot. (2023). *ChatGPT: qué es, cómo usarlo, ventajas y ejemplos*. Buenas prácticas para usar ChatGPT: <https://blog.hubspot.es/website/que-es-chatgpt>
- Lagos, J. (2023). *Diseño e implementación de un generador de código fuente a partir de diagramas de flujo manuscritos*. Universitat Oberta de Catalunya (UOC): <http://hdl.handle.net/10609/148861>

- Leitón, J. (2021). *Método para convertir código fuente escrito en diversos lenguajes de programación a un lenguaje universal*.
<https://hdl.handle.net/2238/13420>
- Leiton, J. (2023). GAST: A Generic AST Representation for Language-Independent Source Code Analysis. *SciELO*, 14(4).
<https://doi.org/10.29019/enfoqueute.957>
- Lifeder. (2022). *Ficha de análisis*. <https://www.lifeder.com/ficha-analisis/>
- López, P. (2004). POBLACIÓN MUESTRA Y MUESTREO. *SciELO*, 9(8).
<https://doi.org/https://acortar.link/AKNthO>
- Lozada, J. (2014). Investigación Aplicada. *Dialnet*, 3(1), 47-50.
<https://doi.org/https://dialnet.unirioja.es/servlet/articulo?codigo=6163749>
- Morcela, O. (2023). ChatGPT: la IA está aquí y nos desafía. *Revista Internacional De Ingeniería Industrial*(6).
- Muguirra, A. (2023). *Tipos de muestreo: Cuáles son y en qué consisten*.
QuestionPro: <https://www.questionpro.com/blog/es/tipos-de-muestreo-para-investigaciones-sociales/>
- Narvaez, M. (2020). *¿Qué es la validez y confiabilidad en la investigación?*
QuestionPro: <https://acortar.link/9yqIWw>
- O'Regan, G. (2016). *Introduction to Software Quality*. Springer.
- OWASP Foundation. (2020). *Top 10 Web Application Security Risks*.
<https://owasp.org/www-project-top-ten/>
- Peña, D., & Carbonell, A. (2021). Herramienta de soporte para la inspección de la calidad del código de las aplicaciones HPC. *Ciencia Y Educación*, 2(3), 6-14.
<https://doi.org/https://cienciayeducacion.com/index.php/journal/article/view/101>
- Pérez, L., Acosta, A., Rodríguez, A., & Rodríguez, L. (2022). Design of a home automation system based on open-source platforms. *SciELO*, 43(2).
<https://doi.org/https://acortar.link/PHINIf>

- Puga, R. (2022). *Exploración de la optimalidad de señales no-visuales de código fuente en ambientes de desarrollo integrado para programadores ciegos*. <https://repositorio.uchile.cl/handle/2250/185974>
- QuestionPro. (2020). *Investigación no experimental: Qué es, características, ventajas y ejemplos*. <https://acortar.link/tU7S4p>
- Quispe, C., Kirschbaum, J., Caceres, J., & Ugas, C. (2023). Explorando la viabilidad de la integración de la inteligencia artificial en la atención médica pediátrica: Un estudio preliminar con ChatGPT. *Investigación E Innovación Clínica Y Quirúrgica Pediátrica*, 1(1), 14-23. <https://doi.org/10.59594/iicqp.2023.v1n1.5>
- Rus, E. (2021). *Economipedia*. <https://economipedia.com/definiciones/investigacion-cuantitativa.html>
- Sidorov, G., Ibarra, M., & Markov, I. (2020). Automatic Detection of Similarity of Programs in Karel Programming Language based on Natural Language Processing Techniques. *SciELO*, 20(2). <https://doi.org/doi.org/10.13053/cys-20-2-2369>
- Significados. (2023). *Significado de Complejidad*. <https://www.significados.com/complejidad/>
- Stallman, R. (2020). The definition of Free Software. *Revista de Imagen, Artes y Educación Crítica y Social*, 1(3), 151-154. <https://doi.org/https://hdl.handle.net/11441/100711>
- Tamayo, L., & Silega, N. (2021). Application of a process for the management of maintainability in software development. *SciELO*, 15(4). <https://doi.org/https://acortar.link/0gzqBv>
- Universidad Veracruzana. (2020). *La muestra*. <https://www.uv.mx/apps/bdh/investigacion/unidad3/muestra.html>
- Vega, A. (2021). Estudios transversales. *SciELO*, 21(1). <https://doi.org/10.25176/rfmh.v21i1.3069>

Yagmour, J. (2023). Narratives of Digital Encodings: Verticality and Speed as Forms of Temporality. *SciELO*, 38(1). <https://doi.org/10.4067/s0718-23762023000100029>

Zapata, C., Gómez, M., & Hernández, J. (2020). SMELLWARE: un juego para la enseñanza de buenas prácticas en el proceso de desarrollo de software. *SciELO*, 28(4). <https://doi.org/10.4067/S0718-33052020000400645>

ANEXOS

Anexo 1 - Matriz de consistencia de la investigación

Título: ChatGPT desarrollando código fuente de software para historias de usuarios en una consultora, Lima, 2023

Autor: Altamirano Tseng, Leenkim Hahnemann

Problema	Objetivo	Hipótesis	Variable
<p>General: ¿ChatGPT se relaciona con el desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023?</p>	<p>General: Determinar la relación de ChatGPT en el desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.</p>	<p>General: ChatGPT se relaciona significativamente con el desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.</p>	<p>Independiente: ChatGPT</p>
<p>Específicos:</p> <ol style="list-style-type: none"> 1. ¿ChatGPT se relaciona con la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023? 2. ¿ChatGPT se relaciona con la precisión del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023? 3. ¿ChatGPT se relaciona con la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023? 	<p>Específicos:</p> <ol style="list-style-type: none"> 1. Determinar la relación de ChatGPT en la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. 2. Determinar la relación de ChatGPT en la precisión del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. 3. Determinar la relación de ChatGPT en la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023. 	<p>Específicas:</p> <p>He1: ChatGPT se relaciona significativamente con la complejidad del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.</p> <p>He2: ChatGPT se relaciona significativamente con la precisión del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.</p> <p>He3: ChatGPT se relaciona significativamente con la eficiencia del desarrollo de código fuente de software para historias de usuarios en una consultora, Lima, 2023.</p>	<p>Dependiente: Código de fuente de software</p>

Metodología

Tipo de investigación:

Aplicada

Enfoque de investigación:

Cuantitativo

Población (N):

La consultora tiene alrededor de 180 historias de usuario al mes.

Técnicas de recolección de datos:

- Análisis documental

Método de análisis de datos:

1. Análisis descriptivo de los datos recolectados a partir de la ficha de observación.

Diseño de investigación:

No experimental, transversal, descriptivo, correlacional

Muestra (n):

$$\text{Tamaño de la muestra} = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + \left(\frac{z^2 \times p(1-p)}{e^2 N}\right)}$$

N = 180 / e = 0.05

p = 0.95 / z = 1.96

n = 124

Instrumentos:

- Ficha de análisis

Ética:

Se respeta la RV de la UCV.

Asimismo, se tomará en cuenta el Sistema de Normas APA 7ma Edición.

Anexo 2 - Matriz de operacionalización de variables

Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicador	Escala de medición
Independiente: ChatGPT	Inteligencia artificial desarrollada por OpenAI, basada en la arquitectura GPT (Transformador Generativo Preentrenado), diseñada para comprender y generar texto de manera autónoma en respuesta a consultas o interacciones humanas. Utiliza un extenso conjunto de datos para aprender patrones lingüísticos, generar respuestas coherentes y relevantes en diversos ámbitos, resolviendo problemas hasta generando contenido creativo. (EDEM, 2023)	Se refiere a la presencia y funcionamiento del modelo de lenguaje de inteligencia artificial desarrollado por OpenAI conocido como "ChatGPT". Esta variable se considera presente cuando el sistema ChatGPT está activo y disponible para interactuar con usuarios a través de texto o voz.	Datos de entrenamiento	a. Promedio de líneas de código	Intervalo
			Calidad de código	b. Promedio de errores	
			Dominio de problema	c. Promedio de criterios de aceptación	
Dependiente: Código fuente de software	Se refiere a una colección de instrucciones escritas en un lenguaje de programación legible por humanos, generalmente en formato de texto sin formato, que se utiliza para facilitar el trabajo de los programadores de computadoras. (Armetrics, 2021)	Se refiere al conjunto de instrucciones escritas en un lenguaje de programación que define el funcionamiento y comportamiento de un programa o aplicación informática.	Complejidad	a. Longitud del código	
			Precisión	b. Porcentajes de pruebas aprobadas	
			Eficiencia	c. Promedio de uso de recursos	

Anexo 3: Ficha de instrumentos

I. Información General:

Título del Proyecto de Investigación: ChatGPT desarrollando código fuente de software para historias de usuarios en una consultora, Lima, 2023

Investigador Principal: Altamirano Tseng, Leenkim Hahnemann

Fecha de Creación de la Ficha: 6 de octubre del 2023

II. Descripción de la Variable Dependiente:

Variable Dependiente: Código fuente de software

III. Dimensiones e Indicadores de la Variable Dependiente:

Dimensión 1: Complejidad

Indicador 1: Longitud del código (Escala en Intervalo)

Dimensión 2: Precisión

Indicador 2: Porcentajes de pruebas aprobadas (Escala en Intervalo)

Dimensión 3: Eficiencia

Indicador 3: Promedio de uso de recursos (Escala en Intervalo)

IV. Preguntas de Análisis:

1. ¿Cuál es la longitud promedio de líneas de código por función para la historia de usuario? (medir en caracteres)
 - a. Menos de 50 líneas de código
 - b. 50 a 100 líneas de código
 - c. 101 a 150 líneas de código
 - d. Más de 150 líneas de código
2. ¿Cómo describirías la estructura general del código en términos de complejidad?
 - a. El código es altamente estructurado, sigue principios de diseño sólidos y es fácil de entender
 - b. El código tiene cierta complejidad, pero sigue buenas prácticas de codificación y es manejable
 - c. El código es moderadamente complejo y podría beneficiarse de una mayor organización y documentación
 - d. El código es bastante complejo y requiere mejoras significativas en su organización y claridad

3. ¿Se han identificado áreas de código particularmente extensas que puedan aumentar la complejidad?
 - a. Menos de 3 funciones
 - b. 3 a 6 funciones
 - c. 7 a 10 funciones
 - d. Más de 10 funciones
4. En términos de la estructura del código, ¿cómo describirías la relación entre funciones o métodos y su complejidad?
 - a. Las funciones/métodos son cortos y altamente legibles
 - b. Las funciones/métodos son de longitud moderada y en su mayoría comprensibles
 - c. Algunas funciones/métodos son largos y complejos, pero la mayoría son manejables
 - d. La mayoría de las funciones/métodos son largos y complejos, dificultando la comprensión
5. ¿Cuántas pruebas funcionales se han realizado en total para la historia de usuario?
 - a. Más de 10 pruebas
 - b. 6 – 10 pruebas
 - c. 3 - 5 pruebas
 - d. Menos de 3 pruebas
6. ¿Cuál es el porcentaje de pruebas funcionales aprobadas en relación con el total de pruebas realizadas?
 - a. 90% - 100%
 - b. 70% - 89%
 - c. 50% - 69%
 - d. Menos del 50%
7. ¿Cuántas pruebas unitarias (testing) se han realizado en total para la historia de usuario?
 - a. Más de 10 pruebas
 - b. 6 – 10 pruebas
 - c. 3 - 5 pruebas
 - d. Menos de 3 pruebas

8. ¿Cuál es el porcentaje de pruebas unitarias aprobadas en relación con el total de pruebas realizadas?
 - a. 90% a 100%
 - b. 70% a 89%
 - c. 50% a 69%
 - d. Menos del 50%
9. ¿Se han llevado a cabo pruebas de carga y rendimiento para simular situaciones de uso intensivo?
 - a. Más de 10 pruebas
 - b. 6 – 10 pruebas
 - c. 3 - 5 pruebas
 - d. Menos de 3 pruebas
10. ¿Cómo se comporta el desarrollo de la historia de usuario en términos de tiempo de respuesta y rendimiento general?
 - a. Excelente
 - b. Bueno
 - c. Regular
 - d. Deficiente
11. ¿Se identificaron cuellos de botella en las funciones desarrolladas para la historia de usuario?
 - a. Menos de 3 funciones
 - b. 3 a 6 funciones
 - c. 7 a 10 funciones
 - d. Más de 10 funciones

Anexo 4: Recolección de datos

N°	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
1	3	2	3	1	1	3	3	4	2	2	3
2	2	2	1	1	2	1	2	2	1	3	4
3	4	1	3	4	3	3	2	3	2	1	2
4	3	2	2	2	1	2	2	2	3	1	1
5	4	2	1	2	3	4	2	2	1	3	3
6	4	2	2	3	2	2	4	2	2	1	4
7	1	2	2	1	2	3	2	4	1	1	2
8	1	3	1	2	3	4	3	4	2	2	1
9	2	1	4	2	3	4	1	4	2	3	3
10	4	2	3	3	2	4	3	3	3	1	4
11	4	3	4	4	2	4	3	4	4	3	1
12	2	1	3	2	4	3	4	4	3	2	4
13	4	4	4	3	4	4	4	4	3	4	2
14	3	4	2	3	4	2	3	4	2	3	3
15	4	4	4	4	4	4	4	4	4	4	1
16	1	2	1	3	2	1	3	2	1	2	4
17	3	4	4	4	3	4	4	3	4	4	2
18	4	3	4	4	4	4	3	4	4	4	3
19	2	1	2	3	2	1	2	3	2	3	1
20	4	4	4	4	4	4	4	4	4	4	4
21	3	4	2	3	4	2	3	4	2	3	2
22	4	4	4	3	4	4	4	3	4	4	3
23	2	1	2	3	2	1	2	3	2	1	1
24	4	3	4	4	3	4	4	3	4	4	4
25	1	2	1	3	2	1	3	2	1	2	2
26	3	4	2	3	4	2	3	4	2	3	3
27	4	4	4	3	4	4	4	3	4	4	1
28	2	1	2	3	2	1	2	3	2	1	4
29	4	3	4	4	3	4	4	3	4	4	1
30	1	2	1	3	2	1	3	2	1	2	2
31	4	4	4	3	4	4	4	3	4	4	3
32	3	2	1	2	3	2	1	2	3	2	4
33	4	3	4	4	3	4	4	3	4	4	1
34	2	1	2	3	2	1	2	3	2	1	2
35	3	4	2	3	4	2	3	4	2	3	3

36	1	2	1	3	2	1	3	2	1	2	4
37	4	3	4	4	3	4	4	3	4	4	1
38	4	4	4	3	4	4	4	3	4	4	2
39	2	1	2	3	2	1	2	3	2	1	2
40	4	3	4	4	3	4	4	3	4	4	3
41	3	3	4	4	4	4	3	4	4	4	1
42	2	2	3	3	4	4	2	3	4	3	4
43	4	4	4	4	4	4	4	4	4	4	2
44	3	3	4	4	3	3	4	4	3	4	3
45	1	1	2	2	3	3	1	2	3	2	1
46	4	4	3	3	2	2	4	3	2	3	4
47	4	4	4	4	3	3	4	4	3	4	3
48	2	2	3	3	4	4	2	3	4	3	2
49	3	3	4	4	4	4	3	4	4	4	1
50	4	4	3	3	2	2	4	3	2	3	1
51	4	4	4	3	4	4	3	4	4	4	4
52	2	3	2	1	3	2	1	3	2	3	2
53	4	4	4	4	4	4	4	4	4	4	3
54	3	2	3	4	2	3	4	2	3	2	1
55	1	2	1	3	2	1	3	2	3	2	4
56	4	3	4	2	3	4	2	3	4	3	2
57	4	4	4	3	4	4	3	4	4	4	2
58	2	1	2	3	1	2	3	1	2	1	2
59	3	4	3	4	4	3	4	4	4	4	3
60	4	4	4	3	4	4	3	4	4	4	4
61	3	4	3	2	4	3	2	4	3	4	1
62	4	4	4	3	4	4	3	4	4	4	1
63	2	3	2	1	3	2	1	3	2	3	3
64	4	3	4	2	3	4	2	3	4	3	2
65	1	1	2	3	2	1	3	2	1	2	1
66	4	4	3	2	4	3	2	4	3	4	3
67	4	4	4	3	4	4	3	4	4	4	2
68	2	2	3	1	2	3	1	2	3	2	2
69	3	3	4	4	4	3	4	4	3	4	2
70	4	4	3	2	4	3	2	4	3	4	2
71	4	3	4	3	4	3	4	3	4	3	3
72	2	1	2	1	2	1	2	1	2	1	4
73	4	4	4	4	4	4	4	4	4	4	1
74	3	2	3	2	3	2	3	2	3	2	2

75	1	2	1	2	1	2	1	2	1	2	2
76	4	3	4	3	4	3	4	3	4	3	2
77	4	4	4	4	4	4	4	4	4	4	1
78	2	1	2	1	2	1	2	1	2	1	1
79	3	4	3	4	3	4	3	4	3	4	2
80	4	4	4	4	4	4	4	4	4	4	3
81	3	4	3	4	3	4	3	4	3	4	1
82	1	2	1	2	1	2	1	2	1	2	2
83	4	4	4	4	4	4	4	4	4	4	4
84	2	3	2	3	2	3	2	3	2	3	3
85	1	1	2	2	1	1	2	2	1	2	1
86	4	4	3	3	4	4	3	3	4	3	2
87	4	4	4	4	4	4	4	4	4	4	4
88	1	1	2	2	1	1	2	2	1	2	3
89	3	3	4	4	3	3	4	4	3	4	1
90	4	4	4	4	4	4	4	4	4	4	2
91	4	3	4	3	4	3	4	3	4	3	3
92	2	1	2	1	2	1	2	1	2	1	1
93	4	4	4	4	4	4	4	4	4	4	4
94	3	2	3	2	3	2	3	2	3	2	2
95	1	2	1	2	1	2	1	2	1	2	3
96	4	3	4	3	4	3	4	3	4	3	1
97	4	4	4	4	4	4	4	4	4	4	4
98	1	2	1	2	1	2	1	2	1	2	2
99	3	4	3	4	3	4	3	4	3	4	3
100	2	3	4	1	2	1	2	3	4	3	3
101	4	3	3	2	2	3	2	2	2	4	3
102	2	3	1	3	1	1	3	2	3	3	1
103	4	3	1	1	1	2	3	1	1	3	2
104	3	4	3	1	1	3	4	3	1	4	1
105	4	2	3	2	3	2	3	1	1	4	2
106	1	2	2	4	2	3	3	2	4	2	1
107	1	1	2	4	2	1	2	2	3	4	3
108	2	3	4	3	3	4	1	4	4	1	4
109	3	2	1	3	4	1	1	3	3	2	3
110	3	1	3	3	2	1	2	4	2	3	4
111	1	3	2	1	2	1	2	3	4	3	1
112	1	2	3	3	3	2	2	3	2	2	1
113	2	2	2	4	3	2	2	4	4	3	2

114	1	1	4	1	4	1	1	4	1	2	4
115	4	1	4	3	4	1	4	2	4	1	3
116	4	3	2	2	2	1	2	4	4	1	4
117	4	1	2	1	4	3	3	3	3	2	3
118	4	3	2	4	2	1	4	1	2	4	1
119	4	2	3	1	3	3	3	3	1	3	1
120	2	2	4	4	3	4	3	2	3	2	3
121	3	1	3	4	3	4	3	3	2	4	4
122	4	1	3	3	3	3	3	1	3	4	1
123	3	4	4	3	4	2	1	2	2	1	1
124	1	2	1	1	4	2	2	1	4	4	4

Anexo 5: Carta de presentación

CARTA DE PRESENTACIÓN

Señor: SÁENZ APARI ABRAHAM RAFAEL

Presente

Asunto: **VALIDACIÓN DE INSTRUMENTOS A TRAVÉS DE JUICIO DE EXPERTO.**

Me es muy grato comunicarme con usted para expresarle mis saludos y así mismo, hacer de su conocimiento que, siendo estudiante del Programa de Maestría en Ingeniería de Sistemas con mención en Tecnologías de la Información de la Escuela de Posgrado de la UCV, en la sede LIMA NORTE, requiero validar los instrumentos con los cuales se recogerá la información necesaria para poder desarrollar mi investigación y con la sustentaré mis competencias investigativas en la Experiencia curricular de Diseño y desarrollo del trabajo de investigación.

El nombre de mis Variables es: ChatGPT y Código de fuente de software; siendo imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos en mención, se ha considerado conveniente recurrir a usted, ante su connotada experiencia en temas educativos y/o investigación educativa.

El expediente de validación, que le hacemos llegar contiene:

- Carta de presentación.
- Formato de Validación.
- Certificado de validez de contenido de los instrumentos.

Expresándole mis sentimientos de respeto y consideración me despido de usted, no sin antes agradecerle por la atención que dispense a la presente.

Atentamente.



.....
ALTAMIRANO TSENG, LEENKIM HAHNEMANN

D.N.I 73179995

Evaluación por juicio de expertos

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento para medir las Variables ChatGPT y Código de fuente de software. La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando al quehacer psicológico. Agradecemos su valiosa colaboración.

1. Datos generales del juez:

Nombre del juez:	Abraham Rafael Sáenz Apari
Grado profesional:	Maestría (x) Doctor ()
Área de formación académica:	Clínica () Social () Educativa (x) Organizacional ()
Áreas de experiencia profesional:	Educación
Institución donde labora:	Universidad Privada César Vallejo
Tiempo de experiencia profesional en el área:	2 a 5 años (x) Más de 5 años ()
Experiencia en Investigación (si corresponde)	Asesor de Tesis en Pregrado y Posgrado

2. Propósito de la evaluación:

Validar el contenido del instrumento, por juicio de expertos.

3. Datos de la escala:

Nombre de la Prueba:	Instrumento para medir la variable Código de fuente de software
Autor:	ALTAMIRANO TSENG, LEENKIM HAHNEMANN
Procedencia:	Ficha de análisis
Administración:	Directa
Tiempo de aplicación:	40 minutos
Ámbito de aplicación:	Entidad consultora
Significación:	La ficha de análisis de la Variable Código de fuente de software está compuesto de 3 dimensiones que son: Complejidad, Precisión y Eficiencia. El objetivo de esta medición es establecerlas relaciones entre ChatGPT y el Código de fuente de software.

4. Soporte teórico:

ChatGPT: Se trata de la herramienta que ha generado más discusión y ha captado la atención por todos los medios como la red social entre otros. Chat GPT marcó un punto que sobresale mundialmente dentro de la IA. (EDEM, 2023)

Código fuente de software: Se refiere a una colección de instrucciones escritas en un lenguaje de programación legible por humanos, generalmente en formato de texto sin formato, que se utiliza para facilitar el trabajo de los programadores de computadoras. (Armetrics, 2021)

Escala/ÁREA	Subescala (dimensiones)	Definición
INTERVALO	Complejidad	Se puede definir como la característica de algo que es complicado en su estructura. En otras palabras, el término "complejidad" se refiere a la presencia de múltiples elementos que están interconectados entre sí. Esta palabra se origina en el término "complejo", que tiene sus raíces en el latín <i>complexus</i> , el participio pasado de <i>complexi</i> , que significa "enlazar". (Significados, 2023)
	Precisión	Estamos hablando de su habilidad para acertar en el corto plazo, es decir, lograr resultados que se ajusten a las expectativas o que estén muy cerca de lo esperado. Aunque en el lenguaje común a menudo se utiliza indistintamente con "exactitud", es importante no mezclar o confundir estos dos conceptos. (Concepto, 2022)
	Eficiencia	Se relaciona con el uso sensato de los recursos disponibles para lograr un objetivo establecido previamente. Implica la habilidad de alcanzar una meta predefinida en el menor tiempo y con la menor cantidad de recursos necesarios, lo que implica una optimización. (Definición, 2022)

5. Presentación de instrucciones para el juez:

A continuación, a usted le presento la ficha de análisis para medir la Variable Código de fuente de software elaborado por Altamirano Tseng, Leenkim Hahnemann. De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

Categoría	Calificación	Indicador
CLARIDAD El ítem se comprende fácilmente, es decir, su sintácticay semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de estas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. totalmente en desacuerdo (no cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencialmente importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

Leer con detenimiento los ítems y calificar en una escala de 1 a 4 su valoración, así como solicitamos brindes sus observaciones que considere pertinente

1 No cumple con el criterio
2. Bajo Nivel
3. Moderado nivel
4. Alto nivel

Dimensiones del instrumento:

- **Primera dimensión:** Complejidad
- **Indicador 1:** Longitud del código (Escala en Intervalo)

Indicador	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Longitud del código	1. ¿Cuál es la longitud promedio de líneas de código por función para la historia de usuario?	3	3	3	
	2. ¿Cómo describirías la estructura general del código en términos de complejidad?	3	3	3	
	3. ¿Se han identificado áreas de código particularmente extensas que puedan aumentar la complejidad?	3	3	3	
	4. En términos de la estructura del código, ¿cómo describirías la relación entre funciones o métodos y su complejidad?	3	3	3	

- **Segunda dimensión:** Precisión
- **Indicador 2:** Porcentaje de pruebas aprobadas (Escala en Intervalo)

Indicador	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Porcentaje de pruebas aprobadas	5. ¿Cuántas pruebas funcionales se han realizado en total para la historia de usuario?	3	3	3	
	6. ¿Cuál es el porcentaje de pruebas funcionales aprobadas en relación con el total de pruebas realizadas?	3	3	3	
	7. ¿Cuántas pruebas unitarias (testing) se han realizado en total para la historia de usuario?	3	3	3	
	8. ¿Cuál es el porcentaje de pruebas unitarias aprobadas en relación con el total de pruebas realizadas?	3	3	3	

- **Tercera dimensión:** Eficiencia
- **Indicador 3:** Promedio de uso de recursos (Escala en Intervalo)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Promedio de uso de recursos	9. ¿Se han llevado a cabo pruebas de carga y rendimiento para simular situaciones de uso intensivo?	3	3	3	
	10. ¿Cómo se comporta el desarrollo de la historia de usuario en términos de tiempo de respuesta y rendimiento general?	3	3	3	
	11. ¿Se identificaron cuellos de botella en las funciones desarrolladas para la historia de usuario?	3	3	3	

Observaciones (precisar si hay suficiencia): El instrumento presenta suficiencia

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: SÁENZ APARI ABRAHAM RAFAEL DNI: 1045496

Especialidad del validador: Metodólogo [] Temático [x]

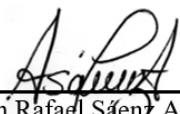
18 de noviembre del 2023.

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión


 Abraham Rafael Sáenz Apari
 DNI 10454966
 CEO
 AJ SOLUTIONS SAC

CARTA DE PRESENTACIÓN

Señor: JUAN BRUES CHUMPE AGESTO

Presente

Asunto: **VALIDACIÓN DE INSTRUMENTOS A TRAVÉS DE JUICIO DE EXPERTO.**

Me es muy grato comunicarme con usted para expresarle mis saludos y así mismo, hacer de su conocimiento que, siendo estudiante del Programa de Maestría en Ingeniería de Sistemas con mención en Tecnologías de la Información de la Escuela de Posgrado de la UCV, en la sede LIMA NORTE, requiero validar los instrumentos con los cuales se recogerá la información necesaria para poder desarrollar mi investigación y con la sustentaré mis competencias investigativas en la Experiencia curricular de Diseño y desarrollo del trabajo de investigación.

El nombre de mis Variables es: ChatGPT y Código de fuente de software; siendo imprescindible contar con la aprobación de docentes especializados para poder aplicar los instrumentos en mención, se ha considerado conveniente recurrir a usted, ante su connotada experiencia en temas educativos y/o investigación educativa.

El expediente de validación, que le hacemos llegar contiene:

- Carta de presentación.
- Formato de Validación.
- Certificado de validez de contenido de los instrumentos.

Expresándole mis sentimientos de respeto y consideración me despido de usted, no sin antes agradecerle por la atención que dispense a la presente.

Atentamente.



.....
ALTAMIRANO TSENG, LEENKIM HAHNEMANN

D.N.I 73179995

Evaluación por juicio de expertos

Respetado juez: Usted ha sido seleccionado para evaluar el instrumento para medir las Variables ChatGPT y Código de fuente de software. La evaluación del instrumento es de gran relevancia para lograr que sea válido y que los resultados obtenidos a partir de éste sean utilizados eficientemente; aportando al quehacer psicológico. Agradecemos su valiosa colaboración.

6. Datos generales del juez:

Nombre del juez:	Juan Brues Chumpe Agosto
Grado profesional:	Maestría () Doctor (x)
Área de formación académica:	Clínica () Social () Educativa (x) Organizacional ()
Áreas de experiencia profesional:	Educación / Ingeniería
Institución donde labora:	Universidad Privada César Vallejo
Tiempo de experiencia profesional en el área:	2 a 5 años () Más de 5 años (x)
Experiencia en Investigación (si corresponde)	Asesor de Tesis en Pregrado y Posgrado

7. Propósito de la evaluación:

Validar el contenido del instrumento, por juicio de expertos.

8. Datos de la escala:

Nombre de la Prueba:	Instrumento para medir la variable Código de fuente de software
Autor:	ALTAMIRANO TSENG, LEENKIM HAHNEMANN
Procedencia:	Ficha de análisis
Administración:	Directa
Tiempo de aplicación:	40 minutos
Ámbito de aplicación:	Entidad consultora
Significación:	La ficha de análisis de la Variable Código de fuente de software está compuesto de 3 dimensiones que son: Complejidad, Precisión y Eficiencia. El objetivo de esta medición es establecerlas relaciones entre ChatGPT y el Código de fuente de software.

9. Soporte teórico:

ChatGPT: Se trata de la herramienta que ha generado más discusión y ha captado la atención por todos los medios como la red social entre otros. Chat GPT marcó un punto que sobresale mundialmente dentro de la IA. (EDEM, 2023)

Código fuente de software: Se refiere a una colección de instrucciones escritas en un lenguaje de programación legible por humanos, generalmente en formato de texto sin formato, que se utiliza para facilitar el trabajo de los programadores de computadoras. (Armetrics, 2021)

Escala/ÁREA	Subescala (dimensiones)	Definición
INTERVALO	Complejidad	Se puede definir como la característica de algo que es complicado en su estructura. En otras palabras, el término "complejidad" se refiere a la presencia de múltiples elementos que están interconectados entre sí. Esta palabra se origina en el término "complejo", que tiene sus raíces en el latín <i>complexus</i> , el participio pasado de <i>complexi</i> , que significa "enlazar". (Significados, 2023)
	Precisión	Estamos hablando de su habilidad para acertar en el corto plazo, es decir, lograr resultados que se ajusten a las expectativas o que estén muy cerca de lo esperado. Aunque en el lenguaje común a menudo se utiliza indistintamente con "exactitud", es importante no mezclar o confundir estos dos conceptos. (Concepto, 2022)
	Eficiencia	Se relaciona con el uso sensato de los recursos disponibles para lograr un objetivo establecido previamente. Implica la habilidad de alcanzar una meta predefinida en el menor tiempo y con la menor cantidad de recursos necesarios, lo que implica una optimización. (Definición, 2022)

10. Presentación de instrucciones para el juez:

A continuación, a usted le presento la ficha de análisis para medir la Variable Código de fuente de software elaborado por Altamirano Tseng, Leenkim Hahnemann. De acuerdo con los siguientes indicadores califique cada uno de los ítems según corresponda.

Categoría	Calificación	Indicador
CLARIDAD El ítem se comprende fácilmente, es decir, su sintácticay semántica son adecuadas.	1. No cumple con el criterio	El ítem no es claro.
	2. Bajo Nivel	El ítem requiere bastantes modificaciones o una modificación muy grande en el uso de las palabras de acuerdo con su significado o por la ordenación de estas.
	3. Moderado nivel	Se requiere una modificación muy específica de algunos de los términos del ítem.
	4. Alto nivel	El ítem es claro, tiene semántica y sintaxis adecuada.
COHERENCIA El ítem tiene relación lógica con la dimensión o indicador que está midiendo.	1. totalmente en desacuerdo (no cumple con el criterio)	El ítem no tiene relación lógica con la dimensión.
	2. Desacuerdo (bajo nivel de acuerdo)	El ítem tiene una relación tangencial /lejana con la dimensión.
	3. Acuerdo (moderado nivel)	El ítem tiene una relación moderada con la dimensión que se está midiendo.
	4. Totalmente de Acuerdo (alto nivel)	El ítem se encuentra está relacionado con la dimensión que está midiendo.
RELEVANCIA El ítem es esencialmente importante, es decir debe ser incluido.	1. No cumple con el criterio	El ítem puede ser eliminado sin que se vea afectada la medición de la dimensión.
	2. Bajo Nivel	El ítem tiene alguna relevancia, pero otro ítem puede estar incluyendo lo que mide éste.
	3. Moderado nivel	El ítem es relativamente importante.
	4. Alto nivel	El ítem es muy relevante y debe ser incluido.

Leer con detenimiento los ítems y calificar en una escala de 1 a 4 su valoración, así como solicitamos brindes sus observaciones que considere pertinente

1 No cumple con el criterio
2. Bajo Nivel
3. Moderado nivel
4. Alto nivel

Dimensiones del instrumento:

- **Primera dimensión:** Complejidad
- **Indicador 1:** Longitud del código (Escala en Intervalo)

Indicador	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Longitud del código	1. ¿Cuál es la longitud promedio de líneas de código por función para la historia de usuario?	3	3	3	
	2. ¿Cómo describirías la estructura general del código en términos de complejidad?	3	3	3	
	3. ¿Se han identificado áreas de código particularmente extensas que puedan aumentar la complejidad?	3	3	3	
	4. En términos de la estructura del código, ¿cómo describirías la relación entre funciones o métodos y su complejidad?	3	3	3	

- **Segunda dimensión:** Precisión
- **Indicador 2:** Porcentaje de pruebas aprobadas (Escala en Intervalo)

Indicador	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Porcentaje de pruebas aprobadas	5. ¿Cuántas pruebas funcionales se han realizado en total para la historia de usuario?	3	3	3	
	6. ¿Cuál es el porcentaje de pruebas funcionales aprobadas en relación con el total de pruebas realizadas?	3	3	3	
	7. ¿Cuántas pruebas unitarias (testing) se han realizado en total para la historia de usuario?	3	3	3	
	8. ¿Cuál es el porcentaje de pruebas unitarias aprobadas en relación con el total de pruebas realizadas?	3	3	3	

- **Tercera dimensión:** Eficiencia
- **Indicador 3:** Promedio de uso de recursos (Escala en Intervalo)

Indicadores	Ítem	Claridad	Coherencia	Relevancia	Observaciones/ Recomendaciones
Promedio de uso de recursos	9. ¿Se han llevado a cabo pruebas de carga y rendimiento para simular situaciones de uso intensivo?	3	3	3	
	10. ¿Cómo se comporta el desarrollo de la historia de usuario en términos de tiempo de respuesta y rendimiento general?	3	3	3	
	11. ¿Se identificaron cuellos de botella en las funciones desarrolladas para la historia de usuario?	3	3	3	

Observaciones (precisar si hay suficiencia): El instrumento presenta suficiencia

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: CHUMPE AGESTO, JUAN BRUES DNI: 44824114

Especialidad del validador: Metodólogo [] Temático [x]

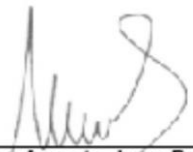
18 de noviembre del 2023.

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión


Chumpe Agosto Juan Brues
DNI: 44824114