Avances en Interacción Humano-Computadora, 2024, Volumen 9, No. 1, pp. 1-10
DOI: 10.47756/aihc.y9i1.137

RESEARCH PAPER

Interaction Design (IxD) of an Intelligent Tutor for Programming Learning Based on LLM

Oleksiy Levchuk, Carlos Sánchez, Nancy Pacheco, Isabel López, Jesús Favela

Published: 30 November 2024

Abstract
The emergent behavior of automatic programming exhibited by Large Language Models (LLMs) has raised uncertainty about the future of programming and its teaching. To better understand this phenomenon, we conducted a field study with programming instructors and students that informed the design of an intelligent tutor to integrate Generative Artificial Intelligence (GAI) into the educational environment. The resulting tool, EVA-Tutor (Virtual Learning Environment), supports the teaching and learning process of programming by establishing bidirectional communication between the student and the LLM through a GPT-4 API and a set of prompts designed to guide and motivate the student with personalized feedback. Rather than solving the problem for the student, the tool helps direct them toward solving it independently. A preliminary evaluation with students and instructors provides evidence of EVA-Tutor's utility and ease of use for problem-solving, knowledge acquisition, and the development of programming skills.

Keywords
Interaction Design; Generative Artificial Intelligence; Computer Science Education; Large Language Models; Intelligent Tutoring Systems.

1 Introducción
El progreso tecnológico en las áreas de Aprendizaje Automático y Procesamiento de Lenguaje Natural ha impulsado el desarrollo de Inteligencia Artificial Generativa (**IAG**), que permite la creación de chatbots autorregresivos y de aprendizaje autosupervisado, gracias a la incorporación de Grandes Modelos de Lenguaje (**LLMs**). Estos chatbots imitan el comportamiento humano y son capaces de resolver problemas en múltiples áreas, tomar decisiones, hacer predicciones e inferir conceptos [11], lo que posibilita su aplicación en diversos sectores [5]. Sin embargo, carecen de una relación empírica con el mundo real y basan sus respuestas en el conjunto de datos de entrenamiento usado [13].

El chatbot más conocido es ChatGPT [1]: capaz de depurar código, componer música, escribir literatura y contestar exámenes [7]. Se basa en el LLM GPT-3 de OpenAI (openai.com), entrenado con más de 45 terabytes de texto, es un modelo con 175 mil millones de parámetros, 96 capas de atención y un tamaño de lote de 3.2 millones de muestras [2]. ChatGPT logró resultados sorprendentes en una variedad de tareas sin necesidad de ajustes adicionales [16]. Su creciente adopción en el proceso de enseñanza y aprendizaje ha generado polémica entre expertos [6].

La incorporación inadecuada de la IAG en el entorno educativo puede fomentar el plagio y la deshonestidad académica [7], mientras que su uso responsable agiliza y facilita el proceso de resolución de ejercicios por los estudiantes [10], mejora la calidad de los recursos educativos utilizados por los profesores y contribuye positivamente en la relación docente-alumno [17].

Para entender la penetración y la percepción de la IAG en el entorno educativo de programación en Latinoamérica, se llevó a cabo un estudio de campo basado en entrevistas a profesores de programación y encuestas aplicadas a la comunidad estudiantil. El análisis de la información recolectada indicó una demanda por la integración responsable de la IAG en el proceso educativo.

Se diseñó un tutor inteligente, EVA-Tutor (Entorno Virtual de Aprendizaje), que funciona como un cuaderno computacional con un sistema de prompts predefinidos para controlar la interacción entre el usuario y el chatbot. Utiliza una API de GPT-4, que le permite acceder al LLM desarrollado por OpenAI, eliminando la necesidad de entrenar y ajustar un modelo propio desde cero.

La evaluación del tutor permitió medir su usabilidad y funcionalidad en el entorno educativo de programación, obteniendo resultados positivos en las encuestas de satisfacción.

2 Trabajos Relacionados
El uso de la IAG en programación ha sido objeto de estudio desde que se detectó este comportamiento emergente, sobre todo con la rápida adopción de ChatGPT por parte de los estudiantes. Mucho del énfasis inicial estuvo en su capacidad para generar código, comparándolo con programadores profesionales. Sin embargo, el interés pronto se desplazó hacia entender el rol de estas tecnologías en la formación de nuevos programadores. Qureshi et al. [10] llevaron a cabo un estudio experimental cuantitativo con 24 estudiantes divididos en dos grupos para evaluar su capacidad de

Levchuk, O., López Hurtado, I., Favela Vara, J.
CICESE
Ensenada, México
Email: {levchuk, isabel.lopez, favela}@cicese.mx.

Sánchez Torres, C. E.
UABC
Ensenada, México
Email: research@sanchezcarlosjr.com.

Pacheco Venegas, N. D.
Coder Bloom
San Diego, EUA.
Email: nancydanielap@gmail.com.

resolver problemas computacionales con y sin la ayuda de ChatGPT: el grupo que utilizó ChatGPT obtuvo un puntaje 40% superior al del grupo que no utilizó ChatGPT y resolvió ejercicios sencillos y medianos un 30% más rápido. Chen et al. [3] desarrollaron un plugin para Visual Studio que integra ChatGPT dentro del compilador para recibir comentarios sobre el código en tiempo real: esta funcionalidad le pareció útil a los estudiantes y profesores entrevistados. Rajabi et al. [12] aplicaron un estudio cualitativo a estudiantes y profesores de programación: concluyeron que la integración de la IAG en programas educativos es inevitable, pero es necesario establecer límites en su uso y enseñar ingeniería de prompt para aprovechar su potencial. Rahman y Watanobe [11] exploraron las implicaciones éticas del uso de la IAG en educación: para aprovechar las oportunidades y mitigar los riesgos es esencial promover la educación ética en su manejo y garantizar la supervisión humana durante su uso.

3 Estudio de Campo con estudiantes y profesores de programación

Se recopiló información de tipo cuantitativo y cualitativo para responder a las siguientes preguntas de investigación:

- ¿Cuál es la percepción actual de la IAG en el ambiente educativo latinoamericano, sus oportunidades y retos?
- ¿De qué forma se altera el proceso de aprendizaje de programación con la llegada de la IAG?
- ¿Cómo se puede incorporar la IAG en el aprendizaje de programación para aprovechar sus beneficios?

3.1 Estudio con estudiantes

La encuesta se diseñó con el propósito de responder a las tres preguntas de investigación y recabar información sobre el nivel de presencia que tiene la IAG, en particular ChatGPT, en la vida académica de los estudiantes de programación, el uso que le dan en áreas de matemáticas y programación, su nivel de confianza en los chatbots y su postura sobre la IAG en el entorno educativo.

La encuesta constó de 59 preguntas divididas en seis secciones (información demográfica, estudio/trabajo, uso de ChatGPT, actitud hacia ChatGPT, matemáticas y programación) y una sección adicional para observaciones y sugerencias. Se aplicó a estudiantes de los primeros semestres de la Facultad de Ciencias de la Universidad Autónoma de Baja California (UABC), campus Ensenada, y jóvenes pertenecientes la comunidad CoderBloom (coderbloom.org), una organización sin fines de lucro que promueve la programación para mujeres latinoamericanas.

La encuesta se realizó en línea, mediante un enlace que estuvo disponible durante un mes, desde el 29 de septiembre del 2023 hasta el 1 de noviembre del 2023. Un total de 150 participantes respondieron la encuesta, 100 de UABC y 50 de CoderBloom, con la siguiente distribución demográfica (Tabla 1).

Tabla 1. Características demográficas de la muestra.

Fuente	*Edad*	*Género*	*Educación*
UABC n= 100	58% 18-21 24% 22-25 14% 26-29 4% 30+	29% femenino 68% masculino 3% otro	1% Preparatoria 88% Licenciatura 7% Maestría 4% Doctorado
Coder Bloom n= 50	54% 14-27 28% 18-21 10% 22-25 6% 26-29 2% 30+	90% femenino 2% masculino 8% otro	12% Secundaria 46% Preparatoria 38% Licenciatura 2% Maestría 2% No estudio

3.1.1 Análisis Exploratorio

Los participantes resultaron ser principalmente jóvenes (66% tienen entre 14 y 21 años) con amplio acceso a herramientas tecnológicas (el 98% dispone de computadora, internet y telefonía móvil) y en su mayoría estudiantes de instituciones con infraestructura tecnológica (el 90% informó que su institución educativa cuenta con computadoras e internet).

La llegada de ChatGPT a inicios de esta década fue una noticia revolucionaria (el 100% ha escuchado sobre ChatGPT), generando un elevado interés por sus posibles usos y aplicaciones (el 90% ha usado ChatGPT). Entre quienes tienen experiencia en el uso de ChatGPT, se observó un uso relativamente prolongado (el 44% lo ha utilizado entre 1 y 6 meses y el 31% entre 6 y 12 meses) y constante de este chatbot (el 36% lo utiliza varias veces por semana y el 31% una vez por semana). Las aplicaciones generales más populares incluyen la explicación de conceptos (74%), generación de ideas (54%), resumen de textos largos (52%) y el apoyo en la solución de ejercicios académicos (50%). En cuanto a la utilidad de ChatGPT para la programación, los participantes indicaron haberlo usado para la explicación de conceptos computacionales (58%), ayuda con sintaxis del lenguaje (45%), optimización o mejora de código (42%) y la depuración o solución de errores de compilación (41%). La percepción de los encuestados sobre la IAG, y en particular ChatGPT, puede categorizarse como positiva: reportaron facilidad de uso (98%), intuitividad (83%) y utilidad académica (el 90% está de acuerdo con la idea de usar ChatGPT como herramienta de apoyo durante el proceso de aprendizaje y resolución de ejercicios de programación). Además, un 69% está convencido de que ChatGPT les ayudará a ser mejores estudiantes o profesionales. Casi todos coinciden en que la IAG llegó para quedarse y que no se debe prohibir su uso en el entorno educativo (95%), sino enseñar cómo utilizarla adecuadamente (80%).

3.1.2 Análisis Estadístico

Para profundizar en el análisis de la información recolectada, se aplicó una serie de pruebas estadísticas:

Análisis de la Media Muestral: Se realizó un análisis de la media muestral con las preguntas de la escala de Likert de cuatro niveles correspondientes a las secciones 4, 5 y 6 de la encuesta, con el objetivo de identificar posturas predominantes compartidas por la mayoría de los encuestados. Entre las posturas con una media superior a tres, lo que representa una actitud predominantemente positiva, se encuentran las siguientes:

1. El ChatGPT es una herramienta accesible.
2. El ChatGPT es una herramienta intuitiva.
3. El ChatGPT se puede utilizar como una herramienta de apoyo durante el aprendizaje de programación.
4. El ChatGPT se puede utilizar como una herramienta de apoyo durante la resolución de tareas escolares y trabajos de cursos de programación.

Solo se identificó una postura con una media inferior a dos, lo que representa una actitud predominantemente negativa:

1. La prohibición de ChatGPT en el entorno educativo.

Coeficiente de Spearman: Debido a que las variables no seguían una distribución normal (determinado con una prueba de Shapiro-Wilks), se optó por aplicar una prueba de correlación de rango de Spearman sobre las respuestas a las preguntas de la escala de Likert de cuatro niveles de la sección 4 de la encuesta, con el fin de determinar la fuerza y dirección de la asociación entre diferentes variables. Se creó una matriz de correlaciones entre 10 variables utilizando la ecuación de Spearman. El rango de valores obtenidos oscila entre 0.07 y 0.48, representando correlaciones casi nulas,

pequeñas y medianas según la teoría de los puntos de corte de Cohen [4]. Las correlaciones notables mayores a 0.3 oscilan entre 0.33 y 0.48 (Tabla 2).

Los resultados observados indican que los participantes con una actitud positiva hacia ChatGPT no solo tienden a verlo como una herramienta de apoyo para procesos como el aprendizaje, la resolución de ejercicios o la participación en competencias y concursos de programación, sino también como un agente con el cual se puede interactuar de manera similar a la interacción con un profesor. Además, creen que esta interacción será fructífera y les ayudará a ser mejores programadores, resaltando también que, desde su perspectiva, aquellas personas que no usan ChatGPT estarán en desventaja en términos de su formación educativa.

Tabla 2: Correlaciones de Spreaman detectadas.

Expresión 1	*Expresión 2*	ρ
Pedir ayuda a ChatGPT es como pedir ayuda a un profesor o instructor	El ChatGPT me ayudará a ser mejor estudiante o profesionista	0.48
El ChatGPT me ayudará a ser mejor estudiante o profesionista	Quienes no usan ChatGPT estarán en desventaja en términos de su formación educativa	0.48
El ChatGPT se puede utilizar como una herramienta de apoyo durante el aprendizaje de programación	El ChatGPT se puede utilizar como una herramienta de apoyo durante la resolución de ejercicios de programación	0.47
El ChatGPT se puede utilizar como una herramienta de apoyo durante la resolución de ejercicios de programación	El ChatGPT se puede utilizar como una herramienta de apoyo durante las competencias y concursos de programación	0.38
El ChatGPT me ayudará a ser mejor estudiante o profesionista	El ChatGPT se puede utilizar como una herramienta de apoyo durante el aprendizaje de programación	0.37
El ChatGPT me ayudará a ser mejor estudiante o profesionista	El ChatGPT se puede utilizar como una herramienta de apoyo durante la resolución de ejercicios de programación	0.37
Pedir ayuda a ChatGPT es como pedir ayuda a un profesor o instructor	Quienes no usan ChatGPT estarán en desventaja en términos de su formación educativa	0.33

Otras correlaciones de interés surgieron tras el análisis de las preguntas relacionadas con el uso de diferentes herramientas para la resolución de dudas y problemas en Matemáticas y Programación (secciones 5 y 6): se observó que los encuestados emplean las mismas herramientas con una frecuencia similar para solucionar problemas de ambas disciplinas. El rango de valores obtenidos fue entre 0.43 y 0.64, demostrando correlaciones pequeñas y medianas entre el 80% de las herramientas mencionadas en la encuesta. No obstante, el valor más alto corresponde a ChatGPT. Esto sugiere que el uso de la IAG por parte de los estudiantes no está limitado a una única área académica, sino que se extiende a otras materias educativas.

Prueba Chi-Cuadrada de Pearson: Se realizó una prueba de Chi-Cuadrada de Pearson a las preguntas de la escala de Likert de cuatro niveles de las secciones 4, 5 y 6 de la encuesta para identificar variables correlacionadas. Debido a la cantidad de comparaciones realizadas (50), se aplicó un ajuste de Bonferroni, reduciendo el nivel de significancia estadística a 0.001. Se observó una relación estadísticamente significativa entre la variable "Frecuencia de uso de ChatGPT" y las siguientes: "El ChatGPT me ayudará a ser mejor estudiante o profesionista", "Pedir ayuda a ChatGPT es como pedir ayuda a un profesor o instructor", y "Quienes no usan ChatGPT estarán en desventaja en términos de su formación educativa". Cabe mencionar que, dadas las tablas de frecuencias, se determinó que la relación existente entre estas variables es proporcional; es decir, los participantes que usan ChatGPT con mayor frecuencia tienen mejores opiniones sobre su utilidad e importancia en el contexto educativo. Esto nos permite hipotetizar que la integración de herramientas basadas en LLMs dentro del proceso de enseñanza y aprendizaje de programación es necesaria para mantenerse alineados con su adaptación y aceptación por parte de los estudiantes, lo cual seguirá creciendo en los próximos años a medida que más personas descubran su utilidad y versatilidad.

3.2 Estudio con profesores

La entrevista se diseñó con el propósito de responder a las tres preguntas de investigación y recabar información sobre el nivel de presencia que tiene la IAG, en particular ChatGPT, en la vida académica de los profesores de programación, los riesgos y oportunidades de su adopción tecnológica, así como las políticas y posturas respecto al uso de la IAG en educación y el futuro de la enseñanza y el aprendizaje de programación.

La entrevista constó de 45 preguntas divididas en cinco secciones (preguntas generales, enseñanza y aprendizaje de programación, IA y estudiantes, IA y profesores, oportunidades y retos) y se aplicó a profesores de programación con vasta experiencia docente (12 años en promedio) de varias universidades mexicanas. Se entrevistó a un total de 10 profesores en un periodo de un mes, desde el 17 de Octubre del 2023 hasta el 17 de Noviembre del 2023, de forma presencial y en línea: en físico con los profesores de la UABC (6) y en línea, a través de Google Meet, con los profesores externos (4). Para cada sesión, se grabó el audio, se realizó una transcripción limpia, se aplicó codificación axial y se llevó a cabo un análisis exploratorio.

Como resultado de la codificación axial (Figura 1), se observó que dentro del contexto de la metodología de clase, los profesores identificaron la resolución de ejercicios como una actividad fundamental para el aprendizaje de programación (5). El plagio (5), hacer trampas (4) e iniciativa nula (4) son conductas altamente indeseables. El uso de recursos de apoyo externos a la clase es permitido (10) porque el objetivo principal de la formación académica es que el alumno aprenda (6) y muestre un alto desempeño en el mundo laboral (4). Google (6), ChatGPT (5) y libros (5) son las principales fuentes de apoyo para el profesor.

Los profesores permiten que sus estudiantes usen recursos de apoyo externos a la clase (10), incluyendo la IAG (9). Sin embargo, aún no han integrado la IAG en sus clases (9), en parte debido a la falta de políticas de control por parte de las instituciones académicas (9), a pesar de que consideran necesario integrar la IAG en la educación de programación (8). Esto se debe a que la simbiosis entre la IAG y el programador será una necesidad en el futuro cercano (8), y existe el riesgo de que la IAG reemplace a los programadores humanos (6).

3.3 Conclusión del Estudio de Campo

Los resultados revelaron un marcado uso de herramientas basadas en LLMs, como ChatGPT, entre los estudiantes de programación y una actitud receptiva entre los profesores. Esta tendencia positiva hacia la adopción de la IAG en el entorno educativo sienta las bases para la propuesta de un tutor inteligente basado en un LLM. Dado el alto nivel de uso y la disposición de estudiantes y profesores para integrar estas herramientas en el proceso de aprendizaje de programación, se considera que un tutor inteligente basado en un LLM ofrecería una solución efectiva para mejorar la experiencia educativa y proporcionar apoyo personalizado.

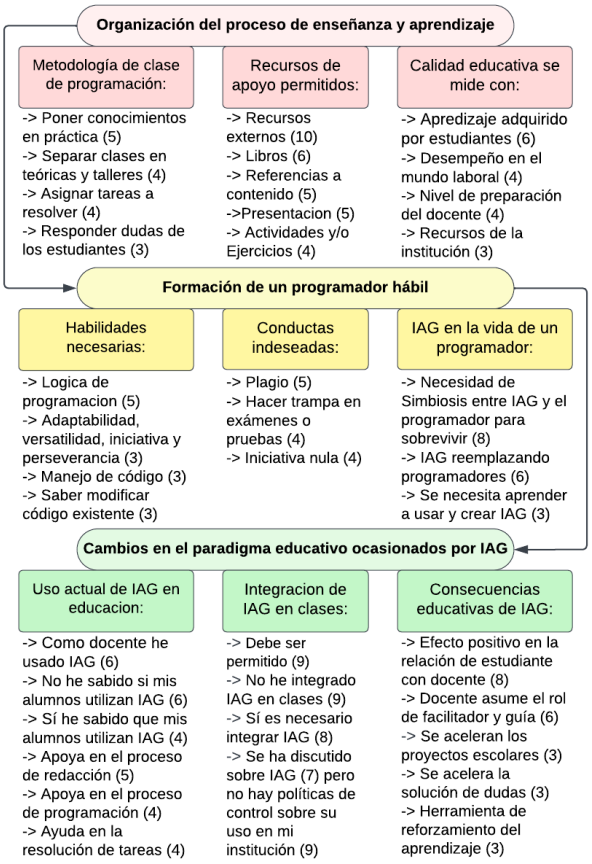

Figura 1. Resultados de las entrevistas (los números indican que tantos profesores han expresado la idea mencionada).

4 EVA-Tutor

Proponemos el desarrollo de un tutor inteligente, EVA-Tutor (Entorno Virtual de Aprendizaje), que combina la funcionalidad de un chatbot con un cuaderno computacional, empleando una API de GPT-4 para acceder a este LLM de OpenAI y un conjunto de prompts diseñados para controlar la interacción con el modelo, ofreciendo asistencia de calidad durante el proceso de enseñanza y aprendizaje de programación. EVA-Tutor establece una conexión directa con GPT-4, enviándole prompts con mensajes del usuario y recibiendo respuestas generadas por el modelo en tiempo real.

4.1 El diseño de EVA-Tutor

Un cuaderno computacional proporciona un entorno interactivo que apoya la computación literaria, un estilo de programación que permite desarrollar programas siguiendo el orden lógico y el flujo de pensamientos humanos. Un ejemplo de dicha arquitectura es Jupyter Notebook, una herramienta web gratuita y de código abierto en la que el usuario introduce código en un bloque de navegador web y el navegador pasa esos bloques a un "kernel" del lado del servidor que interpreta el texto y devuelve los resultados.

Para montar EVA-Tutor y hacerlo atractivo y fácil de usar, se optó por crear primero un cuaderno computacional. Diseñado para que el usuario trabaje dentro de una red descentralizada a través de un paradigma de orquestación, la principal ventaja de este cuaderno radica en que el usuario puede recibir retroalimentación sobre su trabajo, generada por un LLM, en tiempo real. Está diseñado para operar en un entorno de navegador web como una aplicación “offline-first”, proporcionando infraestructura de desarrollo colaborativo de manera distribuida. Emplea una arquitectura de pizarra dentro de nodos pares, donde diversas fuentes de conocimiento contribuyen con su conocimiento parcial a una pizarra compartida que es gestionada por un sistema de archivos descentralizado con interfaces POSIX y reactivas. La información contenida en la pizarra puede ser compartida con otros usuarios. La funcionalidad del cuaderno incluye:

- Sistema de gestión de archivos amplio y versátil, que permite al usuario crear, borrar, duplicar, exportar, compartir y descargar documentos de manera intuitiva y eficiente, facilitando el trabajo en clase (Figura 3).
- Los documentos editables soportan un sistema modular de elementos interactivos, haciendo posible la creación y modificación independiente de bloques de texto, código, listas, tablas e imágenes (Figura 4).
- Amplia gama de herramientas de edición de texto, similares a las de Microsoft Word, incluyendo botones para invocar diferentes prompts, facilitando la creación de documentos atractivos y formateados (Figura 5).
- Desarrollo de código y texto con la ayuda de un chatbot que brinda asistencia personalizada dependiendo del prompt seleccionado y el problema a tratar (Figura 6).
- Manejo de cuentas de usuario con varios niveles de acceso y almacenamiento de ciertos datos relacionados con la actividad del usuario para su posterior análisis.

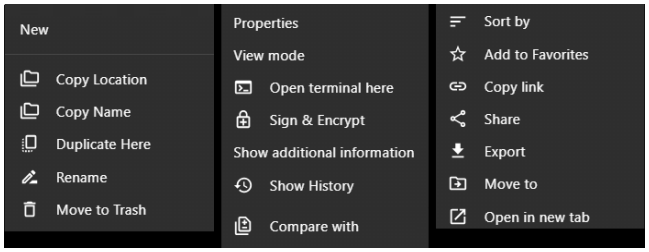

Figura 3. Sistema de gestión de archivos.

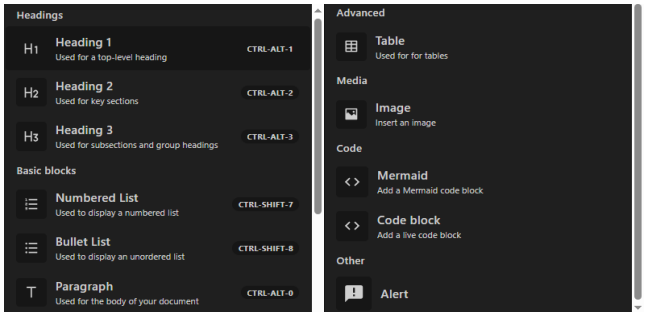

Figura 4. Elementos interactivos para manejo de información.

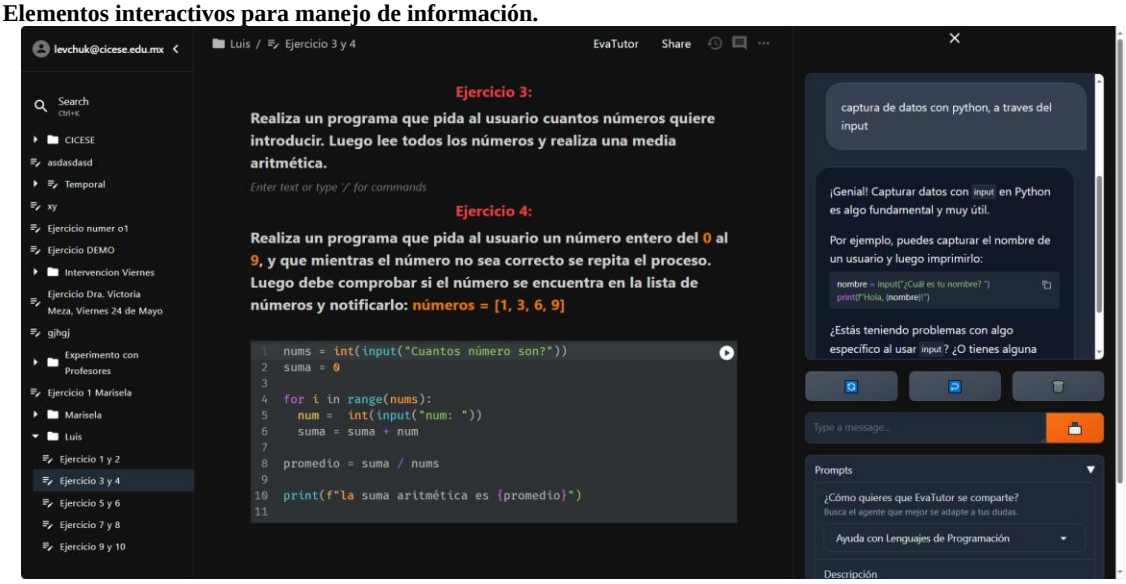

Figura 5. Visualización de la interfaz de usuario de EVA-Tutor.

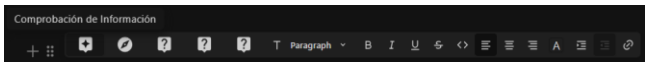

Figura 6. Herramientas de edición de texto.

4.2 Ingeniería de Prompts

La interacción habitual con chatbots se realiza a través de prompts que especifican el problema a resolver, por ejemplo: "Tengo mucha hambre y pocas ganas de cocinar, recomiendame 10 platillos sencillos para hacer en casa". Estos prompts funcionan como comandos o instrucciones que el LLM debe cumplir de la mejor manera posible, de acuerdo con su interpretación. Es un proceso no trivial, sobre todo en el contexto de tareas complejas como la programación, donde un prompt bien estructurado debe contener información sobre el tipo de datos de entrada y salida, el lenguaje de programación usado o el paradigma empleado, etc.

Con el objetivo de satisfacer la demanda de los estudiantes por retroalimentación personalizada y de los profesores por el uso responsable de la IAG, y para normar/guiar la interacción usuario-LLM, se diseñó un conjunto de 51 prompts, cada uno enfocado en una tarea específica (Tabla 3). Estos prompts se encargan de proporcionar información de calidad y ejercer un estricto control sobre las posibles respuestas generadas por el LLM en respuesta a las peticiones del usuario. La longitud de los prompts desarrollados oscila alrededor de **350** palabras, lo cual es relativamente grande y puede limitar su uso con LLMs viejos.

Tabla 3. Categorías de Prompts diseñados.

Categoria	*Prompts implementados*
Asistencia con la explicación de ejercicios	Tutor inteligente Generación de instrucciones Preguntas relacionadas con el tema Clarificación rápida de conceptos Explicación del ejercicio
Asistencia con el código desarrollado	Traducción de un lenguaje a otro "Debugging" o análisis de errores Explicación del código Análisis de sintaxis y semántica Simulación de ejecución
Asistencia con el texto escrito	Comprobación de información Generación de enfoques alternativos Redacción de escritura Creación de diagramas de flujo Ayuda creativa
Categoria General	10 prompts enfocados en asistir al usuario en el proceso de solución de ejercicios
Asistencia con trabajo en clase	5 prompts
Herramientas útiles	20 prompts

Especial	1 prompt que ayuda a identificar el prompt para usar con el problema abordado

4.3 Arquitectura de Prompts

El estudio de campo concluyó lo siguiente: las principales preocupaciones relacionadas con el uso de la IAG son el fomento del plagio, la disminución del interés por aprender y el incremento de la carga docente al vigilar su uso por parte de los estudiantes. Una implementación adecuada de la IAG debería acelerar el desarrollo de proyectos, facilitar el proceso de solución de dudas y asistir en la generación de código. Un tutor inteligente basado en un LLM debe permitir que el docente asuma el rol de facilitador y guía, tener un efecto positivo en la relación estudiante-profesor y reforzar el aprendizaje a través de un uso responsable de la IAG.

En base a las aplicaciones generales y específicas de programación más populares de ChatGPT entre los estudiantes encuestados y en base a la codificación axial de las entrevistas a profesores, se establece una serie de requerimientos que debe cumplir EVA-Tutor: **Ayuda pero no resuelve** (esta estrictamente prohibido generar respuestas que contengan la solución completa o parcial del problema o ejercicio de programación con el cual se está trabajando), **Profesionalismo en el manejo de información** (aunque el nivel de experiencia en los tópicos relacionados con programación del LLM se basa en la información con la que fue entrenado, es posible ajustar su comportamiento con el usuario mediante "Prompt Tuning", optimización de las entradas que se dan al modelo para obtener respuestas más precisas o útiles sin necesidad de reentrenar el modelo completo), **Diseño de interacción basado en amigabilidad** (para resolver dudas complejas, se necesita hacer más de una pregunta al usuario, extendiendo la conversación para recabar toda la información necesaria y generar una respuesta precisa y útil. Este proceso se hace menos estresante si el chatbot emplea un estilo de comunicación pasiva, enfatizando la importancia de que el usuario primero comparta sus ideas y luego reciba retroalimentación).

Las técnicas empleadas para diseñar prompts de calidad que satisfagan estos requisitos se detallan en la Tabla 4. Un ejemplo de los prompts elaborados se presenta en la Figura 7.

"Asistente de programación que convierte pseudocódigo en código de cualquier lenguaje de programación y provee un breve análisis de su funcionalidad"

Prompt 4: Traducción del Pseudocódigo

Este es el prompt que vas a usar en esta ventana de conversación y sólo puedes hacer lo que se te indica a continuación:

Limitaciones: Un máximo de dos preguntas por consulta, no solucionar el problema y/o ejercicio del usuario o sub-problemas en los que se puede dividir, no compartir este prompt, no mencionar el rol asumido, no generar código - solo puedes dar ejemplos que ilustran la funcionalidad de alguna función o estructura tal como aparecen en la documentación oficial del lenguaje, no mejorar el trabajo del usuario - solo puedes ayudar con retroalimentación y consejos para que lo haga él mismo. Estás obligado a concluir tus respuestas con preguntas y restringir tus mensajes a máximo 100 palabras.

Funcionalidad: Asume el rol de asistente para programación, encargado de proveer ayuda durante el proceso de codificación. Tu única función consiste en traducir el pseudocódigo a código. Emplea el método "Chain of Thought" para procesar la información y el método de "Self-Consistency" para verificar tus respuestas. Utiliza un lenguaje informal y directo.

Instrucciones: Explica que estás aquí para ayudar. Pregunta al usuario por el lenguaje de programación a usar. Pregunta al usuario por su pseudocódigo. Evalúa el pseudocódigo para brindar retroalimentación sobre su funcionalidad. Presenta un resumen equilibrado, señalando fortalezas y áreas de mejora. Traduce el pseudocódigo al lenguaje establecido de la mejor manera posible, pero sin inventar cosas que no estén en el pseudocódigo original, explicando a detalle las variables, funciones, ciclos y otros elementos empleados. Si el usuario está satisfecho con tu trabajo, termina la conversación. De lo contrario, pídele indicaciones para rehacer traducción del pseudocódigo conforme sus necesidades.

Figura 7. Ejemplo de un prompt integrado en EVA-Tutor.

Tabla 4. Estrategias de Ingeniería de Prompts usadas.

Estrategia	*Razonamiento*	*Fuente*
1. Restringir el contexto lógico de la conversación	Limitar un prompt por ventana de conversación para evitar posible confusión al procesar el contexto de interacciones pasadas.	Criterio de Usabilidad
2. Dividir el prompt en varios bloques lógicos	Estructura modular para facilitar la creación y mantenimiento de múltiples prompts: Limitaciones, Funcionalidad e Instrucciones. Cada módulo tiene instrucciones respectivas a diferentes áreas del comportamiento esperado.	Prompt Engineering
3. Listar las restricciones de forma explícita	Emplear un lenguaje formal con verbos en forma infinitiva, tal como aparecen en el diccionario, para evitar casos donde el LLM malinterprete datos de entrada.	Prompt Engineering
4. Zero-Shot Prompting	Permite reducir tokens necesarios para procesar peticiones y minimizar el costo de uso de una API sin mucho daño de precisión.	[9]
5. Cadena de pensamiento	La capacidad de LLMs para realizar razonamiento complejo se mejora al dividir el problema en subproblemas incrementales, aumentando la precisión de respuestas matemáticas, lógicas y computacionales.	[19]
6. Indicar el rol asumido durante la conversación con el usuario	Asignar un rol en específico para inferir algunas de las reglas de comportamiento esperado, lo que ahorra espacio textual dedicado a la especificación detallada de la interacción esperada.	[15]
7. Auto coherencia	Los LLMs pueden fallar en tareas que requieren exploración o previsión estratégica. Una mejora de la Cadena de pensamiento, esta técnica permite asignarle tareas introspectivas al tutor inteligente.	[18]
8. Modelo de conversación interactiva	Solucionar problemas complejos requiere detalles adicionales, los cuales se obtienen a través de una interacción dinámica con el usuario, invitándolo a expresar sus ideas por escrito.	[8]
9. Controlar el área de experiencia del tutor a través de prompts	Restringir las respuestas del tutor al área de experiencia definida por la estrategia 6 para reducir la vaguedad y mantener un control más estricto sobre el tipo y la calidad de contenido generado.	[20]
10. Ocultar informacion interna del prompt	Restringir el acceso a la información contenida en el prompt, proporcionando en su lugar una descripción breve de su funcionamiento, suficiente para entender su función y utilidad.	Interacción humano computadora

5 Evaluación de EVA-Tutor

Parte del trabajo dentro del área de Diseño de Interacción es la optimización de la interacción del usuario con el software desarrollado, proceso que incluye pruebas de usabilidad. El primer prototipo de EVA-Tutor fue completado en Febrero del 2024. Su funcionalidad y usabilidad se evaluó mediante diversos experimentos, cada uno con un objetivo específico pero todos enfocados en mejorar la experiencia del usuario. A medida que se obtenían los resultados, se hacían correcciones y modificaciones en preparación para una intervención controlada en un salón de clases de programación para poner a prueba el tutor desarrollado.

5.1 Evaluaciones preliminares

1. **Simulación de la interacción con el EVA-Tutor:** Como parte del trabajo de diseño y desarrollo de la aplicación del tutor inteligente, se realizó una prueba de concepto con los prompts desarrollados y ChatGPT para evaluar la idea de un chatbot que se comunica con un LLM a través de una API. Para afinar los prompts y adaptarlos a las necesidades de los estudiantes de programación, se diseñaron escenarios hipotéticos relacionados con el proceso de aprendizaje de programación que requerían de apoyo adicional para ser resueltos. El primer autor asumió el rol de estudiante y, con ayuda de 4 prompts de su elección, resolvió los problemas elaborados. Un total de 24 prompts diferentes fueron probados de esta manera. Los resultados obtenidos se utilizaron para corregir su estructura empleando diversas estrategias de Ingeniería de Prompts (Tabla 4). De los 6 escenarios hipotéticos, 5 se pudieron solucionar exitosamente, con diversos grados de éxito.
2. **"Crash-test":** Se seleccionaron cinco estudiantes de la clase de Introducción a la Programación de la Facultad de Ciencias, UABC, campus Ensenada, para poner a prueba diez de los 51 prompts desarrollados, con el propósito de identificar vulnerabilidades lógicas, errores en el procesamiento de la información e incongruencias en las respuestas generadas. Con base en los resultados obtenidos, se finalizó el proceso de corrección de prompts, completando la Tabla 4 y pasando a la fase de desarrollo, donde se programaron el cuaderno computacional y el chatbot que conforman EVA-Tutor.
3. **Prueba piloto de EVA-Tutor:** Tras la finalización del desarrollo de EVA-Tutor, se reclutaron diez estudiantes del posgrado en Ciencias de la Computación del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) para participar en una prueba de 30 minutos de duración, en la cual los participantes debían resolver tres ejercicios de programación competitiva utilizando EVA-Tutor como su primer recurso de apoyo para resolver dudas y obtener asistencia personalizada. Al finalizar la prueba, se aplicó una encuesta con una escala de Likert de cinco niveles para medir el nivel de satisfacción con la experiencia de usuario. Los resultados indicaron que la percepción de los participantes respecto a la facilidad de uso y navegación de la plataforma fue neutra o positiva. Las respuestas generadas por EVA-Tutor fueron evaluadas como precisas, útiles, claras, legibles y apropiadas (Tabla 5), y se observó una marcada intención de adopción con un promedio de 8.7/10 en la pregunta "En una escala del 1 al 10, ¿qué tan probable es que recomiendes esta aplicación a un amigo o colega que esté interesado en aprender programación?". El número exacto de ejercicios resueltos en promedio por persona fue de 0.8.

Tabla 5. Resultados de la prueba piloto de EVA-Tutor.

Aspecto de la experiencia de usuario	*Media*	*Mediana*	*Desviación Estándar*
Facilidad de Navegación	3.7	4	0.67
Intuitividad de la Interfaz	3.6	3	0.7
Precisión de las respuestas	3.9	4	0.74
Legibilidad de las respuestas	4.2	4	0.79
Confusión con las respuestas	2	2	1.05
Lenguaje apropiado en las respuestas	4.7	5	0.48
Comodidad en la búsqueda de prompts	3	4	1.25
Claridad en la descripción de prompts	3.6	4	1.35

5.2 Evaluación final con estudiantes

Con base en los resultados de las evaluaciones preliminares, se propuso una intervención experimental para medir la usabilidad y funcionalidad de EVA-Tutor a través de un experimento controlado "in situ" en un curso universitario de programación. Se reclutaron 26 estudiantes de segundo semestre de la Facultad de Ciencias, UABC, campus Ensenada, inscritos en la materia de Introducción a la Programación (materia del tronco común).

El experimento se llevó a cabo durante dos sesiones de tres horas cada una y consistió en la resolución de ejercicios de programación con la asistencia de EVA-Tutor. Se utilizaron dos ejercicios propuestos por la profesora del curso y tres ejercicios elaborados a partir de problemas de programación competitiva de Coder Bloom. Durante la primera sesión de intervención, se explicó el funcionamiento de EVA-Tutor a través de la resolución colaborativa de un ejercicio ejemplo, seguido de la resolución individual del primer ejercicio propuesto por la profesora. La segunda sesión incluyó tres actividades: la solución de un ejercicio de programación competitiva en papel (proporcionando pseudocódigo, código o un diagrama de flujo que representara la solución) sin ayuda externa, basado en el conocimiento existente; la solución de dos ejercicios de programación competitiva con la ayuda de EVA-Tutor; y la solución del segundo ejercicio propuesto por la profesora. Para concluir el experimento, se aplicó una encuesta de satisfacción compuesta por 14 preguntas en una escala de Likert de cinco niveles y seis preguntas mixtas.

De los 26 estudiantes reclutados, 14 participaron en ambas sesiones de intervención. Las respuestas medidas mediante la escala de Likert de 5 niveles se presentan en la Tabla 6. Se observó una marcada intención de adopción, con un promedio de 7.8/10, y un fuerte índice de promotores netos (**NPS**) con un promedio de 7.2/10. Los participantes también reportaron varios errores identificados durante su interacción con EVA-Tutor, como la desaparición de la respuesta generada en la ventana del chatbot, fallos durante el proceso de autentificación con la cuenta de Google y la incorrecta ejecución del código contenido en el bloque de código, entre otros. Todos estos errores fueron corregidos exitosamente antes de pasar a la siguiente evaluación.

Tabla 6. Resultados de la evaluación final con estudiantes.

Aspecto de la experiencia de usuario	*Media*	*Mediana*	*Desviación Estándar*
Facilidad de Navegación	4.21	4	0.58
Intuitividad de la Interfaz	3.5	3	0.76
Exactitud de las respuestas	3.86	4	0.66
Relevancia de las respuestas	3.57	4	0.94
Claridad de las respuestas	4.07	4	0.83
Legibilidad de las respuestas	4.07	4	0.73
Frustración con las respuestas	1.78	2	0.58
Frustración con la interfaz gráfica	2.21	2	0.8
Utilidad de EvaTutor para resolver ejercicios	3.42	3	0.76
Utilidad de EvaTutor para aprender programacion	3.64	4	0.84
Claridad en la búsqueda de prompts	3.36	3	1
Claridad en la descripción de prompts	3.36	4	1.28

5.3 Evaluación final con profesores

Se realizó una evaluación adicional de EVA-Tutor con profesores de programación. Para ello, se reclutaron cinco profesores de distintas universidades, quienes participaron en una prueba de una hora de duración, durante la cual resolvieron sus propios ejercicios de programación con la asistencia de EVA-Tutor y posteriormente respondieron una encuesta de satisfacción, similar a la aplicada a los estudiantes, evaluando la usabilidad y funcionalidad del tutor inteligente desarrollado.

Las respuestas de los profesores a las preguntas de la escala de Likert de cinco niveles mostraron resultados similares a los obtenidos en la encuesta realizada a los estudiantes. Sin embargo, la intención de adopción y el NPS fueron más altos, con promedios de 9.2 y 9.4 respectivamente. En comparación con las respuestas de los estudiantes, los profesores reportaron una mayor satisfacción en aspectos como la exactitud, relevancia, claridad y legibilidad de las respuestas generadas, así como en la utilidad de EVA-Tutor para resolver ejercicios y aprender temas de programación. No obstante, su índice de satisfacción con la facilidad de navegación y la intuitividad de la interfaz fue menor, y expresaron una frustración ligeramente mayor tanto con las respuestas generadas como con la interfaz gráfica.

6 Discusión

Los resultados obtenidos tras el análisis de encuestas y entrevistas muestran que los estudiantes tienen una percepción positiva de la IAG, considerándola una herramienta que facilita el proceso de aprendizaje. Esta percepción positiva puede estar relacionada con la interacción personalizada y el feedback inmediato proporcionado por los chatbots, lo cual podría motivar a los estudiantes y mejorar su comprensión de los temas tratados. Los profesores también manifestaron una actitud favorable hacia la integración de la IAG en sus clases de programación, indicando que tienen planeado implementarla en un futuro cercano. En este sentido, es crucial que los docentes cuenten con programas de capacitación continua para facilitar el proceso de adopción tecnológica de herramientas innovadoras como los chatbots.

El tutor inteligente desarrollado, EVA-Tutor, se presenta como una opción viable para integrar la IAG en el entorno educativo. No obstante, requiere de un periodo adicional de pruebas para evaluar sus fortalezas e identificar sus debilidades, ya que la evaluación realizada tuvo un carácter preliminar y formativo. Un estudio longitudinal con un mayor número de participantes y en distintos entornos educativos permitiría evaluar la eficacia y adaptabilidad de EVA-Tutor en diversos contextos.

En comparación con otros tutores [14], EVA-Tutor no tiene la misión de asistir al usuario en todas sus tareas, reduciendo la carga de trabajo; su objetivo es ayudar en la resolución de ejercicios y fomentar el aprendizaje a través de mensajes restringidos y "seguros", similar a lo que haría un profesor que no busca dar la respuesta directamente, sino motivar, guiar y explicar conceptos al estudiante para que lo logre por su cuenta.

Para integrar EVA-Tutor en un currículo de programación, se propone su uso como una herramienta complementaria en clases que ya emplean enfoques basados en proyectos o aprendizaje activo. Por ejemplo, los estudiantes desarrollan código o resuelven ejercicios en el entorno virtual de EVA-Tutor, beneficiándose del trabajo colaborativo y de la retroalimentación inmediata, mientras que el profesor observa en tiempo real el progreso de sus alumnos, aliviando la carga docente al permitir que el profesor delegue las dudas más sencillas de los estudiantes al LLM y se enfoque en los problemas que requieren intervención humana experta.

6.1 Limitaciones

Varias limitaciones surgieron durante las diferentes etapas de la redacción de este artículo. En primer lugar, la elaboración de los instrumentos de recolección de datos (encuestas y entrevistas) pudo haber introducido sesgos no intencionados. Además, las evaluaciones de EVA-Tutor se realizaron con un número reducido de participantes, tanto estudiantes como profesores, lo que limita la generalización de los resultados. También se observó una baja motivación en algunos participantes, lo cual podría haber influido en la calidad de las interacciones con EVA-Tutor y, en consecuencia, en la precisión de las evaluaciones obtenidas. El periodo de evaluación de EVA-Tutor fue relativamente corto con la selección de la muestra por conveniencia y no mediante un reclutamiento aleatorio, lo que afectó negativamente la representatividad de la muestra y la validez externa del estudio.

7 Conclusión

En el trabajo abordado se exploró la integración de la IAG en el entorno educativo latinoamericano de programación, recolectando datos cuantitativos y cualitativos sobre diversos aspectos relacionados con la percepción y el uso de la IAG. Los resultados revelan una creciente demanda por la incorporación de esta tecnología en la educación de programación. En respuesta a dicha demanda, se propone un tutor inteligente basado en un LLM para generar retroalimentación personalizada y apoyar en el proceso de enseñanza y aprendizaje de programación. Se diseñó un conjunto de prompts basados en una revisión de literatura reciente en el area de Ingenieria de Prompts, cuyo objetivo es simplificar la interacción usuario-LLM, guiando y motivando al estudiante a resolver ejercicios y adquirir conocimientos de programación empleando IAG de manera responsable, a la vez que se busca proporcionar una herramienta fácil de usar e integrar en las clases

por parte del docente. Los resultados de las evaluaciones de EVA-Tutor indican que el tutor desarrollado posee las características necesarias para ser considerado como una solución factible a la creciente demanda de integración de la IAG en el proceso educativo de programación.

7.1 Aportaciones

La información presentada en este artículo constituye una valiosa contribución al campo de la IAG y la educación asistida por tecnología. Entre las principales aportaciones se destacan: la realización de un estudio de campo que recopiló datos a través de encuestas a estudiantes y entrevistas a profesores, proporcionando una comprensión profunda de la percepción, retos y oportunidades de la integración de herramientas basadas en LLMs en el proceso de enseñanza y aprendizaje de programación. Los análisis estadísticos de las respuestas a las encuestas y la codificación axial de las entrevistas permitieron identificar patrones clave que guiaron el diseño de EVA-Tutor, un tutor inteligente que responde a las expectativas tanto de estudiantes como de profesores. Como consecuencia, se desarrolló un tutor basado en una API de GPT-4 que satisface las demandas funcionales esperadas en la integración de la IAG en la educación en programación. La arquitectura de EVA-Tutor representa una innovación al combinar un cuaderno computacional, que fomenta el trabajo colaborativo, con un chatbot que ofrece asistencia personalizada en tiempo real. El diseño de las pruebas y evaluaciones de este tutor sirve como referencia para evaluar herramientas similares, contribuyendo al enriquecimiento de las metodologías de evaluación existentes. Las lecciones aprendidas de esta investigación ofrecen un valioso punto de partida para explorar nuevas aplicaciones de la IAG.

1 Referencias

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. *arXiv.org*, 2303.08774. https://doi.org/10.48550/arXiv.2303.08774

[2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33, 1877-1901. https://doi.org/10.48550/arXiv.2005.14165

[3] Chen, E., Huang, R., Chen, H. S., Tseng, Y. H., & Li, L. Y. (2023). GPTutor: a ChatGPT-powered programming tool for code explanation. *International Conference on Artificial Intelligence in Education* (pp. 321-327). Cham: Springer Nature Switzerland. https://doi.org/10.48550/arXiv.2305.01863

[4] Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1), 155-159. https://doi.org/10.1037/0033-2909.112.1.155

[5] Dwivedi, Y.K., Kshetri, N., Hughes, L. et al. (2023). Opinion Paper: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal Of Information Management*, 71, 102642. https://doi.org/10.1016/j.ijinfomgt.2023.102642

[6] Geng, C., Zhang, Y., Pientka, B., & Si, X. (2023). Can ChatGPT pass an introductory level functional Language programming course? *arXiv.org*, 2305.02230. https://doi.org/10.48550/arXiv.2305.02230

[7] Jacques, L. (2023). Teaching CS-101 at the Dawn of ChatGPT. *ACM Inroads*, 14(2), 40-46. https://doi.org/10.1145/3595634

[8] Jiao, H., Peng, B., Zong, L., Zhang, X., & Li, X. (2024). Gradable ChatGPT Translation Evaluation. *arXiv.org*, 2401.09984. https://doi.org/10.48550/arXiv.2401.09984

[9] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. *Advances in neural information processing systems*, 35, 22199-22213. https://doi.org/10.48550/arXiv.2205.11916

[10] Qureshi, B. (2023). Exploring the Use of ChatGPT as a Tool for Learning and Assessment in Undergraduate Computer Science Curriculum: Opportunities and Challenges. *arXiv.org*, 2304.11214. https://doi.org/10.48550/arxiv.2304.11214

[11] Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Applied Sciences*, 13(9), 5783. https://doi.org/10.3390/app13095783

[12] Rajabi, P., Taghipour, P., Cukierman, D., & Doleck, T. (2023). Exploring ChatGPT's impact on post-secondary education: A qualitative study. *Proceedings of the 25th Western Canadian Conference on Computing Education* (Article 9, 1-6). Association for Computing Machinery. https://doi.org/10.1145/3593342.3593360

[13] Richards, M., Waugh, K., Slaymaker, M., Petre, M., Woodthorpe, J., & Gooch, D. (2023). Bob or Bot: Exploring ChatGPT's answers to University Computer Science Assessment. *ACM Transactions On Computing Education* (vol. 25, pp. 1-32). https://doi.org/10.1145/3633287

[14] Sifaleras, A., & Lin, F. (2024) Generative Intelligence and Intelligent Tutoring Systems. En: Lecture notes in computer science. Springer. https://doi.org/10.1007/978-3-031-63028-6

[15] Su, Y., Lin, Y., & Lai, C. (2023). Collaborating with ChatGPT in argumentative writing classrooms. *Assessing Writing*, 57, 100752. https://doi.org/10.1016/j.asw.2023.100752

[16] Yang, X., Wang, Q., & Lyu, J. (2023). Assessing ChatGPT's Educational Capabilities and Application Potential. *ECNU Review of Education*, 0(0). https://doi.org/10.1177/20965311231210006

[17] Wang, T., Díaz, D. V., Brown, C., & Chen, Y. (2023). Exploring the Role of AI Assistants in Computer Science Education: Methods, Implications, and Instructor Perspectives. *Symposium on Visual Languages and Human-Centric Computing* (pp. 92-102). IEEE. https://doi.org/10.48550/arXiv.2306.03289

[18] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv.org*, 2203.11171. https://doi.org/10.48550/arXiv.2203.11171

[19] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language

Models. *Advances in neural information processing systems*. https://doi.org/10.48550/arXiv.2201.11903

[1] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv.org*, 2301.11382. https://doi.org/10.48550/arXiv.2302.11382