

# CheatGPT

David Humphrey | 19 de febrero de 2023

Esta semana salí de un mundo y aterricé en otro. Comenzó mientras estaba calificando tareas en algunos de mis cursos de programación.

¡La calidad del código que estaba leyendo era increíble! Mis estudiantes estaban haciendo el tipo de cosas que hacen mis compañeros de código abierto: usando sintaxis moderna, usando enfoques funcionales en lugar de crear muchos estados innecesarios, componiendo llamadas de funciones profundamente anidadas y empleando patrones interesantes para resolver problemas.

Elogié el código que estaba viendo en varias tareas. Pero luego comencé a notar la frecuencia con la que escribía estos mismos comentarios. A pesar del hecho de que a los profesores les gusta pensar que se están convirtiendo en mejores profesores con el tiempo, es difícil imaginar que haya mejorado tanto desde el último trimestre, o el trimestre anterior, o cualquiera de las docenas de veces que he enseñado este mismo curso.

No creo que haya cambiado, ni tampoco el tema. También es difícil imaginar que estos estudiantes sean un orden de magnitud mejores. Sí, las cohortes de estudiantes pueden ser muy diferentes; ¿pero esto es diferente? Creo que hay otra explicación.

Durante mucho tiempo, me he estado preguntando en voz alta cómo la

educación superior va a lidiar con la sofisticación de ChatGPT, Copilot y otras herramientas de IA. Bueno, no me preguntes más, porque aquí es donde vivo ahora.

"Dave, estás haciendo suposiciones. ¿Puedes probar algo de esto?" Puedo, en realidad, ya que algunos envíos que requerían capturas de pantalla también incluían pestañas del navegador ChatGPT, que incluían útilmente el texto inicial del aviso. Aparentemente, ni siquiera es algo que los estudiantes sientan que necesitan ocultar.

Me encanta GPT, y he estado programando usando la API GPT-3 de Open AI durante mucho tiempo. Ha estado rompiendo mi comprensión de lo que es y no es posible, y me ha estado obligando a pensar en nuevas formas de resolver problemas técnicos. Entiendo por qué todo el mundo está emocionado de usarlo.

Sin embargo, no estoy seguro de cómo entenderlo como parte del aprendizaje de los estudiantes. Mis hijas, que están viendo que sucede lo mismo entre sus compañeros en la universidad, me animaron a pensar en esto fuera del contexto de mis propios cursos: "Nada de lo que hagas en tu clase va a cambiar lo que está sucediendo, este es un fenómeno global".

Por supuesto que tienen razón, así que seamos claros: esto está sucediendo. La gente está usando modelos de lenguaje grande (LLM) para escribir su código. Del propio marketing de GitHub:

*En junio de 2022, informamos que GitHub Copilot ya estaba generando el 27 % del código de los desarrolladores. Hoy en día, estamos viendo que esto sucede cada vez más con un promedio del 46 % del código que se construye utilizando GitHub Copilot en todos los lenguajes de*

*programación, y el 61 % entre los desarrolladores que usan Java.*

¡Piensa en todo el tiempo ahorrado! Pero, ¿y si reformulo esto un poco: "un promedio del 46% de las pruebas y tareas en línea de los estudiantes [se están] construyendo usando GitHub Copilot". Si añadimos ChatGPT a la mezcla, podríamos sustituir palabras como "ensayos" e "informes" por "código". ¿Eso es bueno o malo? ¿Innovador o regresivo? ¿Ayudando o perjudicando a nuestros estudiantes?

Muchas personas que no tienen que lidiar con las implicaciones prácticas de esto en su vida laboral me han dado todo tipo de ideas *útiles*: usar pruebas orales o en papel sin Internet; "simplemente" dejar de usar evaluaciones en las que la gente puede hacer trampa; o hacer que los estudiantes escriban sobre las respuestas de la IA, discutiendo por qué son buenas o malas.

La primera solución es difícil por muchas razones, entre otras ellas el modelo de financiación actual de las instituciones postsecundarias, que no prioriza la proporción de profesores y estudiantes necesarias para métodos de evaluación cada vez más personalizados o en tiempo real. Las clases cada vez más grandes hacen que muchas de estas buenas ideas sean poco prácticas. La facultad no tiene control sobre esto, pero por supuesto, por favor, hable con nuestro alto liderazgo. Sería genial.

La segunda solución es un reto con los estudiantes al principio de sus estudios. Enseño dos tipos de estudiantes: 1) algunos que están aprendiendo a programar en su primer año; 2) el resto en su último año, después de haber aprendido media docena de idiomas y escrito muchos programas.

En mi opinión, los estudiantes que aprenden a programar no se benefician de que la IA ayude a "eliminar el arudo" de la programación. En algún momento, tienes que aprender a programar. No puedes llegar allí evitando las inevitables dificultades de aprender a programar. "Dave, tal vez tuviste que aprender de esa manera, ¡pero ese mundo se ha ido!" Es verdad. He estado programando continuamente durante ~40 años. Seguir mi camino ineficiente y serpenteante hoy tendría ningún beneficio. Es imposible volver, y no me gustaría. El potencial de las herramientas modernas, incluida la IA, es demasiado significativo. No podemos ignorar o prohibir la IA en la educación. Pero hacer que los estudiantes lo usen para escribir literalmente sus tareas tampoco va a funcionar.

El segundo grupo de estudiantes, los que están en su último año, están en una posición diferente. (Normalmente) no estoy tratando de que entiendan OOP, diferentes formas de enumerar colecciones, o cómo y cuándo escribir una función. Tampoco estoy evaluando esto. En esta etapa de su desarrollo, se supone que pueden programar.

Para estos estudiantes, la IA proporciona algo valioso. Por ejemplo, esta semana estaba revisando el código que uno de mis estudiantes de código abierto había escrito. Mientras leía, necesitaba saber más sobre cómo se estaba utilizando una API. Hice una búsqueda en Google de la respuesta, y lo único que volvió fue la publicación del blog de este mismo estudiante (una ocurrencia común en este nivel). Mi problema me había llevado a encontrarme con los límites de la "búsqueda", que ya no era suficiente. Así que fui a ChatGPT y le pregunté sobre mis ideas y si había una mejor manera de usar esta API. Efectivamente, ChatGPT escupió 25 líneas de código que confirmaron lo que pensaba y demostraron una forma alternativa de escribir este código.

Excepto que la mitad del código fue infentado. Una de las llamadas API

que estaba usando no existe (¡ojalá lo hiciera!). No me importaba, porque no estaba buscando usar este código tal como está (fui y leí los documentos oficiales): solo quería leer sobre otros enfoques. Quería esbozar una idea.

El texto devuelto por los LLM requiere un nivel increíble de discernimiento y una lectura cercana. El hecho de que a veces puedas copiar/pegar cosas directamente desde ChatGPT puede engañar a la gente para que piense que siempre puedes. No puedes. Pero, ¿y si eso es todo lo que estás tratando de hacer? ¿Qué pasa con los estudiantes que realmente no saben lo que están haciendo y solo necesitan enviar algo para esta maldita tarea antes de las 11:59 p. m.? Te prometo que este caso de uso es más común de lo que la gente se da cuenta.

No sé cuál es la respuesta correcta. No quiero ser como muchos de los profesores que tuve, que eran escépticos sobre la capacidad de los estudiantes de incorporar de forma segura computadoras, luego CD-ROM, luego Internet, en su aprendizaje.

Es una pesada carga tener que resolver esto, y ningún otro cambio que haya visto en los últimos más de 20 años de enseñanza se ha comparado realmente con este. Los estudiantes copian/pelan desde Stack Overflow parece trillado en comparación. Hemos entrado en un nuevo mundo y va a requerir un enfoque reflexivo.

He apreciado que personas como Ethan Mollick escriban sobre cómo abrazar frente al miedo a la IA en la educación. Así es absolutamente como quiero abordar las cosas. Sin embargo, como he escrito anteriormente, no es fácil (para mí) ver cómo hacerlo con todos mis estudiantes.

Una de las cosas en las que estoy pensando es que podría tener que empezar a enseñar a los estudiantes cómo usar GPT/Codex y otros LLM en su programación. En lugar de "Tú no lo harás", podría intentar modelar algunos usos éticos y pragmáticos de la IA en la programación. ¿Cómo es la programación y el aprendizaje para programar después de GPT? Tenemos que hablar de esto. Espero que más de mis colegas escriban sobre esto, tanto en educación como en la industria.

Seneca



## David Humphrey

Profesor de CS enseñando, haciendo y escribiendo sobre el desarrollo de código abierto.

📍 Ontario rural    🔗 sitio web    🐦 TWITTER

## Starchart 0,3

◀ Hoy hemos enviado Starchart 0.3, que sienta las bases para...

## Espacio en desarrollo

Estamos empezando a sacar los proyectos de código abierto de OSD/DPS de... ▶