



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

PROCESO DE TITULACIÓN

DICIEMBRE 2022 - MAYO 2023

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PRUEBA PRÁCTICA

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

TEMA:

ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS DE IA (INTELIGENCIA ARTIFICIAL) CODE GPT Y GITHUB COPILOT COMO ASISTENTES DE PROGRAMACIÓN EN EL DESARROLLO DE SOFTWARE.

EGRESADO:

BASANTES SUÑIGA RONNY ALEJANDRO

TUTOR:

ING. FREDY MAXIMILIANO JORDÁN CORDONEZ

AÑO:

2023

Resumen

El presente estudio comparativo analiza dos herramientas de inteligencia artificial (IA), Code GPT y GitHub Copilot, como asistentes de programación en el desarrollo de software. Se utilizó una metodología descriptiva con un enfoque cualitativo y la técnica de la entrevista para comparar las herramientas, analizando sus características, funcionalidades, eficiencia e identificando sus limitaciones. Como resultado, se concluye que tanto Code GPT como GitHub Copilot tienen un nivel de entrenamiento muy alto que les permite generar código de alta calidad y precisión. Ambas herramientas representan un gran avance en el campo de la inteligencia artificial para la generación de código automatizado. Sin embargo, una de las limitaciones es, en ocasiones, la sugerencia de código obsoleto o que no cumple con las buenas prácticas de programación debido a que se basan en el análisis de grandes cantidades de datos de código existente, que puede estar desactualizado o con errores. Este estudio proporciona una comprensión más profunda de ambas herramientas, permitiendo a los desarrolladores aprovechar al máximo sus funciones y optimizar el tiempo y trabajo durante la programación en el desarrollo de software.

Palabras claves: Inteligencia artificial (IA), Code GPT, GitHub Copilot, Código, Programación.

Abstract

This comparative study analyzes two artificial intelligence (AI) tools, Code GPT and GitHub Copilot, as programming assistants in software development. A descriptive methodology with a qualitative approach and the interview technique was used to compare the tools, analyzing their features, functionalities, efficiency and identifying their limitations. As a result, it is concluded that both Code GPT and GitHub Copilot have a very high level of training that allows them to generate high quality and accurate code. Both tools represent a breakthrough in the field of artificial intelligence for automated code generation. However, one of the limitations is sometimes the suggestion of obsolete code or code that does not comply with good programming practices because they are based on the analysis of large amounts of existing code data, which may be outdated or buggy. This study provides a deeper understanding of both tools, allowing developers to take full advantage of their functions and optimize time and work during programming in software development.

Keywords: Artificial Intelligence (AI), Code GPT, GitHub Copilot, Code, Programming.

Introducción

Actualmente el mercado de los programadores es muy competitivo, las empresas buscan la eficiencia, por lo que es necesario el uso de herramientas que agilicen los procesos de codificación. Durante los últimos años, han surgido nuevas herramientas de inteligencia artificial (IA) que buscan facilitar las tareas de los desarrolladores de software, especialmente en la fase de programación, entre las cuales destacan CODE GPT y GitHub Copilot.

Al momento de programar, comúnmente surgen muchas dificultades que afectan directamente la eficiencia del programador. Entre ellas están la de cometer errores de sintaxis o de lógica, tener que buscar y leer documentación para poder utilizar ciertas funciones, escribir fragmentos de códigos repetitivos de expresiones regulares o funciones simples, entre otros. Estos son factores que afectan en el tiempo de desarrollo de software.

El estudio comparativo de CODE GPT y GitHub Copilot proporciona una comprensión más profunda de estas herramientas, permitiendo a los desarrolladores aprovechar al máximo sus funciones optimizando el tiempo y trabajo durante la programación en el desarrollo de software.

Este estudio tiene como objetivo comparar las herramientas de IA CODE GPT y GitHub Copilot como asistentes de programación en el desarrollo de software, analizando las características, funcionalidad, eficiencia e identificando sus limitaciones. La línea de investigación corresponde a la de Sistemas de Información y Comunicación, Emprendimiento e Innovación, cuya sublínea es la de Redes y Tecnologías Inteligentes de Software y Hardware.

La metodología aplicada en este estudio es la descriptiva con un enfoque cualitativo para identificar, recolectar, organizar y presentar la información de estas dos herramientas. Además, se aplicó la técnica de la entrevista, cuyo instrumento es la guía de la entrevista, a la Ing. Kathy

Sánchez, jefa de proyectos informáticos de la empresa de desarrollo de software “Links” de la ciudad de Guayaquil y al Ing. Livingston Rodríguez, encargado de desarrollo de software del Gobierno Autónomo Descentralizado de la provincia de Orellana (GADPO) con el fin de recabar información relevante y complementaria sobre el uso de las herramientas de IA en la programación durante el desarrollo de software.

Desarrollo

En la actualidad, la inteligencia artificial (IA) ha revolucionado muchos aspectos de la vida diaria, incluso en diversos campos laborales y el desarrollo de software no es una excepción (ENAE, 2021). En el desarrollo de software, durante la etapa de programación, la creación de código es una tarea que requiere de mucho tiempo y esfuerzo por parte de los programadores. Es por eso que han surgido herramientas de IA como Code GPT y GitHub Copilot como asistentes de programación para ayudar a los desarrolladores en la escritura de código.

Ambas herramientas han generado una gran expectativa en la comunidad de programadores debido a su capacidad para completar automáticamente líneas de códigos y sugerir soluciones a problemas de programación simples y complejos, encontrar errores y corregirlos, refactorizar y buscar la solución más óptima e incluso explicar lo que hace el código y entender el contexto para próximas sugerencias.

A continuación, se muestra la tendencia de búsqueda de ambas herramientas en los últimos 12 meses en todo el mundo:



Figura 1: Registro de búsqueda de las palabras Code GPT y GitHub Copilot, últimos 12 meses

Fuente: Google Trends

Como se observa en la figura 1, la búsqueda de estas herramientas está en crecimiento, lo que demuestra el interés de estas tecnologías a nivel mundial.

En esta sección se presenta un estudio comparativo de las herramientas de IA Code GPT y GitHub Copilot como asistentes de programación en el desarrollo de software. Se describe en detalle las características y funcionamiento de cada herramienta, la precisión y calidad de código y se analizan sus principales ventajas y desventajas tomando en cuenta diversos factores.

Para la realización del presente estudio comparativo es importante definir ciertos conceptos generales, claves para el desarrollo del tema, los cuales se explicarán a continuación:

El desarrollo de software se refiere a un conjunto de actividades informáticas dedicadas al proceso de creación, diseño, despliegue y puesta en marcha de un software (IBM, 2019).

Según Solera (2020) existen cinco fases que se llevan a cabo en el desarrollo de software entre las que se encuentran:

1. **Análisis y requisitos.** Es la fase inicial donde se detallan los requerimientos del software en base a las necesidades del cliente. Esta una de las fases más importante, ya que es la base para que en las siguientes fases no existan complicaciones.
2. **Diseño y arquitectura.** En esta fase, el programador o equipo de programadores tendrá que establecer el funcionamiento general del sistema. Se consideran aspectos como el hardware, la arquitectura de red, los casos de uso, entre otros.
3. **Programación e implementación.** Esta es la fase que requiere mayor tiempo y esfuerzo, ya que en esta se codifica el software utilizando uno o varios lenguajes de programación según las características y tipo de proyecto. Se debe asegurar el buen funcionamiento y cumplir con todos los requisitos del software, evitando errores que puedan causar fallas en el futuro.
4. **Pruebas y revisión.** Es la fase donde se prueba el software para determinar si cumple o no con todos los requerimientos, se pueden realizar pruebas a cada módulo de manera individual

y por último probarlo de manera general. También se realiza una revisión para asegurar que el software cumpla con los estándares de calidad y seguridad necesarios.

5. Mantenimiento y cuidado. Una vez se haya puesto en marcha el software, es la fase donde se realiza el mantenimiento y la mejora del sistema, así como para enfrentar los posibles errores y responder a nuevos requisitos.

El presente estudio se enfoca específicamente en la fase de programación, la fase más extensa y donde los programadores se encuentran con varias dificultades al momento de codificar (Intelequia, 2020). En esta etapa se realizan actividades que deben asegurar la calidad y buen funcionamiento del software, entre ellas están:

- Buscar y corregir los posibles errores de sintaxis y lógicos que se puedan presentar en el código.
- Se deben realizar pruebas unitarias del código para comprobar si está funcionando de manera correcta.
- En algunas ocasiones se debe refactorizar el código para optimizarlo, aplicando nuevas técnicas y buenas prácticas de programación.
- Se debe documentar el código de manera detallada para que próximos programadores lo puedan comprender de manera clara en caso de requerir mejoras o modificaciones en el software.

Todas estas tareas conllevan un gran esfuerzo y les toma mucho tiempo a los programadores, por lo que es de gran ayuda el uso de herramientas de inteligencia artificial (IA) como asistentes de programación en esta fase.

En términos generales la inteligencia artificial (IA) se puede definir como un área de estudio de las ciencias computacionales que se basa en la creación de sistemas con la capacidad

para realizar tareas y tomar decisiones tal como lo haría un ser humano. Utilizan algoritmos para obtener una gran cantidad de datos, aprender de ellos y tomar decisiones (Avila-Tomás, 2020).

Las tecnologías que utilizan IA actualmente están siendo aplicadas en casi todos los campos, ayudando a las personas a realizar tareas y obtener una mayor eficiencia ya sea en la vida diaria o en la laboral (Vélez, 2022).

Las herramientas de IA tienen la capacidad de ofrecer sugerencias y predicciones relacionadas con asuntos importantes como la salud, la educación, relaciones sociales y el trabajo. De igual manera, el uso de la IA proporciona una gran ventaja competitiva a las empresas que aplican estas herramientas para optimizar el tiempo en sus procesos.

La IA tiene diferentes enfoques y técnicas, incluyendo el aprendizaje automático (machine learning), las redes neuronales, el procesamiento del lenguaje natural (NLP), entre otros. Los mismos que son importantes conocer para entender el funcionamiento de las herramientas del presente estudio.

El aprendizaje automático (machine learning) es una técnica fundamental de la inteligencia artificial que permite a las computadoras y máquinas aprender mediante entrenamiento con grandes cantidades de datos. Se basa en la aplicación de algoritmos para identificar patrones en los datos, lo que les permite tomar decisiones y hacer predicciones precisas. Un ejemplo común de esta técnica se puede observar en las sugerencias o predicciones que aparecen en los motores de búsqueda en la web, donde los algoritmos utilizan los datos recopilados de sus búsquedas anteriores para establecer patrones y optimizar las sugerencias futuras (Rouhiainen, 2018).

De acuerdo con Vahid Mirjalili (2019) el aprendizaje automático se subdivide en tres conjuntos: aprendizaje supervisado, no supervisado y reforzado.

El aprendizaje supervisado utiliza datos de entrenamiento que han sido previamente etiquetados y organizados, de esta manera realizan predicciones sobre datos futuros y como se debe clasificar la información. Este método requiere de la participación humana para realizar su entrenamiento.

Por otra parte, el aprendizaje no supervisado, a diferencia del anterior, el algoritmo no necesita datos previamente organizado para saber cómo debería clasificarse la nueva información ingresada, debe encontrar la forma de clasificarla por su cuenta, sin la necesidad de la intervención humana.

Por último, el aprendizaje por refuerzo, toma decisiones en base a la experiencia de los aciertos. El algoritmo se entrena a partir de un proceso de prueba y error, donde obtiene una recompensa o castigo por cada decisión tomada. A partir de esto, el algoritmo recibe una retroalimentación sobre su desempeño, optimizando las respuestas positivas y logrando mejores sugerencias y predicciones.

Actualmente el aprendizaje automático está siendo utilizado para resolver problemas complejos con grandes cantidades de información, por lo que surge un subcampo de este denominado aprendizaje profundo (Deep Learning). El aprendizaje profundo maneja grandes volúmenes de datos mediante uso de redes neuronales, que se organizan en capas para identificar relaciones y patrones complejos de dichos datos (Azure, 2020).

Las redes neuronales son un modelo de algoritmo computacional inspirado en el funcionamiento del cerebro humano. Este modelo es utilizado para procesar datos de manera automática, y es ampliamente utilizado en procesos de clasificación y regresión. En la clasificación, se organiza la información de entrada en diferentes grupos, mientras que en la regresión se estima un posible valor de salida. Por lo tanto, se puede afirmar que el potencial de

las redes neuronales radica en su capacidad para reconocer patrones y predecir comportamientos (Sarmiento-Ramos, 2020).

Una red neuronal es una técnica de IA que busca emular el funcionamiento del cerebro humano. Esta técnica consiste en organizar múltiples unidades de procesamiento denominadas nodos, en la que cada nodo procesa y transmite información a otros nodos. De esta manera, se puede analizar grandes cantidades de datos y detectar patrones que permiten a los algoritmos realizar predicciones y tomar decisiones más acertadas. Las redes neuronales que están siendo utilizadas en las herramientas de IA recientemente son las de tipo transformer, esta es una arquitectura de red neuronal de aprendizaje profundo aplicada especialmente en tareas de procesamiento de lenguaje natural (NLP), se basan en la autoatención, mediante un conjunto de algoritmos que detectan como los elementos de datos se relacionan entre sí. Las redes neuronales de tipo Transformer se aplican en las herramientas de IA para comprender el texto natural ingresado por el usuario y de esta manera, generar sugerencias precisas y relevantes en función del contexto (IBM, 2021).

El procesamiento de lenguaje natural (NLP) es una técnica del aprendizaje profundo que permite a las computadoras interpretar y comprender el lenguaje humano. El NLP obtiene la información o peticiones del usuario de manera oral o escrita en cualquier idioma y da una respuesta o sugerencia en lenguaje natural. Es la tecnología principal que utilizan los asistentes virtuales, como Oracle Digital Assistant, Siri, Cortana o Alexa y en los asistentes de programación como Code Gpt y GitHub Copilot (Oracle, 2020).

El procesamiento de lenguaje natural (NLP) es una técnica del aprendizaje profundo, que estudia la interacción entre las máquinas y el humano a partir del análisis sintáctico y semántico, permitiendo a las computadoras interpretar y comprender el lenguaje humano. Se establecen reglas

de análisis de patrones sistemáticos que, combinadas con la información almacenada en diccionarios computacionales, bases de datos que contienen información sobre el lenguaje humano, como palabras, frases, significados, sinónimos, antónimos y otras características lingüísticas, se procesa el lenguaje natural a partir de voz o texto escrito (Torres, 2020).

Una vez definidos los conceptos relevantes se describirán las características y funciones de las herramientas de IA Code GPT y GitHub Copilot para realizar la comparativa entre ambas a partir de un cuadro comparativo, objetivo del presente estudio de caso.

Code GPT es una herramienta de IA desarrollada por OpenAI que utiliza GPT-3 (Generative Pre-training Transformer 3) el cual es un modelo de inteligencia artificial de última generación que utiliza el aprendizaje profundo y NLP. Esta herramienta se puede instalar como extensión en entornos de editores de código como Visual Studio Code, Sublime Text y Atom (Visual Studio, 2022).

De acuerdo con Visual Studio esta herramienta permite a los desarrolladores facilitar algunas tareas al momento de programar. Las funciones con las que cuenta son las siguientes:

- Ask CodeGPT: se abre una ventana emergente y permite realizar preguntas sobre códigos y se abrirá una nueva pestaña en el editor con la respuesta a la consulta realizada.
- Compile & Run CodeGPT: Se selecciona el código y esta función permite compilarlo y ejecutarlo, se abre otra ventana donde se muestra el resultado de la ejecución.
- Explain CodeGPT: Seleccionando el código, esta opción abrirá una nueva pestaña en el editor y explicará de manera detallada lo que hace el código.
- Refactor CodeGPT: Seleccionando el código, esta opción abrirá una nueva pestaña en el editor y refactorizará el código sugiriendo uno que realice la misma función, pero optimizado.

- Document CodeGPT: Esta opción abrirá una nueva pestaña en el editor y documentará el código de manera detallada.
- Find Problems CodeGPT: Esta opción permite encontrar errores de sintaxis o de lógica en el código que se selecciona y sugiere un nuevo código con las posibles soluciones.
- Unit Test CodeGPT: Permite realizar pruebas unitarias del código seleccionado para revisar si funciona correctamente o no.

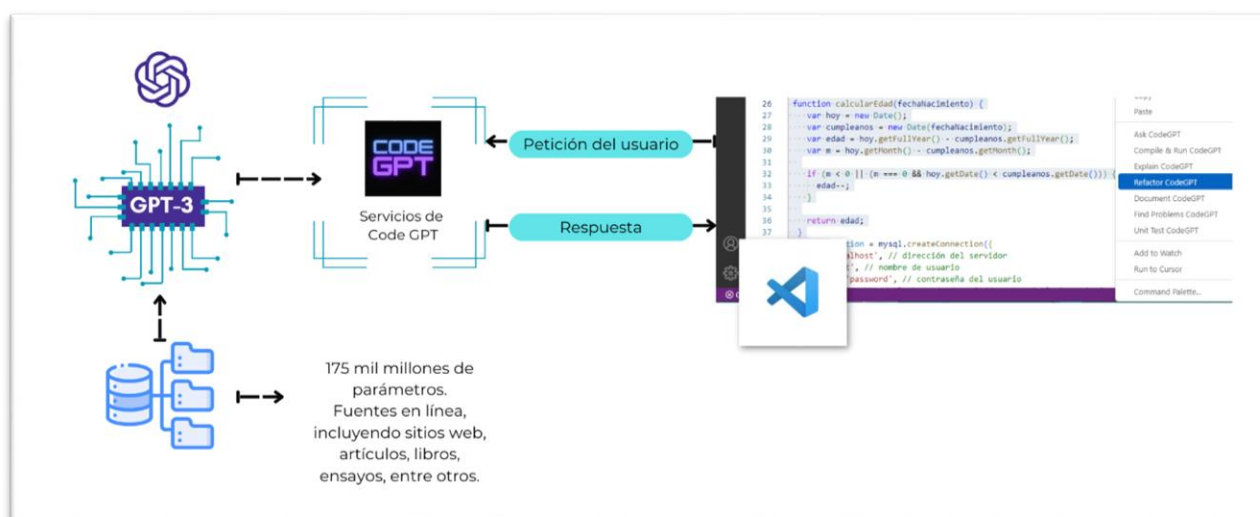


Figura 2: Funcionamiento de la herramienta Code GPT

Elaborado por Ronny Alejandro Basantes Suñiga

Code GPT Como asistente de programación ayuda a los desarrolladores en su proceso de codificación brindando algunas ventajas como:

- Está disponible de manera gratuita mediante la utilización de un API key generada al iniciar sesión en la página de oficial de OpenAI.
- Sugiere y optimiza código de manera efectiva.
- Genera código de casi todos los lenguajes de programación, por petición de forma escrita y en el idioma en el que se esté trabajando.

- Permite realizar pruebas para detectar errores en el código y corregirlos.
- Agiliza el proceso de la documentación del código.

Todas estas ventajas permiten a los desarrolladores agilizar sus tareas en la fase de programación. Sin embargo, también existen algunas desventajas:

- El código que se genera, muchas veces suele contener errores de lógica o de sintaxis, por lo que es necesario revisar y modificar una o varias líneas de código para que funcione correctamente.
- Cuando el código que se desea generar es muy extenso, se obtiene un código incompleto o con fallas.
- Muchas veces el código que se genera es obsoleto, por lo que es necesario adaptarlo a los nuevos métodos y buenas prácticas de programación.

Por otra parte, GitHub Copilot es una herramienta de IA desarrollada por GitHub y OpenAI. Utiliza Codex, una versión mejorada de GPT-3 que ha sido entrenada con millones de líneas de código de documentación y de repositorios públicos. GitHub Copilot utiliza el aprendizaje automático y el NLP para analizar el código y los patrones de escritura de los desarrolladores, de esta manera realiza sugerencias claras y precisas de código mientras se está programando. La herramienta es de paga, funciona como plugin y está disponible en varios editores de código, incluidos Visual Studio Code y Atom (Github, 2021).

La herramienta puede completar, sugerir y comentar código en tiempo real, a continuación, se detallará su funcionamiento:

- GitHub Copilot se integra con el editor de código del usuario y funciona como un complemento.

- El usuario puede escribir una línea de código o un comentario en lenguaje natural de lo que desea hacer y Copilot sugiere el código correspondiente de manera automática.
- La herramienta utiliza el contexto del código existente y los patrones de programación para ofrecer sugerencias precisas según lo que se esté realizando.
- El usuario puede aceptar o rechazar las sugerencias de Copilot y también puede modificarlas si lo requiere.



Figura 3: Funcionamiento de la herramienta Github Copilot

Elaborado por Ronny Alejandro Basantes Suñiga

Algunas de sus ventajas son las siguientes:

- Copilot es capaz de aprender de las correcciones y cambios que realiza el programador en el código, lo que le permite ofrecer sugerencias más precisas y relevantes en el futuro.
- GitHub Copilot permite reducir el tiempo que el programador dedica a escribir código manualmente.

- La herramienta puede ayudar a los programadores a resolver problemas complejos y tareas repetitivas proporcionando soluciones más eficientes y efectivas.
- Copilot puede aprender del código existente y adaptarse a las preferencias y estilos de codificación del programador.

Esta herramienta también presenta algunas desventajas:

- Al encontrarse en una etapa temprana de desarrollo, puede que el código generado no sea completamente preciso.
- GitHub Copilot puede generar código con errores, esto podría causar vulnerabilidad en el software o sistema que se esté desarrollando, si no se tienen conocimientos de seguridad en el desarrollo de software.
- Copilot no tiene un juicio crítico como lo tiene un programador humano, los programadores deben revisar y realizar pruebas al código antes de utilizarlo.

En síntesis, Code GPT y GitHub Copilot son herramientas de asistencia de programación que utiliza la inteligencia artificial para ofrecer funciones que facilitan la tarea de los programadores en la escritura de código. Las herramientas pueden optimizar tiempo y ayudar a resolver problemas simples y complejos, pero también tienen algunas limitaciones y riesgos que los programadores deben tener en cuenta al utilizarlas.

A continuación, se presenta un cuadro comparativo de estas dos herramientas tomando en cuenta diversos parámetros:

Parámetro	Code GPT	GitHub Copilot
Tecnología usada	Utiliza GPT-3	Utiliza Codex, versión mejorada de GPT-3
Nivel de entrenamiento	Entrenada con 175 mil millones de parámetros. El conjunto de datos utilizado para entrenar el modelo se basa en textos recopilados de una variedad de fuentes en línea, incluyendo sitios web, artículos, libros, ensayos, entre otros.	El modelo de lenguaje de GitHub Copilot ha sido entrenado con 6 mil millones de parámetros de una gran cantidad de datos de programación de repositorios públicos, que incluyen millones de líneas de código fuente, comentarios de código y documentación.
Funcionalidad	Genera, sugiere, explica, refactoriza y documenta código. Además, permite realizar pruebas y encontrar errores lógicos o de sintaxis.	Permite generar código a partir de comentarios, autocompletar código entendiendo el contexto existente y sugiere fragmentos de código personalizados adaptándose al estilo de código del programador.
Accesibilidad	Es de uso gratuito teniendo una cuenta de OpenAI y utilizando la API key.	Está disponible para su uso en los planes de suscripción de GitHub. Es de uso gratuito para estudiantes y docentes con correo institucional.
Nivel de conocimiento necesario	Se requiere un nivel de conocimiento básico en programación.	Se requiere un nivel de conocimiento básico en programación.
Lenguajes de programación compatibles	Permite trabajar con la mayoría lenguajes de programación: C#, C++, Go, Java, JavaScript, TypeScript, Python, Ruby, PHP, Swift, Kotlin, Rust, SQL, entre otros.	Actualmente permite trabajar con: C#, C++, Go, Java, JavaScript, TypeScript, Python, Ruby, PHP, Swift, Kotlin, Rust, SQL, entre otros.
Editores de código compatibles	Visual Studio Code, Sublime Text y Atom	Visual Studio, Visual Studio Code, Neovim y JetBrains IDE

Sistemas operativos compatibles	Windows (7, 8, 10 o posterior) macOS (10 o posterior) Linux (Ubuntu, Debian, Fedora, Red Hat Enterprise Linux, entre otros)	Windows (7, 8, 10 o posterior) macOS (10 o posterior) Linux (Ubuntu, Debian, Fedora, Red Hat Enterprise Linux, entre otros)
--	---	---

Tabla 1: Cuadro comparativo de las herramientas de IA Code GPT y GitHub Copilot

Elaborado por: Ronny Alejandro Basantes Suñiga

Aplicando la técnica de la entrevista con un enfoque cualitativo se logró recolectar información relevante y complementaria sobre el uso de las herramientas Code GPT y Github Copilot en el la fase de programación del desarrollo de software.

La entrevista se realizó de manera virtual a dos profesionales del área del desarrollo de software. El Ing. Livingston Rodríguez Bone, Encargado de desarrollo de software del Gobierno Autónomo Descentralizado de la Provincia de Orellana (GADPO) y a la Ing. Kathy Sánchez, jefa de proyectos informáticos de la empresa de desarrollo de software “Links” de la ciudad de Guayaquil, de lo cual se obtuvo lo siguiente:

Ambos profesionales coinciden que es importante incorporar el uso de herramientas de IA en la fase de programación del desarrollo de software porque permite ahorrar tiempo, ya que facilitan la codificación y las búsquedas de fragmentos de código en tiempo real. En cuanto a su experiencia, solo han trabajado con la herramienta Github copilot en sus proyectos para realizar cálculos básicos y en los códigos repetitivos, no obstante, si conocen la herramienta Code GPT. Los beneficios que han encontrado en esta herramienta son los de poder encontrar fragmentos de código mucho más rápido que buscarlos de manera manual o revisar documentación. Con respecto a la calidad y precisión opinan que estas herramientas no son del todo precisas, sin embargo, se puede adaptar el código al proyecto con el que se esté trabajando. Las limitaciones que han

encontrado en estas herramientas son que muchas veces el código que se genera es obsoleto, para versiones antiguas o que no cumplen con los métodos de las buenas prácticas de programación, es por eso que recomiendan revisar el código que se genera y realizar pruebas para evitar vulnerabilidades. Opinan que estas herramientas son accesibles para cualquier tipo de usuario con conocimientos básicos en el manejo de lenguajes de programación, especialmente son muy útiles para programadores JR.

Conclusiones

Se logró Comparar las herramientas de IA (Inteligencia Artificial) Code GPT y GitHub Copilot como asistentes de programación en el desarrollo de software y se obtuvieron las siguientes conclusiones:

Code GPT y GitHub Copilot tienen un nivel de entrenamiento muy alto, ambas utilizan técnicas de aprendizaje profundo basadas en el procesamiento del lenguaje natural y han sido entrenadas con una gran cantidad de datos, lo que les permite generar código de alta calidad y precisión. En general, ambas herramientas representan un gran avance en el campo de la inteligencia artificial para la generación de código automatizado.

Tanto Code GPT como GitHub Copilot ofrecen funciones que permiten optimizar tiempo y esfuerzo en la fase de programación. Code GPT genera, sugiere, explica, refactoriza, documenta y prueba código, mientras que GitHub Copilot se centra en autocompletar código, adaptándose al estilo del programador y entendiendo el contexto del código existente. Esto las convierte en herramientas eficientes en el desarrollo de software.

Code GPT y GitHub Copilot permiten trabajar con una amplia variedad de lenguajes de programación y están disponibles para algunos editores de código que pueden usarse en diversos sistemas operativos como Windows, MacOS y Linux. Esto las convierte en herramientas útiles y versátiles para programadores de diferentes ámbitos, con distintas preferencias y distintos tipos de proyectos.

Una de las limitaciones que presentan estas herramientas, en ocasiones, es la sugerencia de código obsoleto o que no cumple con las buenas prácticas de programación. Esto se debe a que estas herramientas se basan en el análisis de grandes cantidades de datos de código existente, incluyendo código que puede estar desactualizado o con presencia de errores.

Referencias

- Avila-Tomás, J. F.-P.-V. (2020). La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica. *Atención Primaria* , 778-784.
- Azure. (2020). *Microsoft Azure*. Obtenido de Microsoft Azure Web site: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-deep-learning/>
- ENAE. (13 de Abril de 2021). *ENAE Business School*. Obtenido de ENAE: <https://www.enaes.es/blog/la-inteligencia-artificial-en-nuestra-vida-diaria>
- Github. (2021). *Github*. Obtenido de <https://docs.github.com/es/copilot/overview-of-github-copilot/about-github-copilot>
- IBM. (2019). *IBM*. Obtenido de IBM Web site: <https://www.ibm.com/es-es/topics/software-development>
- IBM. (17 de Agosto de 2021). *IBM*. Obtenido de IBM Web site: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>
- Intelequia. (28 de Noviembre de 2020). *Intelequia ciclo de vida del software*. Obtenido de Intelequia: <https://intelequia.com/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
- Oracle. (2020). *Oracle*. Obtenido de Oracle Web site: <https://www.oracle.com/mx/artificial-intelligence/what-is-natural-language-processing/>
- Rouhiainen, L. P. (2018). Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro. En L. P. Rouhiainen, *Lasse Petteri Rouhiainen* (pág. 18). España: Planeta, S.A.
- Sarmiento-Ramos, J. L. (2020). Aplicaciones de las redes neuronales y el deep learning a la ingeniería biomédica. *Revista UIS Ingenierías*, 1-18.
- Solera, S. (24 de Marzo de 2020). *Occam*. Obtenido de Occam Web site: <https://www.occamagenciadigital.com/blog/conoce-las-fases-de-un-proyecto-de-desarrollo-de-software>
- Torres, M. M.-B. (2020). Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural. *Revista Politécnica*, 85-96.
- Vahid Mirjalili, S. R. (2019). *Python Machine Learning*. España: MARCOMBO, S.A.
- Vélez, M. I. (2022). *Conceptos fundamentales y uso responsable de la Inteligencia Artificial en el sector público. Informe 2*. Caracas: CAF.
- Visual Studio. (2022). *Visual Studio*. Obtenido de <https://marketplace.visualstudio.com/items?itemName=gencay.vscode-chatgpt>

Anexos

Anexo 1. Guía de la entrevista



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA



Tema: Estudio comparativo de las herramientas de IA (Inteligencia Artificial) CODE GPT y GITHUB COPILOT como asistentes de programación en el desarrollo de software.

Dirigida:

Ocupación:

Objetivo: Analizar el uso de herramientas de Inteligencia Artificial como asistentes de programación en el desarrollo de software.

Fecha:

1. ¿Cree que es importante incorporar nuevas herramientas de Inteligencia Artificial (IA) a la fase de programación durante el desarrollo de software? y ¿Por qué?
2. ¿Cuál es su experiencia con herramientas de inteligencia artificial para la codificación en el desarrollo de software?
3. ¿Qué beneficios encuentra en el uso de asistentes de programación como Code GPT y GitHub Copilot en la programación?
4. ¿Cuál es su opinión sobre la precisión y calidad de estas herramientas de IA en la generación de código?
5. ¿En qué tipo de proyectos ha utilizado o utilizaría herramientas de IA como GPT o GitHub Copilot?
6. ¿Cómo considera que estas herramientas de IA pueden mejorar el proceso de desarrollo de software?
7. ¿Considera que existen limitación en el uso de GPT o GitHub Copilot?
8. ¿Qué precauciones cree que se deben tomar al utilizar estas herramientas para evitar errores en el código generado?
9. ¿Qué opinión tiene acerca del potencial de estas herramientas de IA para el desarrollo de software en el futuro?
10. ¿Cree que estas herramientas de IA son accesibles para todos los desarrolladores o se requiere de conocimientos especializados para utilizarlas de manera adecuada?

Firma

Anexo 2. Recolección de la información



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA



Tema: Estudio comparativo de las herramientas de IA (Inteligencia Artificial) CODE GPT y GITHUB COPILOT como asistentes de programación en el desarrollo de software.

Dirigida: Ing. Livingston Rodríguez Bone

Ocupación: Encargado de desarrollo de software del Gobierno Autónomo Descentralizado de la Provincia de Orellana (GADPO)

Objetivo: Analizar el uso de herramientas de Inteligencia Artificial como asistentes de programación en el desarrollo de software.

Fecha: 07/03/2023

1. **¿Cree que es importante incorporar nuevas herramientas de Inteligencia Artificial (IA) a la fase de programación durante el desarrollo de software? y ¿Por qué?**

“Si, porque ahorra tiempo y ayuda a escribir código más claro y comprensible para el resto de integrantes del proyecto.”

2. **¿Cuál es su experiencia con herramientas de inteligencia artificial para la codificación en el desarrollo de software?**

“Utilicé Github copilot para realizar cálculos básicos y para búsquedas en arreglos multidimensionales.”

3. **¿Qué beneficios encuentra en el uso de asistentes de programación como Code GPT y GitHub Copilot en la programación?**

“Permite buscar fragmentos de código más rápido y más preciso a diferencia de realizar búsquedas manuales en motores de búsqueda, además de que lo integra directamente en el código que se está trabajando.”

4. **¿Cuál es su opinión sobre la precisión y calidad de estas herramientas de IA en la generación de código?**

“Estas herramientas de IA son muy buenas para buscar fragmentos de código, además de que funcionan muy bien con el español”

5. ¿En qué tipo de proyectos ha utilizado o utilizaría herramientas de IA como Code GPT o GitHub Copilot?

“Para el desarrollo de sistemas de información local (SIL) el cual funciona con JavaScript y plugin para cargar mapas en tiempo real desde un geoservidor.”

6. ¿Cómo considera que estas herramientas de IA pueden mejorar el proceso de desarrollo de software?

“Mejoran el proceso ofreciendo búsquedas y fragmentos de código en tiempo real, además de presentarlos en el idioma nativo del desarrollador.”

7. ¿Considera que existen limitación en el uso de Code GPT o GitHub Copilot?

“Una de las limitaciones es que muchas veces puede ofrecer códigos para versiones antiguas que ya no cumplen los estándares de los métodos actuales de las buenas prácticas de programación.”

8. ¿Qué precauciones cree que se deben tomar al utilizar estas herramientas para evitar errores en el código generado?

“Se debe revisar todo el código generado por este tipo de herramientas o al menos auditar los resultados obtenidos con el fin de evitar brechas de seguridad.”

9. ¿Qué opinión tiene acerca del potencial de estas herramientas de IA para el desarrollo de software en el futuro?

“Si se continua con el desarrollo de estas herramientas a un nivel más heurístico generará mejores resultados, además de que encontrará fragmentos de código más avanzados.”

10. ¿Cree que estas herramientas de IA son accesibles para todos los desarrolladores o se requiere de conocimientos especializados para utilizarlas de manera adecuada?

“Son accesibles para cualquier tipo de usuario con conocimientos básicos de manejo de lenguajes de programación.”



Firma



Tema: Estudio comparativo de las herramientas de IA (Inteligencia Artificial) CODE GPT y GITHUB COPILOT como asistentes de programación en el desarrollo de software.

Dirigida: Ing. Kathy Sánchez

Ocupación: Jefa de proyectos informáticos de la empresa de desarrollo de software “Links” de la ciudad de Guayaquil.

Objetivo: Analizar el uso de herramientas de Inteligencia Artificial como asistentes de programación en el desarrollo de software.

Fecha: 09/03/2023

1. **¿Cree que es importante incorporar nuevas herramientas de Inteligencia Artificial (IA) a la fase de programación durante el desarrollo de software? y ¿Por qué?**

“Si, porque facilita escribir código de manera ordenada, automática y mucho más rápido que hacerlo de manera manual.”

2. **¿Cuál es su experiencia con herramientas de inteligencia artificial para la codificación en el desarrollo de software?**

“Con Github copilot me ayudó en el proceso de programación ya que se encuentran sugerencias de código de manera precisa y permitieron adaptarlos a los proyectos con los que he trabajado.”

3. **¿Qué beneficios encuentra en el uso de asistentes de programación como Code GPT y GitHub Copilot en la programación?**

“Facilidad de encontrar las respuestas y la optimización del tiempo, además de poder ser utilizadas por programadores Jr que recién estén iniciando con nuevos lenguajes de programación.”

4. **¿Cuál es su opinión sobre la precisión y calidad de estas herramientas de IA en la generación de código?**

“No es una herramienta cien por ciento precisa, pero ayuda obteniendo código para adaptarlo usando las buenas prácticas de programación y las nuevas metodologías.”

5. ¿En qué tipo de proyectos ha utilizado o utilizaría herramientas de IA como Code GPT o GitHub Copilot?

“Lo he utilizado en el desarrollo de un ERP, especialmente en los códigos repetitivos para ahorrar tiempo.”

6. ¿Cómo considera que estas herramientas de IA pueden mejorar el proceso de desarrollo de software?

“Ayuda con las ideas, sugiere código de otros proyectos similares a lo que se está desarrollando.”

7. ¿Considera que existen limitación en el uso de Code GPT o GitHub Copilot?

“Muchas veces genera código obsoleto o con malas prácticas de programación.”

8. ¿Qué precauciones cree que se deben tomar al utilizar estas herramientas para evitar errores en el código generado?

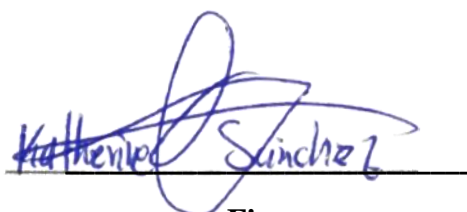
“Revisar el código generado y consultar si lo que se le está pidiendo a estas herramientas esta formulado de manera correcta para que no haya malas interpretaciones en el código.”

9. ¿Qué opinión tiene acerca del potencial de estas herramientas de IA para el desarrollo de software en el futuro?

“Tienen mucho potencial, año a año están en constante mejoras y en un futuro deberían ser usadas en todas las empresas.”

10. ¿Cree que estas herramientas de IA son accesibles para todos los desarrolladores o se requiere de conocimientos especializados para utilizarlas de manera adecuada?

“Creo que no se necesita de conocimientos especializados ya que son fácil de utilizar y los programadores Jr se adaptarían rápido a ellas.”


Firma

Anexo 3. Entrevistas realizadas de manera virtual utilizando la plataforma Google meet.

