

PARTE 1.

APPLICATIONS-HORIZONTAL

Aplicaciones centradas en ser el apoyo para el desarrollo de otros trabajos, tecnologías...

Code & Documentation

Aplicaciones usadas para facilitar la creación de código de software así como documentación del propio código y/o de partes de la empresa.

Aquí se podrían incluir los famosos copilotos de código, como Github Copilot, CodeWhisperer o Tabnine

Text

Grupos de inteligencia artificial especialmente centrado en la interpretación, generación y/o corrección de textos

Audio & Voice

Grupos de aplicaciones centradas en la generación de audio o voz, ya sea crear canciones desde 0 con pequeños prompts como Suno o replicar voces de famosos

Image

Aplicaciones destinadas a la creación de imágenes a través de inteligencia artificial, aquí tendríamos sistemas como Dall-e o stability.ai. Son de las inteligencias artificiales que más polémica han generado en los últimos tiempos dada la facilidad de repercusión que tiene el arte

Presentación & Design

Aplicaciones dedicadas a mejorar y agilizar la experiencia de diseño y presentación visual, aquí tendríamos toda la suite de adobe así como uno de los programas más usados a día de hoy para el diseño web como es Figma, o una herramienta potentísima de presentación como puede ser canvas

Video Editing

Aplicaciones usadas para automatizar y mejorar el proceso de edición de video, facilitando tareas como el corte automático, la mejora de la calidad de imagen, la estabilización de video y la adición de efectos especiales.

Video Generation

Aplicaciones que permite la creación automática de videos a partir de texto, imágenes u otros datos. Esto incluye la generación de contenido multimedia, como presentaciones animadas, videos de marketing y material educativo.

Animation & 3D Gaming

Herramientas que agilizan la creación de animaciones y juegos en 3D mediante la automatización de movimientos de personajes, la generación de paisajes realistas y la mejora de la interacción en tiempo real... Uno de los usos más actuales seria sobre el motor Unity con apoyo de copilotos (vistos en el primer apartado) y diversas herramientas de este. Spline es otra herramienta cada vez más usada ya que facilita de maneras enormes el desarrollo de webs 3d

Search / Conversational AI

El siguiente prompt esta generado por una IA conversacional

La IA conversacional y los motores de búsqueda mejorados por IA permiten interacciones más naturales y precisas con los usuarios. Estas aplicaciones incluyen chatbots, asistentes virtuales y sistemas de recomendación que comprenden y responden de manera eficiente a consultas complejas

Fin del prompt

Estas IAs son de las que más sectores tocan, ya que facilitan extremadamente tareas tediosas y repetitivas. Son algo problemáticas en la educación ya que son extremadamente accesibles y extremadamente difíciles de detectar.

Como curiosidad dejo diversas fuentes de grandes universidades y estudios sobre justamente este tema, ya que la tenía recolectados desde hace tiempo porque a un amigo le detectaron como 30% IA un trabajo que le llevo meses hacer y tuvo que recurrir por amenazas de suspenderle.

Turnitin explicitly advises not to use its tool against students, stating that it is not reliable enough:

<https://help.turnitin.com/ai-writing-detection.htm>

“Our AI writing detection model may not always be accurate (it may misidentify both human and AI-generated text) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.”

Here’s a warning specifically from OpenAI:

<https://help.openai.com/en/articles/8313351-how-can-educators-respond-to-students-presenting-ai-generated-content-as-their-own>

This paper references literally hundreds of studies 100% of which concluded that AI text detection is not accurate:

A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions

<https://arxiv.org/abs/2310.14724>

And here are statements from various major American universities on why they won't support or allow the use of any of these "detector" tools for academic integrity:

MIT – AI Detectors Don’t Work. Here’s What to do Instead

<https://mitsloanedtech.mit.edu/ai/teach/ai-detectors-dont-work/>

UC Berkley – Availability of Turnitin Artificial Intelligence Detection

<https://rtl.berkeley.edu/news/availability-turnitin-artificial-intelligence-detection>

UCF - Faculty Center - Artificial Intelligence

<https://fctl.ucf.edu/technology/artificial-intelligence/>

Colorado State - Why you can’t find Turnitin’s AI Writing Detection tool

<https://tilt.colostate.edu/why-you-cant-find-turnitins-ai-writing-detection-tool/>

Missouri – Detecting Artificial Intelligence (AI) Plagiarism

<https://teachingtools.umsystem.edu/support/solutions/articles/11000119557-detecting-artificial-intelligence-ai-plagiarism>

Northwestern – Use of Generative Artificial Intelligence in Courses

<https://ai.northwestern.edu/education/use-of-generative-artificial-intelligence-in-courses.html>

SMU – Changes to Turnitin AI Detection Tool at SMU

<https://blog.smu.edu/itconnect/2023/12/13/discontinue-turnitin-ai-detection-tool/>

Vanderbilt – Guidance on AI Detection and Why We’re Disabling Turnitin’s AI Detector

<https://www.vanderbilt.edu/brightspace/2023/08/16/guidance-on-ai-detection-and-why-were-disabling-turnitins-ai-detector/>

Yale – AI Guidance for Teachers

<https://poorvucenter.yale.edu/AIguidance>

Alabama - Turnitin AI writing detection unavailable

<https://cit.ua.edu/known-issue-turnitin-ai-writing-detection-unavailable/>

Un último caso curioso, estas herramientas detectan como 100% generada por IA la Constitución de EEUU

Parte 2

Análisis Code & Documentation

Estas herramientas se centran en agilizar los procesos de documentación y desarrollo de tanto código como documentos. Aquí podríamos incluir a Tabnine, codewhisperer, GitHub copilot, codium.

Estas herramientas representan una revolución en el sector del desarrollo de software, no solo permiten a los desarrolladores concentrarse más en la lógica y el diseño del software evitando tareas repetitivas, sino que además fomentan una mejor documentación de los códigos, y disminuyen en gran cantidad las posibilidades de burnout por tareas repetitivas.

En mi experiencia personal y profesional así como equipos en los que he trabajado y trabajo con las herramientas, estas herramientas han agilizado en casi un 40% la velocidad de desarrollos en todo tipo de tecnologías, han permitido que los desarrolladores sigan los estándares del proyecto con menos problemas y que la calidad de la documentación incremente considerablemente.

Aunque sí que es cierto que en proyectos que trabajen con las últimas versiones de las tecnologías, como Astro 4, Spring Security 6... Estos asistentes tienen a tener errores, y las mejores aproximaciones, aunque algo peores es alimentar modelos propios con los parches nuevos o directamente no usarlos para los apartados que toquen las nuevas versiones. Mi experiencia personal me ha dado que estos asistentes es bueno usarlos para tecnologías nuevas a partir de los 9-12 meses, pero siempre teniendo cuidado e identificando que versiones usan, por ejemplo, Github copilot que es el asistente de código más usado a día de hoy, usa gpt 3.5 en su asistente de código y gpt-4 en su chat, por lo que en gran parte se encuentra limitado a los conocimientos de esos modelos, aunque Microsoft los haya entrenado con el contenido de GitHub público.

Parte 3

En general todas las herramientas que vamos a comentar tienen varias fallas en común que las empresas evitan mencionar dado que aún estamos en el nacimiento de estas tecnologías.

1. Con las últimas tecnologías o documentaciones tienden a tener problemas ya que como tal se alimentan de código que han hecho otras personas para recomendar, por lo que si algo ha salido hace 1 mes o 3, o 6 aun no habrá suficiente volumen de código para que la propia IA entienda que en esa versión o esa tecnología se hace de otra manera.

2. Son muy útiles para situaciones repetitivas, pero cuando se entra en apartados más complejos hay que tener en cuenta que en muchos casos darán respuestas incorrectas

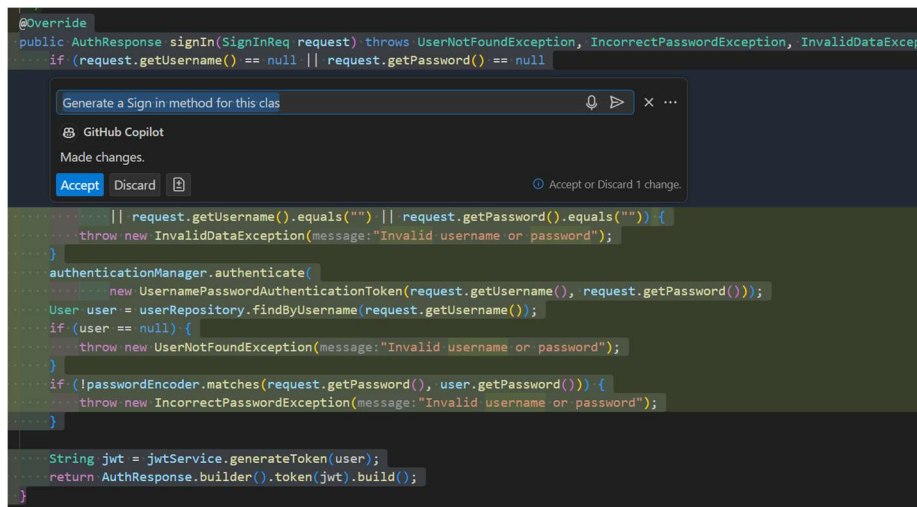
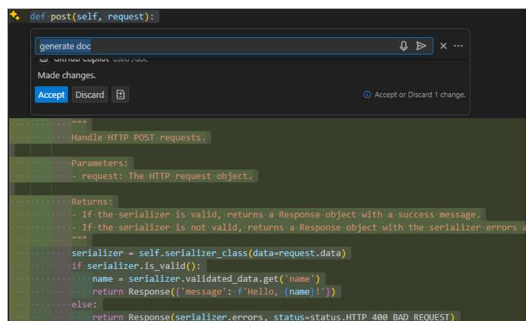
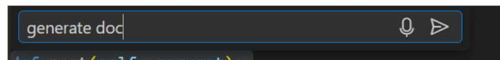
3. Son copilotos, es decir, no tienen la capacidad de funcionar por ellos solos. Ya se intentó vender el humo con aplicaciones como “Devin, el primer ingeniero de software de AI” y se ve como claramente hay que guiarlos paso a paso para cada una de las tareas, además de que en mi experiencia personal hay veces que entran en bucle.

4. Hay una frase que me gustó mucho, no recuerdo quien me la dijo, pero” Hay veces que se tarda más analizando una pull request que haciendo de 0 esa funcionalidad o corrección por tu cuenta”, esto hace referencia a que realmente como aun estas tecnologías están verdes, hacen barbaridades que se tardan más en analizar que en hacerlo tú a mano. Por ejemplo, en la demo que hicieron de Devin, la demo fue Devin creando sin querer un bug intentando arreglar algo que ya estaba arreglado y luego tratando de arreglar el bug que la propia IA había generado

Github copilot

Es el asistente de código más extendido, está implementado en toda la suite de JetBrains así como en Android Studio, Visual Studio y el propio GitHub Enterprise & workspace. Es extremadamente fácil de implementar dado que dispone de addons oficiales para todos los IDEs mencionados.

Unos ejemplos de uso:



Ha tenido bastante humo de por medio y sí que es verdad que, aunque la vendan como el culmen de los copilotos tiene los problemas al trabajar con las últimas versiones de las tecnologías, dado que primero necesita alimentarse de código hecho por otros para poder funcionar. Es decir, no tiene capacidad de innovación como tal

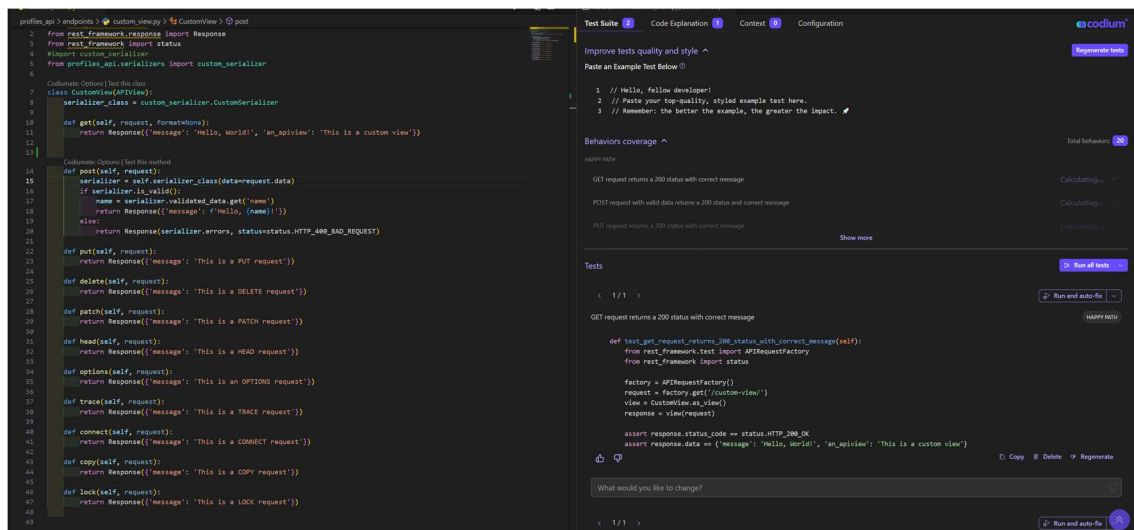
Tabnine

Tabnine es la alternativa que más está compitiendo a nivel de IDEs con copilot, especialmente en el sector java dado que Copilot no es compatible de manera oficial con eclipse, y usar plugins de terceros para estas tareas es demasiado arriesgado, en mi experiencia personal usando ambos es un poco más lento que copilot, pero están trabajando en versiones offline muy potentes.

Codium & Codiumate

La alternativa con nivel gratuito a Tabnine y Copilot, extremadamente potente, en mi uso personal le he dado especial uso al apartado de generación de test unitarios de Codiumate, automatizando una de las partes más repetitivas y tediosas del desarrollo de código

Un testeo por ejemplo de una clase básica de una api



Debuild

Una de las múltiples alternativas para crear Webapps “0 code”, no son la mejor alternativa si necesitas que tu App sea SEO friendly, pero es una herramienta extremadamente potente para crear prototipos o para páginas que no requieran ni de SEO ni de alta complejidad

AI2sql

Generador de consultas SQL, útil especialmente para gente que no tenga altos conocimientos de SQL, aunque compite con las apps de generadores de código. A mí

personalmente no me llama la atención especialmente por esa competencia, aunque se haya vendido como la herramienta que es, por que querrías usar esta aplicación cuando tienes alternativas con precios similares que te cumplen esto y mas