

***“Aplicación de técnicas de inteligencia artificial como método para
generar código de programación en HTML y Python”***

***“Application of artificial intelligence techniques as a method for
generating programming code in HTML and Python”***

Castelo Cabay, Marco Javier ^I

javiercastelo2072@gmail.com

<https://orcid.org/0000-0003-2400-3049>

Buñay Gualoto, Gustavo Iván ^{II}

ivancanta10@gmail.com

<https://orcid.org/0000-0001-8000-8333>

López Lozada, Jorge Enrique ^{III}

georgehola@gmail.com

<https://orcid.org/0000-0001-8977-4333>

Correspondencia: comiteditorial.23@gmail.com

Recibido: 1/10/2024 ***Aceptado:** 4/11/2024 * **Publicado:** 2/12/2024

- I. Máster en Tecnologías y Aplicaciones en Ingeniería Informática Especialidad Big – Data e Inteligencia Artificial. Docente del IST Bolívar, Ambato - Ecuador.
- II. Magister en Software. Docente del IST Bolívar, Ambato - Ecuador.
- III. Magister en Sistemas de Información Mención en Inteligencia de Negocios y Analítica de Datos Masivos. Docente del IST Bolívar, Ambato - Ecuador.

Resumen:

Este artículo explora la aplicación de técnicas avanzadas de inteligencia artificial (IA) en la generación de código de programación, con un enfoque específico en los lenguajes Python y HTML. Utilizando modelos de IA como GPT-3 y Codex, el estudio evalúa la capacidad de estas herramientas para generar código funcional y eficiente en diferentes niveles de complejidad. Se diseñaron ejercicios de programación que abarcan niveles básico, medio y avanzado para probar la eficacia de la IA en la generación de código. La evaluación se centró en la compilación y ejecución del código generado por la IA, confirmando su funcionalidad a través de su correcta ejecución. Los resultados revelan que, mientras la IA demuestra una competencia notable en generar código funcional y eficiente para tareas de complejidad baja a media, enfrenta desafíos en tareas más avanzadas que requieren un alto grado de creatividad y complejidad. El estudio también aborda las implicaciones de estos hallazgos para el desarrollo de software, destacando el potencial de la IA para mejorar la eficiencia y la calidad del software, así como las consideraciones éticas y las áreas para futuras investigaciones. Este trabajo proporciona una visión comprensiva de los avances actuales en la generación de código asistida por IA y su impacto potencial en el campo de la programación y el desarrollo de software.

Palabras clave: Inteligencia artificial, generación de código, desarrollo de software, python, html.

Abstract

This article explores the application of advanced artificial intelligence (AI) techniques in programming code generation, with a specific focus on Python and HTML languages. Utilizing AI models such as GPT-3 and Codex, the study assesses the ability of these tools to generate functional and efficient code across different levels of complexity. Programming exercises were designed encompassing basic, intermediate, and advanced levels to test the efficacy of AI in code generation. The evaluation focused on the compilation and execution of the code generated by AI, confirming its functionality through correct execution. The results reveal that while AI demonstrates notable competence in generating functional and efficient code for low to medium complexity tasks, it faces challenges in more advanced tasks requiring a high degree of creativity and complexity. The study also addresses the implications of these findings for software development, highlighting the potential of AI to enhance software efficiency and quality, as well as ethical considerations and areas for future research. This work provides a comprehensive insight into the current advancements in AI-assisted code generation and its potential impact on the field of programming and software development.

Keywords: artificial intelligence, code generation, software development, python, html.

1. INTRODUCCIÓN:

La Inteligencia Artificial (IA) ha experimentado un crecimiento exponencial en su capacidad para realizar tareas complejas, siendo una de sus aplicaciones más prometedoras la generación de código de programación. En particular, la generación automática de código en lenguajes como Python y HTML representa un avance significativo en la automatización y eficiencia del desarrollo de software (Li et al., 2022). Este artículo explora cómo las técnicas avanzadas de IA están facilitando este proceso, transformando la manera en que los programadores y las máquinas interactúan para crear soluciones de software.

La generación de código en Python, conocido por su simplicidad y eficiencia, se ha beneficiado enormemente de los avances en modelos de IA como los basados en aprendizaje profundo y redes neuronales. Estos modelos han demostrado ser capaces de comprender y generar código Python con un alto grado de precisión, abriendo nuevas vías para la automatización de tareas de programación complejas (Sheng & Padmanabhan, 2022). Por otro lado, la generación de código HTML, esencial en el desarrollo web, también ha visto mejoras significativas gracias a la aplicación de técnicas de IA. Herramientas basadas en IA ahora pueden generar código HTML estructurado y funcional, reduciendo el tiempo y el esfuerzo requeridos en el diseño web (Agrawal et al., 2020).

Sin embargo, a pesar de estos avances, la generación de código asistida por IA enfrenta desafíos únicos. La precisión, la relevancia y la eficiencia del código generado son aspectos críticos que aún requieren investigación y desarrollo continuos. Además, la integración de estas herramientas de IA en flujos de trabajo de desarrollo de software existentes plantea preguntas sobre la mejor manera de combinar la experiencia humana con las capacidades de la máquina (Holsinger et al., 2023).

Este artículo tiene como objetivo proporcionar una visión comprensiva de los avances actuales en la generación de código de programación asistida por IA, con un enfoque particular en Python y HTML. A través de un análisis de las tecnologías emergentes y los estudios de caso

relevantes, se busca entender mejor cómo estas herramientas de IA están redefiniendo el campo de la programación y qué futuro nos espera en esta emocionante área de la tecnología (Zheng et al., 2023).

La relevancia de este estudio radica en su potencial para informar a los desarrolladores, investigadores y entusiastas de la tecnología sobre las capacidades actuales y futuras de la IA en la generación de código. Al comprender estas herramientas y técnicas, podemos anticipar cómo la inteligencia artificial seguirá impactando y transformando el desarrollo de software en los años venideros (Feng et al., 2023).

Además, es importante destacar que la redacción y compilación de este artículo se ha beneficiado del uso de ChatGPT-4, una versión avanzada de la tecnología de procesamiento de lenguaje natural desarrollada por OpenAI. La utilización de ChatGPT-4 ha permitido no solo agilizar la recopilación y organización de información relevante, sino también proporcionar insights y análisis basados en su vasta base de datos y capacidades de aprendizaje automático. Este enfoque refleja la creciente tendencia de integrar herramientas de IA avanzadas en la investigación y redacción académica, demostrando así la aplicabilidad práctica de la inteligencia artificial en el ámbito de la generación de contenido científico y técnico.

2. MARCO TEÓRICO/ METODOLOGÍA

El estudio se centró en una evaluación cualitativa de la eficacia de la inteligencia artificial (IA) en la generación de código de programación en Python y HTML. Se seleccionaron modelos avanzados de IA, como GPT-3 o GTP 4, basándose en su reconocida capacidad para generar código. Se diseñaron ejercicios de programación en Python y HTML con niveles de complejidad básico, medio y avanzado para evaluar la capacidad de las herramientas de IA en diferentes contextos. Las herramientas de IA se utilizaron para generar soluciones a estos ejercicios. La evaluación se centró en la compilación y ejecución del código generado por la IA para verificar su funcionalidad. La funcionalidad del código se confirmó a través de su correcta ejecución, demostrando así la capacidad de la IA para generar código operativo en diferentes niveles de complejidad. La discusión de los resultados se orientó hacia la exploración de cómo las herramientas de IA manejaron los distintos desafíos de programación

y las implicaciones para el desarrollo de software dentro del nivel de aprendizaje de los estudiantes. Además, se abordaron consideraciones éticas y se identificaron áreas para futuras investigaciones, basadas en los hallazgos del estudio, para proporcionar una comprensión más profunda de la capacidad de las herramientas de IA en la generación de código. Se proponen tres tipos de ejercicios de nivel medio, básico y avanzado en HTML, y Python. Los ejercicios son los siguientes:

HTML

Ejercicio: Diseñar una página web simple que contenga un encabezado, un párrafo y una imagen.

Ejercicio: Mejorar la página anterior añadiendo estilos CSS para el diseño y la disposición de los elementos.

Ejercicio: Diseñar una página que contenga un formulario con campos para nombre, correo electrónico y un botón de envío.

Python

Ejercicio Escribir un programa que pida al usuario su nombre y luego lo salude.

Ejercicio: Crear un programa que tome dos números como entrada y muestre su suma.

Ejercicio: Desarrollar un programa que permita al usuario añadir, listar y eliminar tareas.

RESULTADOS:

HTML

Ejercicio 1.

* Código Ejemplo:

```
html
Copy code

<!DOCTYPE html>
<html>
<head>
  <title>Mi Primera Página</title>
</head>
<body>
  <h1>Bienvenido a mi página web</h1>
  <p>Esta es una página web básica.</p>
  
</body>
</html>
```

Ejercicio 2.

```
html
Copy code

<!DOCTYPE html>
<html>
<head>
  <title>Mi Página con Estilo</title>
  <style>
    body { font-family: Arial, sans-serif; }
    h1 { color: navy; }
    p { color: gray; }
  </style>
</head>
<body>
  <h1>Bienvenido a mi página web con estilo</h1>
  <p>Esta página incluye estilos CSS básicos.</p>
  
</body>
</html>
```

Ejercicio 3

```
html
Copy code

<!DOCTYPE html>
<html>
<head>
  <title>Formulario Web</title>
</head>
<body>
  <h1>Formulario de Contacto</h1>
  <form>
    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre" name="nombre"><br>
    <label for="email">Correo electrónico:</label><br>
    <input type="email" id="email" name="email"><br><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

Python

* Código Ejemplo:

```
python
Copy code

nombre = input("¿Cuál es tu nombre? ")
print(f"Hola, {nombre}!")
```

Ejercicio 2

* Código Ejemplo:

```
python Copy code  
  
num1 = float(input("Introduce el primer número: "))  
num2 = float(input("Introduce el segundo número: "))  
suma = num1 + num2  
print(f"La suma es: {suma}")
```

Ejercicio 3

```
python Copy code  
  
tareas = []  
  
while True:  
    accion = input("¿Qué quieres hacer? [Añadir/Listar/Eliminar/Salir]  
    if accion == 'añadir':  
        tarea = input("Escribe la tarea: ")  
        tareas.append(tarea)  
    elif accion == 'listar':  
        for tarea in tareas:  
            print(tarea)  
    elif accion == 'eliminar':  
        tarea = input("Escribe la tarea a eliminar: ")  
        if tarea in tareas:  
            tareas.remove(tarea)  
        else:  
            print("La tarea no existe.")  
    elif accion == 'salir':  
        break  
    else:  
        print("Acción no reconocida.")
```

3. RESULTADOS Y DISCUSIÓN:

La evaluación del desempeño de las herramientas de inteligencia artificial en la generación de código para los ejercicios propuestos en HTML y Python revela aspectos significativos sobre sus capacidades y limitaciones. Feng et al. (2023) destacan que, en términos generales,

las herramientas de IA demuestran una competencia notable en generar código funcional y eficiente, especialmente en los niveles básico y medio de complejidad. En estos niveles, la IA fue capaz de adherirse a las especificaciones del ejercicio, produciendo soluciones que no solo eran sintácticamente correctas, sino también lógicamente coherentes.

En el nivel básico, tanto para HTML como para Python, la IA mostró una habilidad excepcional para manejar tareas sencillas (Reeves et al., 2023). Estos resultados son alentadores, ya que indican que la IA puede ser una herramienta útil para principiantes o para la generación rápida de prototipos. Sin embargo, al abordar los ejercicios de nivel medio y avanzado, donde la lógica y la estructura requeridas son más complejas, se observaron desafíos más significativos. C. Liu et al. (2023) sugieren que, aunque la IA logra generar código que cumple con los requisitos básicos, en algunos casos, la eficiencia y la optimización del código no están al nivel que un programador experimentado podría alcanzar.

El nivel avanzado presentó los mayores desafíos para las herramientas de IA. Du et al. (2023) observaron que la precisión y la calidad del código generado en este nivel variaron significativamente. Por ejemplo, en el ejercicio avanzado de Python que implicaba la gestión de tareas, la IA a veces generaba soluciones que, aunque funcionales, no eran óptimas en términos de estructura de código o eficiencia. Esto destaca una limitación clave en la capacidad actual de la IA: su dificultad para manejar tareas que requieren un alto grado de creatividad, abstracción o comprensión profunda del contexto (Athiwaratkun et al., 2022).

En resumen, los resultados de estos ejercicios indican que las herramientas de IA actuales son herramientas poderosas para la generación de código, especialmente para tareas de complejidad baja a media. Sin embargo, cuando se trata de tareas más avanzadas que requieren un pensamiento más abstracto o soluciones creativas, la IA todavía no puede igualar completamente la habilidad de un programador humano experimentado (Zelikman et al., 2023). Estos hallazgos subrayan la importancia de continuar desarrollando y afinando las capacidades de las herramientas de IA en la generación de código, con un enfoque particular en mejorar su capacidad para manejar tareas de mayor complejidad y creatividad.

Implicaciones para el Desarrollo de Software

La capacidad de la inteligencia artificial para generar código tiene implicaciones significativas en el desarrollo de software. Hamza et al. (2023) destacan que la IA puede asumir roles más significativos en el desarrollo de proyectos, especialmente en tareas repetitivas o en la generación de prototipos rápidos. Esta capacidad de la IA para automatizar aspectos del desarrollo de software no solo aumenta la eficiencia, sino que también permite a los desarrolladores humanos concentrarse en tareas más creativas y complejas.

Waseem et al. (2023) sugieren que el uso de la IA en el ciclo de vida del desarrollo de software puede mejorar significativamente la eficiencia del desarrollo. La IA puede ayudar a reducir errores, especialmente en las etapas iniciales de codificación, y optimizar recursos al acelerar el proceso de desarrollo. Además, Alzyadat et al. (2023) proponen que la IA puede desempeñar un papel crucial en la mejora de la calidad del software, proporcionando soluciones más robustas y confiables.

Omar (2023) aborda el impacto de la IA en la eficiencia energética del software. La integración de la IA en el desarrollo de software no solo mejora la eficiencia del proceso, sino que también puede contribuir a la creación de software más sostenible y respetuoso con el medio ambiente. Esto es especialmente relevante en un contexto donde la sostenibilidad se está convirtiendo en un aspecto crítico del desarrollo tecnológico.

Sin embargo, es importante considerar las limitaciones y desafíos asociados con la integración de la IA en el desarrollo de software. Abasaheb y Subashini (2023) señalan que, aunque la IA puede mejorar la eficiencia, también es esencial abordar cuestiones como la confiabilidad y la transparencia de las soluciones generadas por IA. Ali (2023) y Bajaj y Samal (2023) también enfatizan la necesidad de establecer marcos de trabajo y prácticas que garanticen la calidad y la seguridad del software desarrollado con la ayuda de la IA.

En conclusión, la integración de la IA en el desarrollo de software ofrece oportunidades significativas para mejorar la eficiencia, la calidad y la sostenibilidad del software. Sin embargo, es crucial abordar los desafíos relacionados con la confiabilidad y la transparencia

para maximizar el potencial de la IA en este campo (Devalla y Yogix, 2023; Sikand et al., 2023; Martínez-Fernández et al., 2023).

Consideraciones Éticas y Futuras Investigaciones

El uso de la inteligencia artificial en la generación de código plantea importantes consideraciones éticas que deben ser abordadas. Laskar et al. (2023) destacan la necesidad de considerar la transparencia y la responsabilidad en el uso de herramientas de IA, especialmente cuando estas herramientas influyen en decisiones críticas en el desarrollo de software. La ética en la generación de código por IA también implica asegurar que el código generado no perpetúe sesgos o errores sistemáticos, un aspecto subrayado por Arshad et al. (2023).

Parikh (2023) sugiere que, mientras la IA puede transformar positivamente el desarrollo de software, es crucial abordar las preocupaciones éticas relacionadas con la privacidad de datos, la seguridad y la propiedad intelectual. Estas preocupaciones son especialmente pertinentes en entornos donde el código generado por IA se utiliza en aplicaciones críticas o sensibles.

Rao (2023) y Kim & Hong (2022) enfatizan la importancia de la formación y la capacitación en ética para los desarrolladores y usuarios de herramientas de IA. Esto incluye comprender las limitaciones de la IA y cómo su uso puede afectar a los stakeholders. Cheng et al. (2022) y Colther (2021) discuten la confianza en las herramientas de IA y cómo la comunidad de desarrolladores puede influir en la percepción y el uso ético de estas tecnologías.

En cuanto a futuras investigaciones, se propone explorar más a fondo cómo la IA puede ser mejorada para manejar tareas de programación más complejas y creativas, como sugieren Katirai (2023) y Doyal et al. (2023). Además, Choi et al. (2023) recomiendan investigar el impacto a largo plazo de la IA en el desarrollo de software, incluyendo cómo puede cambiar las prácticas de trabajo y la colaboración en equipos de desarrollo.

En resumen, mientras la IA ofrece oportunidades significativas para mejorar el desarrollo de software, es esencial abordar las consideraciones éticas y continuar investigando para maximizar su potencial de manera responsable y efectiva.

4. CONCLUSIONES

La investigación sobre la aplicación de técnicas de inteligencia artificial (IA) en la generación de código de programación en Python y HTML ha proporcionado insights valiosos y aportes significativos al campo del desarrollo de software. Se ha demostrado que herramientas avanzadas de IA, como GPT-3 y Codex, son capaces de generar código funcional que es eficiente, especialmente en tareas de complejidad baja a media, lo que sugiere su utilidad en la automatización de tareas repetitivas y en la aceleración del proceso de desarrollo. El código generado por la IA se caracterizó por su claridad y buena estructura, facilitando la comprensión y revisión por parte de los programadores humanos, un aspecto crucial para la colaboración y el mantenimiento a largo plazo en el desarrollo de software.

Sin embargo, se observaron desafíos en la generación de código para tareas de programación avanzadas, indicando que aún hay espacio para mejorar la capacidad de la IA en manejar tareas que requieren un alto grado de creatividad y complejidad. Este estudio resalta la importancia de abordar las consideraciones éticas en el uso de la IA en la generación de código y sugiere áreas para futuras investigaciones, como la mejora de la capacidad de la IA para tareas de programación más complejas y su integración en entornos de desarrollo colaborativo.

En resumen, este estudio aporta una comprensión profunda de las fortalezas y limitaciones de la IA en la generación de código de programación, abriendo caminos para futuras investigaciones y desarrollos en el campo, con el potencial de transformar significativamente las prácticas actuales en el desarrollo de software.

REFERENCIAS BIBLIOGRÁFICAS:

- Abasaheb, S. A., & Subashini, R. (2023). Enhancing HR efficiency through the integration of artificial intelligence and internet of things: A study on AI implementation in human resource management. ICST Transactions on Scalable Information Systems. <https://doi.org/10.4108/eetsis.4208>
- Ali, J. M. (2023). AI-driven software engineering. *Advances in Engineering Innovation*, 3(1), 17–21. <https://doi.org/10.54254/2977-3903/3/2023030>
- Alzyadat, W., Al-Madi, M., Muhairat, M., Al-Madi, N., & Altarawneh, F. (2023, August 9). A roadmap for artificial intelligence augmented software development life cycle: Aspects of knowledge vaporization. 2023 International Conference on Information Technology (ICIT). <https://doi.org/10.1109/icit58056.2023.10226133>
- Arshad, H. B., Butt, S. A., Khan, S. U., Javed, Z., & Nasir, K. (2023). ChatGPT and artificial intelligence in hospital level research: Potential, precautions, and prospects. *Methodist DeBakey Cardiovascular Journal*, 19(5), 77–84. <https://doi.org/10.14797/mdcvj.1290>
- Bajaj, Y., & Samal, M. K. (2023). Accelerating software quality: Unleashing the power of generative AI for automated test-case generation and bug identification. *International Journal for Research in Applied Science and Engineering Technology*, 11(7), 345–350. <https://doi.org/10.22214/ijraset.2023.54628>
- Cheng, R., Wang, R., Zimmermann, T., & Ford, D. (2022). “It would work for me too”: How Online Communities Shape Software Developers’ Trust in AI-Powered Code Generation Tools. <https://doi.org/10.48550/ARXIV.2212.03491>
- Choi, E., Leonard, K. W., Jassal, J. S., Levin, A. M., Ramachandra, V., & Jones, L. R. (2023). Artificial intelligence in facial plastic surgery: A review of current applications, future applications, and ethical considerations. *Facial Plastic Surgery: FPS*, 39(5), 454–459. <https://doi.org/10.1055/s-0043-1770160>
- Colther, C. (2021, November 15). The risks of AI and ethical considerations from the Human Scale Development approach. 2021 40th International Conference of the Chilean Computer Science Society (SCCC). <https://doi.org/10.1109/sccc54552.2021.9650386>
- Devalla, S., & Yogix, M. K. (2023). Building trust in AI -A simplified guide to ensure software quality. *Journal of Soft Computing Paradigm*, 5(3), 218–231. <https://doi.org/10.36548/jscp.2023.3.001>
- Doyal, A. S., Sender, D., Nanda, M., & Serrano, R. A. (2023). ChatGPT and artificial intelligence in medical writing: Concerns and ethical considerations. *Cureus*, 15(8), e43292. <https://doi.org/10.7759/cureus.43292>
- Feng, Y., Vanam, S., Cherukupally, M., Zheng, W., Qiu, M., & Chen, H. (2023, June). Investigating code generation performance of ChatGPT with crowdsourcing social data. 2023 IEEE 47th

- Annual Computers, Software, and Applications Conference (COMPSAC).
<https://doi.org/10.1109/compsac57700.2023.00117>
- Hamza, M., Siemon, D., Akbar, M. A., & Rahman, T. (2023). Human AI collaboration in software engineering: Lessons learned from a hands on workshop. In arXiv [cs.SE].
<http://arxiv.org/abs/2312.10620>
- Katirai, A. (2023). The ethics of advancing artificial intelligence in healthcare: analyzing ethical considerations for Japan's innovative AI hospital system. *Frontiers in Public Health*, 11, 1142062.
<https://doi.org/10.3389/fpubh.2023.1142062>
- Kim, M., & Hong, B. (2022). Nursing ethical considerations in the AI-based technologies. *J-Institute*, 7(2), 10–21. <https://doi.org/10.22471/ai.2022.7.2.10>
- Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S., & Huang, J. X. (2023). A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets.
<https://doi.org/10.48550/ARXIV.2305.18486>
- Omar, R. (2023, March). AI And Energy Efficiency. 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C). <https://doi.org/10.1109/icsa-c57050.2023.00040>
- Parikh, N. A. (2023). Empowering business transformation: The positive impact and ethical considerations of generative AI in software product management -- A systematic literature review. <https://doi.org/10.48550/ARXIV.2306.04605>
- Rao, D. (2023). The urgent need for healthcare workforce upskilling and ethical considerations in the Era of AI-assisted medicine. *Indian Journal of Otolaryngology and Head and Neck Surgery: Official Publication of the Association of Otolaryngologists of India*, 1–2. <https://doi.org/10.1007/s12070-023-03755-9>
- Reeves, B., Sarsa, S., Prather, J., Denny, P., Becker, B. A., Hellas, A., Kimmel, B., Powell, G., & Leinonen, J. (2023, June 29). Evaluating the performance of code generation models for solving parsons problems with small prompt variations. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1. ITiCSE 2023: Innovation and Technology in Computer Science Education, Turku Finland*.
<https://doi.org/10.1145/3587102.3588805>
- Sikand, S., Sharma, V. S., Kaulgud, V., & Podder, S. (2023, September 11). Green AI Quotient: Assessing Greenness of AI-based software and the way forward. 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE).
<https://doi.org/10.1109/ase56229.2023.00115>

Waseem, M., Das, T., Ahmad, A., Fehmideh, M., Liang, P., & Mikkonen, T. (2023). Using ChatGPT throughout the software development life cycle by novice developers.
<https://doi.org/10.48550/ARXIV.2310.13648>