La matemática es computación mal enseñada: hacia una pedagogía intuicionista aumentada con IA

T. de-Camino-Beck
Escuela de Sistemas Inteligentes, Universidad CENFOTEC, tdecamino@ucenfotec.ac.cr

Introducción

La enseñanza de las matemáticas ha estado marcada, por siglos, por un enfoque mecanicista. Se aprende memorizando operaciones básicas, aplicando fórmulas y resolviendo ejercicios diseñados para ser verificados de manera rutinaria. Este modelo, heredado de la modernidad, respondía a un contexto donde la rapidez y la exactitud eran indispensables para la vida práctica en ausencia de calculadoras o computadoras. La memorización se justificaba como un modo de reducir la carga cognitiva y de facilitar la estandarización de la enseñanza, pero también instaló una confusión persistente entre velocidad y verdadero dominio matemático.

Por otro lado, la reiteración de diagnósticos y titulares que denuncian, año tras año, en casi todas las partes del mundo, "deficiencias" en los estudiantes de primer ingreso a la universidad, revela que seguimos midiendo con instrumentos mecánicos un aprendizaje igualmente mecánico, en un área donde lo mecánico ya no tiene sentido. El problema no es sólo la falta de preparación, sino la forma misma en que concebimos qué significa "saber matemáticas" y cómo evaluamos ese saber, en un mundo totalmente diferente. Las pruebas reproducen el paradigma que dicen criticar, verifican la memoria de algoritmos y la aplicación de fórmulas, pero dejan intacta la raíz del problema.

La IA generativa abre una posibilidad inédita para aprender matemáticas como un proceso en tiempo real de construcción asistida. Si aceptamos que la matemática es, en el fondo, computación, entonces la IA puede convertirse en un aliado para hacer esa computación visible, manipulable y exploratoria. En lugar de limitarse a entregar respuestas, la IA permite generar código, simular objetos, verificar cálculos y proponer variaciones en tiempo real, creando un entorno donde el estudiante construye activamente lo que estudia en diálogo con un sistema que amplifica sus capacidades. De este modo, la matemática se enseña y se aprende como un acto directo de construcción, ahora potenciado por herramientas que hacen tangible lo abstracto y convierten cada problema en una oportunidad de experimentar.

En este panorama, el intuicionismo de L.E.J. Brouwer [1] se vuelve un aire fresco frente a la enseñanza mecanicista. Brouwer decía que las matemáticas no son un conjunto de verdades esperando ser descubiertas ni una lista de fórmulas para aplicar, sino una actividad mental, es algo que existe solo cuando lo construimos. Una proposición es verdadera si la podemos demostrar, no porque alguien nos diga que lo es. Necesitamos una pedagogía donde el estudiante no se limite a repetir, sino que construya, verifique y explore. En la era de la IA generativa esto es aún más urgente, porque ahora sí tenemos herramientas que permiten llevar esa filosofía a la práctica. Podemos crear, probar y visualizar objetos matemáticos en tiempo real, experimentar con ellos y equivocarnos sin miedo. Y lo mejor, ya no tenemos que elegir entre "primero las bases" o "luego la complejidad"; podemos enseñar mezclando ambos niveles, dejando que la complejidad alimente las bases, y que las bases sostengan la exploración de lo complejo.

En este ensayo comparto algunas ideas, de manera deliberadamente informal, sobre la necesidad de un nuevo paradigma para la enseñanza de las matemáticas. Un paradigma que asuma su vínculo esencial con la computación y que aproveche el potencial de la IA generativa como medio para replantear la forma en que construimos, exploramos y damos sentido a los objetos matemáticos.

I. "Los estudiantes no tienen bases matemáticas"

Por años he visto titulares repitiendo que los estudiantes no llegan preparados para las matemáticas a la universidad (y que tienen mala comprensión de lectura). ¿Cuál ha sido la solución? Claramente ninguna, al menos en lo que respecta a la enseñanza de las matemáticas. Seguimos enseñando lo mismo, enseñando de forma mecánica, evaluando de forma mecánica, y lo hemos hecho así durante siglos.

De hecho, en Costa Rica, por ejemplo, si retrocedemos 20 años, encontramos las mismas noticias año con año. Vean por ejemplo este titular del Semanario Universidad en 2005: "Estudio de la UCR: Desastre en promoción de matemática y cálculo" [2]. Y así, casi cada año desde entonces, aparecen titulares similares [3,4,5,6,7,8,9]. Aquí aplica el viejo dicho, si hago lo mismo, obtendré lo mismo.

Se han hecho algunas cosas, en 1986 surgió MATEM, un proyecto para que estudiantes de colegio cursaran pre cálculo y cálculo universitario. Sin embargo, este programa no cambia nada fundamental en la forma de enseñanza, más bien quiere reparar algo para la universidad, no para la educación en general, y la mayoría de los colegios en este programa son privados o científicos/técnicos; y muy, pero muy pocos colegios públicos, por tanto, está lejos de ser un plan universal. De hecho, yo recuerdo en 1987 haber llevado pre-cálculo de la universidad (mate 125), y la verdad no podría decir, que había algo diferente en lo que se enseñó, o que cambió algo en mi vida académica o profesional, aunque es un contrafactual difícil de probar.

También pongo en duda las pruebas de diagnóstico que se aplican como tal. La realidad es que están diseñadas para evaluar a estudiantes formados bajo una enseñanza de la matemática mecanicista, centrada en la repetición y la memoria, una forma mecánica de hacer matemáticas. Me pregunto: ¿acaso esas quince generaciones de profesionales desde el 2005, que, según estas pruebas, tenían "deficiencias" matemáticas, hoy son malos profesionales? Probablemente no.

Es hecho las universidades no basan su admisión en estas pruebas sino en otro tipo de pruebas, si lo hicieran, dejarían afuera muchos y muchas estudiantes valiosos. El punto es que seguimos midiendo con un instrumento mecánico, un aprendizaje igualmente mecánico. ¿Afectarán esas deficiencias la posibilidad de que esos y esas profesionales sean más creativos, más emprendedores y mejores? No lo sabemos, pero en principio diría que sí. Tal vez una forma distinta de enseñar podría dar lugar a generaciones más creativas, críticas y productivas. No lo sé, pero sí creo que vale la pena averiguarlo con un nuevo paradigma de enseñanza, y más aún en la era de la IA generativa. ¡No más matemáticas mecánicas!

II. Aprendizaje Mecanicista de las Matemáticas

El aprendizaje mecanicista de las matemáticas es un enfoque en el que el estudiante adquiere procedimientos, reglas y resultados de manera repetitiva y automática, sin comprender los fundamentos conceptuales que los sustentan [10,11,12]. En este modelo, la meta principal es la ejecución correcta y rápida de algoritmos o fórmulas previamente enseñadas, casi siempre a través de ejercicios repetitivos, memorización de pasos y práctica cronometrada.

En este enfoque, el conocimiento se reduce a la aplicación de recetas, el estudiante reconoce un patrón superficial en el problema, selecciona la fórmula o procedimiento asociado y lo ejecuta, sin cuestionar por qué funciona ni explorar métodos alternativos. Esto produce una "competencia" aparente, es decir, una aparente capacidad de obtener resultados correctos. La realidad es que esta capacidad es frágil, ya que falla cuando el contexto cambia, cuando el problema requiere adaptación, o cuando el problema debe ser formulado como tal.

¿Por qué mecanicista? Durante gran parte de la historia escolar, se enseñó a memorizar sumas básicas no por un valor matemático intrínseco, sino por razones prácticas y pedagógicas propias de su época. En un mundo sin calculadoras ni computadoras, la rapidez y exactitud en operaciones mentales eran esenciales para el comercio, la contabilidad y la vida cotidiana. La memorización reducía la carga cognitiva, y permitía aplicar con fluidez algoritmos escritos lo que se ajustaba al modelo educativo mecanicista del siglo XIX y XX, basado en la repetición y la práctica intensiva como indicadores de buena formación. Además, esta forma de enseñar, facilitaba la estandarización y evaluación del aprendizaje mediante pruebas rápidas, reforzando un enfoque donde la velocidad se confundía con el dominio matemático. Esta forma de enseñar ha persistido incluso a pesar de los cambios tecnológicos [13]. ¡Nos quedamos pegados en esa época!

Puede ser que no enseñemos explícitamente a memorizar, pues conozco profesores y profesoras que ponen esfuerzo para enseñar a profundidad las matemáticas, pero el formato de las evaluaciones, la cantidad de contenidos y el tiempo, empujan a un o una estudiante a usar la memorización como una especie de estrategia de supervivencia. En un examen no hay tiempo de ponerse creativo. Si en una prueba pido resolver 20 sumas y restas en 5 minutos, un o una estudiante no tendrá tiempo de razonar cada una con estrategias novedosas, así que va a recurrir a la memoria para reducir la carga cognitiva y ganar velocidad. Esto no necesariamente significa que entienda lo que está haciendo, pero le funciona para el examen. Es el clásico "me lo aprendí de memoria para pasar".

Para poner en perspectiva, les planteo el siguiente problema, que es similar a una pregunta de una prueba de diagnóstico en geometría (Figura 1).

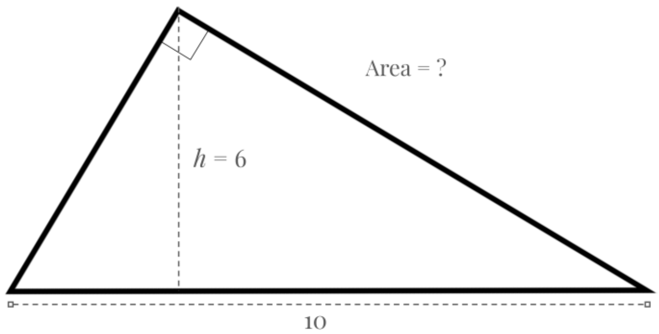

Figura 1.

¿dado el triángulo rectángulo de la Figura 1, cuál es su área? La mayoría de ustedes utilizarán la fórmula que dice que el área de un triángulo es la base multiplicada por la altura, dividido entre 2, en este caso nos daría 30, pero ese resultado es incorrecto, ¿pueden determinar ustedes por qué? Ya les explicaré luego.

El aprendizaje mecanicista no desarrolla flexibilidad ni sentido numérico, y limita la capacidad de transferir conocimientos a nuevas situaciones. Históricamente, ha sido favorecido por sistemas educativos que priorizan la estandarización, la velocidad y la evaluación masiva de resultados, más que la comprensión profunda, porque seguimos pensando en la formación escolar y colegial, como algo utilitario. Esto ocurre en prácticamente todo el mundo. En el caso del problema del triángulo, lo inmediato es usar una fórmula matemática, porque hemos sido "entrenados" para hacerlo.

III. ¿POR QUÉ NO ES 30? OTRA FORMA DE APRENDER

Este problema de triángulo lo planteó Vladimir Arnold en un libro de acertijos matemáticos [14], la pregunta es “Un triángulo rectángulo tiene un cateto de 10 pulgadas y una altura sobre ese cateto de 6. ¿Cuál es su área?”. La mayoría responde 30 sin pensarlo, incluso a mí me pasó. La realidad es que un triángulo así no puede existir. Arnold lo usaba para mostrar que las matemáticas no son sólo calcular, sino detenerse y cuestionar lo que parece evidente. Esto me lleva a un enfoque intuicionista de aprender matemáticas.

En un enfoque intuicionista, la existencia matemática de un objeto. Un triángulo, un conjunto, un número, una ecuación; depende de su posibilidad de construcción. No basta, por ejemplo, con enunciar un triángulo con ciertas propiedades, hay que poder efectivamente construirlo. Al preguntar directamente por el área, se desplaza la atención sobre la existencia, al pensar sólo en el cálculo, llevando a los y las estudiantes (o incluso a una IA) a aplicar una fórmula sin verificar si el objeto sobre el que se calcula realmente existe.

Este problema puede verse como un ejemplo de lo que él llamaría una “pregunta trivial”, o una “pregunta ilegítima” [15], que es una pregunta que se resuelve siguiendo un procedimiento mecánico ya enseñado. Primero se instruye a los estudiantes con una fórmula (área = ½·base·altura) y luego se plantea un ejercicio que solo requiere aplicarla, como si la mente funcionara como una máquina expendedora, se introduce el dato, se acciona el mecanismo y sale la respuesta.

En ese contexto, la pregunta por el área es trivial, porque se configura para obtener una respuesta directa mediante un mecanismo ya aprendido. Pero lo que realmente importaría en una educación matemática más rica no es el resultado del cálculo, sino la exploración de sí ese triángulo es posible, una pregunta no trivial que obliga a pensar, modelar, y cuestionar lo que se da por hecho. ¿Pero cómo?

Hagamos el siguiente ejercicio (invito a que lo hagan ustedes). Si yo pido a la IA, como ChatGPT o Gemini que responda la pregunta,

Prompt:

> “¿Cuál es el área de un triángulo rectángulo de base 10, que essu hipotenusa, y altura 6?”

Respuesta Gemini:

> “El área de un triángulo con base 10 y altura 6 es 30. La fórmula para calcular el área de un triángulo es (1/2) * base * altura. En este caso, sería (1/2) * 10 * 6 = 30.”

Cae en el mismo error. ¿La razón? La misma por la que profesionales y estudiantes lo hacen, el cuerpo de conocimiento sobre estos problemas es mecánico, e históricamente se ha descrito como la aplicación de una fórmula. Pero en realidad, la aplicación de una fórmula aplica sólo bajo el supuesto de que ese triángulo puede existir. Este principio de existencia ni lo está verificando el estudiante, ni la IA. Esto es válido para cualquier enunciado matemático, es decir, aplica dadas ciertas condiciones, y esas condiciones hay que chequearlas, hay que verificarlas.

El problema y el error en la aplicación del triángulo surgen del contexto, hemos “automatizado” principios matemáticos a través de fórmulas, y saber usarlas no implica necesariamente un entendimiento profundo de la matemática. Más aún, la figura del triángulo en el enunciado (Figura 1) no es un triángulo real, sino una representación de un triángulo, un simple dibujo acompañado de números. Entonces, ¿cómo sé que eso es realmente un triángulo rectángulo? La respuesta está en pasar del dibujo a una construcción “paramétrica” mediante código en computadora. Con “paramétrica” me refiero a la posibilidad de variar uno o varios parámetros o variables para generar una familia de triángulos, y así poder, en principio, calcular el área de cualquiera de ellos.

Hagamos la construcción de forma asistida con IA. Para construir triángulos en la computadora, pidamos a una IA, en este caso Claude, que construya numéricamente un triángulo rectángulo cuya hipotenusa es 10 y cuya altura sobre ese cateto es 6, como en el dibujo,

Prompt:

> "Escribe un código interactivo que dibuje un triángulo rectángulo acostado sobre su hipotenusa, cuya longitud siempre sea 10 unidades.
>
> Agrega un slider, que vaya de 0 a 8, que permita modificar la altura del triángulo (distancia desde la hipotenusa hasta el vértice del ángulo recto). Al mover el slider, recalcula y ajusta dinámicamente las longitudes de los catetos para mantener fija la hipotenusa en 10 unidades.
>
> Dibuja también la altura como un segmento desde la hipotenusa hasta el vértice del ángulo recto. Incluye etiquetas o líneas que ayuden a visualizar qué parte corresponde a la hipotenusa, los catetos y la altura."

Noten que el prompt mismo, evidencia la necesidad de “entender” lo que quiero. Cuando escribo este prompt, lo que realmente estoy haciendo es darle a la computadora un conjunto de instrucciones que convierten al triángulo en algo computable. No es una figura asumida ni dibujada a mano, sino el resultado de un procedimiento preciso que pido a la computadora a través de una descripción en lenguaje natural, donde se genera un código en javascript, que es un lenguaje de programación, que luego la máquina ejecuta paso a paso. Al definir la hipotenusa fija, la altura variable y las relaciones geométricas que unen todos sus lados, estoy creando junto con la IA un algoritmo que, para cualquier valor válido de la altura, construye el triángulo y lo representa.

La Figura 2 muestra el resultado, el cual pueden accesar en [16]. Si movemos el slider hasta una altura de 4.0, veremos que obtenemos el largo de cada cateto para construir el triángulo, hipotenusa de 10, y catetos de 8 y 2. Sin embargo, si seguimos aumentando la altura, algo curioso sucede, después de 5.0, el triángulo desaparece. En lugar de los valores de los catetos, aparece “NaN” (Not a Number), que en computación significa que el resultado no es un número válido. ¿Por qué ocurre esto? Le pedí a Claude que explicara el problema, y me mostró la parte del código donde surge:

```
const discriminante = Math.sqrt(hipotenusa
* hipotenusa - 4 * altura * altura);
```

Sin necesariamente saber Javascript, cualquiera puede leer esa porción del código. Este término es el discriminante, usado para calcular el largo de los catetos. Si observamos la ecuación, veremos que hay un punto en el que el término (`4 * altura * altura`) supera a (`hipotenusa * hipotenusa`). Esto produce una raíz cuadrada negativa (`Math.sqrt`), algo que en este contexto no tiene sentido matemático. Dicho de otra forma, a partir de cierta altura, que es aparentemente 5, no existe un triángulo rectángulo como el del dibujo, por lo que tampoco se puede calcular su área.

Figura 2.

En todo este proceso lo que hicimos fue 1) construir un triángulo mediante código asistido con IA, 2) explorar cuáles triángulos son posibles, 3) encontrar un límite y un error numérico, es un ejemplo de un enfoque intuicionista para entender el problema. Lograrlo sin asistencia de IA generativa sería muy lento, y hacerlo completamente a mano resultaría impráctico, si no imposible pues no se pueden hacer figuras dinámicas a mano. En paralelo, se entendieron conceptos básicos y exploraciones complejas reales.

IV. Enseñanza Intuicionista de las Matemáticas

La matemática intuicionista es una forma de entender y hacer matemáticas que parte de una idea sencilla, en matemáticas, algo “existe” sólo si podemos construirlo. No basta con decir “en teoría debería existir” o demostrarlo por contradicción; para el intuicionismo, debemos ser capaces de mostrar cómo hacerlo paso a paso. Este enfoque, propuesto por el matemático L.E.J. Brouwer a principios del siglo XX, ve las matemáticas como una actividad mental creativa, más parecida a construir con piezas de Lego que a simplemente aceptar un objeto que ya estaba ahí [1, 17]. Por eso, se aleja de demostraciones puramente abstractas que dependen de infinitos o de nociones que no podamos realizar de forma concreta. En la enseñanza, esto se traduce en invitar a los estudiantes a “hacer” las matemáticas, con una computadora, con programación, de manera que cada concepto se entienda como algo construido, no como un conjunto de fórmulas para memorizar.

En un enfoque intuicionista de la enseñanza de las matemáticas, el problema del triángulo de Arnold no se resolvería aplicando mecánicamente una fórmula para el área, sino explorando la posibilidad misma de construir el triángulo con las condiciones planteadas. Desde esta perspectiva, la existencia de un objeto matemático sólo es válida si puede ser efectivamente construido, por lo que él o la estudiante, asistido por herramientas como la programación o la simulación, debe intentar generar el triángulo y verificar si es posible. El énfasis no está en llegar a un número final, sino en el proceso de construcción, en las limitaciones que surgen y en el momento en que la figura deja de ser geométricamente viable. Así, el aprendizaje se centra en la interacción directa con el objeto matemático como algo computable, fomentando una comprensión profunda que nace de la experiencia y no de la mera aplicación de reglas memorizadas.

Con este enfoque, la IA generativa añade la capacidad de crear y ajustar código de manera asistida y casi instantánea, permitiendo que las construcciones matemáticas se realicen en tiempo real. Esto no sólo acelera el proceso de traducir una idea a un objeto computable, sino que también facilita la verificación inmediata de los cálculos dentro del propio código. Ya no es construcción de código bajo el mismo principio mecánico, sino el código con intención. El o la estudiante debe poder “leer” el código, y en la lectura aprende. Así, conceptos que antes requerían largas operaciones manuales o cálculos tediosos pueden explorarse de forma práctica y visual, convirtiéndose en experiencias tangibles incluso aquellas construcciones matemáticas que, por su complejidad, serían poco realistas de realizar a mano.

Claro, en la pregunta del área del triángulo, uno podría pensar que la pregunta instiga un uso mecánico de la fórmula, pero en principio, si se enseña de otra manera, un o una estudiante se cuestiona la pregunta, pues antes de calcular un área, sentiría la necesidad de hacer la construcción, con programación, del triángulo primero. Pero de todas maneras un o una docente, en este enfoque, haría una pregunta diferente, una pregunta como,

> “Un triángulo rectángulo tiene un cateto de 10 pulgadas y una altura trazada sobre ese mismo cateto de 6 pulgadas. Considera esa situación y reflexiona sobre cómo podrías encontrar su área.”

Que invita a una reflexión, poniendo la idea de construcción antes del cálculo del área.

V. Matemática Como Computación

En la educación muchas veces proponemos aprendizaje a través de los ejercicios, y creamos ejercicios que son en principio realizables a mano. Esto va desde las primeras clases de matemática, hasta cursos avanzados. Pero ¿quién determina qué está correcto y qué no?, ¿cómo verificamos? En los ejercicios de un curso, o examen, hay un resultado conocido, los libros tienen las respuestas en la parte de atrás, o alguien revisa que lo haya hecho bien. La vida real no es “un curso”, ni las complicaciones de la vida son “exámenes”, ni hay alguien o algo que revise lo que hacemos, debemos ser nosotros mismos que verificamos. Si hacemos matemáticas o computación de la vida real, ¿cómo sabemos si un problema que se me presenta, y la solución que proponga, está bien o está mal?

En un curso básico de primaria, por ejemplo, le pediríamos a un estudiante que realice 7 + 2. El resultado es conocido de antemano, porque es trivial. Sabemos el resultado, no porque conocemos los fundamentos de la suma de números enteros, sino porque lo memorizamos. La única verificación es la memoria, o porque está escrito en un libro, o porque el profesor o profesora me lo dice. Luego, más avanzado, pedimos 72 +37, donde además de memorizar la suma de números de un dígito, memorizamos un procedimiento para alinear, ambos números y hacer la suma con acarreo. Esta operación también es trivial. Pero qué tal si preguntamos: ¿987654321234567890 + 1234567890987654321? Esto no es algo sencillo de hacer a mano, y es una suma que haríamos con una calculadora, pero entonces, ¿qué sentido tiene el hacer cálculos a mano del todo?

Bueno, en principio, usar sumas simples permite desarrollar estructuras cognitivas fundamentales [18], pero pienso que tiene poco que ver con la suma misma. Por otro lado, la idea fundamental de enseñar sumas simples y memorizarlas, es que reduce la carga cognitiva al abordar problemas complejos, porque libera memoria de trabajo [19]. Pero ¿por qué nos quedamos siempre en estrategias que reducen la carga cognitiva? Es definitivo que las operaciones simples con objetos concretos actúan como puente hacia la manipulación simbólica, clave para entender conceptos abstractos como ecuaciones [20], pero confundimos el fin con el medio. En el proceso, nos quedamos en la matemática mecanicista, que reduce siempre carga cognitiva, y nunca, empujamos para que más bien aumente la carga cognitiva, es decir, que los y las estudiantes piensen. Nunca llegamos a la “enseñanza generativa” donde los y las estudiantes requieren producir algo nuevo. Más aún, prácticamente nunca llegamos a simular entornos reales y mucho menos al desarrollo de la metacognición.

En la práctica, esa “reducción de carga cognitiva” permanente puede derivar en enseñanza trivializada, procesos mecánicos y memorísticos, y una falta de oportunidades para que el estudiante se enfrente a la complejidad real [21]. Cuando todo se fracciona en micro-pasos, en “bases” sin contexto, se entrena más la ejecución repetitiva que el pensamiento. Esto produce egresados con destreza en rutinas, pero sin flexibilidad para situaciones nuevas, reales o complejas.

Pienso que las bases, y la complejidad pueden y deben crecer juntas. Una alternativa es apostar por la complejidad temprana con mediación, presentando problemas auténticos desde el inicio y permitiendo que él o la estudiante utilice herramientas externas, como inteligencia artificial, software especializado o materiales manipulativos, para abordar aquello que aún no domina mecánicamente. De este modo, las tareas se diseñan para integrar lo que el estudiante “todavía no sabe bien” con lo que ya sabe, haciendo que la carga cognitiva productiva sea inevitable y que el aprendizaje ocurra en un contexto donde la complejidad no se pospone, sino que se vive y se construye desde el principio. El ejemplo del triángulo que mostré con anterioridad, que se resuelve a través de la generación de un código, es un buen ejemplo de esto.

Si concebimos la matemática como construcción o computación, no como repetición, entonces “las bases” no son un prerrequisito para ver problemas complejos, sino que se forman en el mismo acto de construir y reconstruir esas soluciones. El ejemplo del triángulo con la construcción en código de triángulos, puede ser un ejemplo, donde a través de la construcción, por ejemplo, se aprendió o encontró el término discriminante, o ciertas operaciones matemáticas, el cual él o la estudiante pueden explorar de forma independiente.

VI. El valor de la IA Generativa

Al explorar nuevas formas de enseñar matemáticas, es inevitable preguntarse cómo podría integrarse la inteligencia artificial generativa en estos procesos sin reducirla a un simple instrumento de automatización o corrección mecánica. Podríamos concebir a la IA como una herramienta activa, como un agente intencional formal [22]. Con un agente intencional formal, me refiero a que puede comportarse como si tuviera propósitos o intenciones, aunque no tenga conciencia ni emociones. Es decir, puede tomar decisiones guiadas por metas específicas y que adapta sus acciones según una meta. Por ejemplo, si el objetivo es ayudar a un estudiante a entender un concepto matemático, la IA no sólo corrige errores, sino que puede formular preguntas, ofrecer pistas, construir o cambiar su enfoque según el progreso del estudiante.

Todo esto se hace de forma estructurada y programada, permitiendo que la máquina participe activamente en procesos que antes eran solo humanos, como enseñar, guiar o dialogar con sentido. Bajo esta perspectiva, su papel en el aula no se limita a proporcionar respuestas correctas, sino que puede asumir una función activa como interlocutor, capaz de reconocer fines, formular preguntas significativas y participar en la construcción compartida del conocimiento.

Desde una mirada intuicionista, en la que la matemática no se descubre, sino que se construye, una IA podría modelar al estudiante no como un autómata que ejecuta reglas, sino como un agente cognitivo que actúa con intención. En este marco, las construcciones matemáticas no emergen de una lógica externa impuesta, sino que son fruto de una interacción concreta, paso a paso, con la experiencia de él o la estudiante. La IA, entonces, no verificaría simplemente la corrección de un resultado, sino que los y las estudiantes verifican el resultado con la IA, y es un proceso de construcción y exploración compartida [23].

Lo que se vuelve verdaderamente interesante es cómo esta IA podría operar a través de un lenguaje funcional e intencional, diferente del lenguaje puramente simbólico que domina la tradición matemática formal. No se trata de deducir teoremas desde axiomas abstractos, eso tiene su lugar en otros contextos, sino de acompañar la creación progresiva de entidades matemáticas en diálogo con las intuiciones del estudiante, a través de códigos de programación, como en el ejemplo del triángulo. De esa manera, la inteligencia artificial se transforma en un mediador entre la intención y la formalización. Permite establecer una relación entre la imaginación matemática y su verificación computacional. Así, el aula, o el trabajo matemático en educación, deja de ser un escenario de reproducción de verdades preestablecidas y se convierte en un taller donde el conocimiento emerge como acto colectivo de construcción.

VII. Modelo de Enseñanza Computacional y Mediada por IA

La estrategia de enseñanza de una matemática entendida como computación la propongo en cuatro fases: i) Inmersión con complejidad guiada, ii) Consolidación mediante el uso, iii) Integración y variación, y iv) Autonomía creativa. A continuación, detallo cada una de estas fases con un ejemplo sencillo, aplicable tanto en la escuela como en el colegio o la universidad.

Fase 1 Inmersión con complejidad guiada. En la primera fase, el estudiante entra directamente en la complejidad. No hay una larga espera para "aprender las bases", desde el primer día enfrenta un problema auténtico que por sí solo, excede sus habilidades actuales. La IA generativa se convierte en un compañero que escribe fragmentos de código para representar la situación, ya sea graficando, simulando o calculando. La matemática, en este punto, se revela como algo que puede construirse, ejecutarse y verse en acción, se ve a través de un código escrito, no a través de la notación tradicional de las matemáticas. Un ejemplo puede ser similar al problema del triángulo que hemos estado discutiendo, un triángulo rectángulo cuya altura se ajusta con un control deslizante, usando código generado por la IA para recalcular y mostrar los catetos en tiempo real.

Fase 2 Consolidación a través del uso. En la segunda fase. El estudiante ahora explora el código, modifica y completa fragmentos, entendiendo el papel de cada línea, con explicaciones asistidas por la IA. De esa forma cada paso computacional y cada línea corresponde a un razonamiento, y cada error en el código se convierte en una oportunidad para depurar ideas. Así, las bases se consolidan como una consecuencia de enfrentarse a lo complejo. Un ejemplo sería tomar un código generado por IA para resolver un sistema de ecuaciones y adaptarlo para trabajar con un conjunto distinto de coeficientes y/o condiciones iniciales.

Fase 3 Integración y variación. En la tercera fase, la experiencia se expande a otros contextos. El estudiante reutiliza estructuras de código y modelos que ya había trabajado, pero ahora en dominios diferentes, de la geometría a la biología, de un modelo de crecimiento poblacional a la simulación de redes de datos, etc. La IA sigue siendo una aliada, pero la combinación de piezas, la adaptación de funciones y la integración de conceptos corren a cargo del estudiante. La matemática se muestra como un lenguaje flexible que puede adoptar muchas formas computacionales. En esta fase se podría, por ejemplo, adaptar un modelo de crecimiento poblacional en Python para simular la propagación de un virus en una red de transporte público.

Fase 4: Autonomía creativa. El estudiante se mueve con autonomía creativa. No espera que la IA "resuelva" el problema, sino que la utiliza como colaborador para explorar ideas, optimizar soluciones y ampliar la complejidad de sus proyectos. Toda la matemática que produce existe como código ejecutable, un programa es más que un sistema que responde, sino que permite representar, simular y explicar diferentes conceptos. Como ejemplo, se puede crear una simulación interactiva de un ecosistema marino, integrando datos reales de temperatura, salinidad y corrientes, para modelar el desplazamiento de especies a lo largo del tiempo.

Creo que estas fases pueden aplicarse a prácticamente cualquier concepto matemático relevante y en cualquier nivel educativo, desde la primaria hasta el doctorado.

VIII. Reflexión Final

Desde hace tiempo me resulta cada vez más claro que la enseñanza de la matemática ha sido atrapada por una lógica mecánica. Enseñamos fórmulas como si fueran recetas y luego preguntamos problemas donde se espera que el estudiante identifique el tipo de fórmula adecuada y la aplique. Pero en ese proceso, aparentemente pedagógico, hay una trampa, el estudiante no piensa, responde de forma automática. Memoriza un patrón, ejecuta un procedimiento, obtiene un número. Y el resultado, correcto o no, rara vez dice algo sobre su comprensión de lo matemático. Hemos convertido a los y las estudiantes en calculadoras.

La pregunta por el área del triángulo, como la muestro acá, es un ejemplo claro. Se da un dato que no tiene sentido, pero disfrazado con la familiaridad de los términos conocidos, "cateto", "altura", "área". El estudiante y profesionales, activan un reflejo condicionado, no una pregunta, y allí está el problema, el estudiante no se detiene a pensar si ese triángulo puede existir. Y todos nosotros y nosotras hemos caído en esa trampa

Yo creo que toda la matemática debería enseñarse como construcción, como computación. Y no hablo sólo de manipular objetos concretos, sino de construir con código. En el momento en que un estudiante programa una función para crear un triángulo con ciertas propiedades, incluso si ese código es generado de forma asistida con IA, está enfrentando el problema en su forma más pura y profunda, y no puede avanzar si no entiende. El código, o el diálogo con la IA para generarlo, refleja su pensamiento, y su intención constructivista.

Cuando pienso en cómo enseñar operaciones básicas sin recurrir a la memorización, es mucho más honesto plantearlas como un proceso de construcción. Dada una información inicial, que la construcción de tablas de multiplicar, por ejemplo, la haga el estudiante. La suma y la multiplicación no son datos que uno "recita", sino procesos para construir conjuntos y patrones. Desde un marco intuicionista, el o la estudiante sabe porque lo construyeron, no porque se les dictó.

Hoy, con la programación asistida, esa experiencia puede amplificarse, así, el aprendizaje deja de ser un ejercicio de repetición y se convierte en un descubrimiento, donde la memoria es un resultado colateral individual de cada estudiante, y no una meta. Ahora bien, tanto el aprendizaje como la evaluación deben ser procesos de construcción abierta mediados por inteligencia artificial y computación.

Cuando un estudiante intenta modelar una situación y el código falla, no es un error problemático, sino una posibilidad de descubrir y resolver. Si le pido que construya un triángulo rectángulo con un cateto de 10 y una altura sobre ese cateto de 6, y ninguna combinación de coordenadas funciona, llega a una conclusión que no necesita demostración formal, ese triángulo no puede existir. Al construir el triángulo con código, está haciendo computación, está haciendo matemáticas.

No se trata de convertir todas las clases en ejercicios de programación. Se trata de usar el código como una forma de hacer explícito lo que está oculto detrás de los símbolos. Las fórmulas pueden engañar, el código no. Esto obliga a pensar el proceso, y no sólo el resultado. Eso lo cambia todo.

La IA, en este contexto, no es el enemigo del pensamiento. Al contrario. Bien utilizada, puede ser una compañera que permita al estudiante probar hipótesis, explorar variantes, verificar construcciones. Pero siempre dentro de un marco donde el pensamiento no es delegado, sino estimulado.

En una educación así, no se pregunta por el área de un triángulo para que alguien repita la fórmula. Se plantea una situación, se invita a construir y a probar. El cálculo viene después, como consecuencia de haber entendido algo. No como su sustituto.

¡No más matemáticas mal enseñadas!

REFERENCIAS

[1] L. E. J. Brouwer, "Intuitionism and Formalism," Bulletin of the American Mathematical Society, vol. 20, no. 2, pp. 81–96, 1913.
[2] “Estudio de la UCR: Desastre en promoción de matemática y cálculo,” Semanario Universidad, 2005.
[3] “Alta reprobación obliga a UTN a reforzar temas de Matemáticas que alumnos vieron en el colegio,” La Nación, 2018.
[4] “Alumnos de primer ingreso a UCR tropiezan en prueba de Matemática: el 94 % reprobó,” La Nación, 2019.
[5] “Colegiales llegan a universidades sin conocimientos básicos,” La Nación, 2021.
[6] “Diagnóstico de UCR a estudiantes de primer ingreso confirma desastre en Matemática,” La Nación, 2022.
[7] “Puntuación 0: Estudiantes llegan a la UCR con conocimientos muy bajos en mate,” CRHoy, 2023.
[8] “Universidades públicas advierten que estudiantes llegan con serios vacíos académicos,” CR Hoy, 2024.
[9] “Brecha educativa en matemáticas compromete futuro académico y profesional,” Extra TV, 2025.
[10] R. R. Skemp, "Relational understanding and instrumental understanding," Mathematics Teaching, no. 77, pp. 20–26, Dec. 1976.
[11] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas. New York, NY, USA: Basic Books, 1980.
[12] P. Freire, Pedagogy of the Oppressed. New York, NY, USA: Continuum, 1970.
[13] E. A. C. Berube, “Rote learning in the age of technology: A quantitative study of the effectiveness of rote learning in spelling and basic math facts,” Ph.D. dissertation, Northcentral University, Prescott Valley, AZ, USA, 2011.
[14] V. I. Arnold, Problems for Children from 5 to 15. Oberwolfach: IMAGINARY gGmbH, 2004.
[15] H. von Foerster, Perception of the Future and the Future of Perception, in Instructional Science, vol. 1, no. 1, pp. 31–43, 1972.
[16] Claude, "Código interactivo de triángulo rectángulo," Claude.ai. [Online]. Available: https://claude.ai/public/artifacts/cdab1295-b375-4c6d-baef-099b29bed2a7. [Accessed: Aug. 15, 2025].
[17] R. Iemhoff, “Intuitionism in the Philosophy of Mathematics,” The Stanford Encyclopedia of Philosophy (Fall 2012 Edition), E. N. Zalta, Ed. [Online]. Disponible en: Stanford Encyclopedia of Philosophy entry on intuitionism.
[18] S. Dehaene, The Number Sense: How the Mind Creates Mathematics, Rev. and updated ed. Oxford, U.K.: Oxford Univ. Press, 2011.
[19] J. Sweller, P. Ayres, and S. Kalyuga, Cognitive Load Theory. New York, NY, USA: Springer, 2011.
[20] L. B. Resnick, “Developing mathematical knowledge,” American Psychologist, vol. 44, no. 2, pp. 162–169, 1989
[21] B. Jonsson, “Gaining mathematical understanding: The effects of creative mathematical reasoning and algorithmic reasoning instructional approaches,” PLoS One, vol. 15, no. 12, pp. 1–21, Dec. 2020.
[22] M. A. Boden, Purposive Explanation in Psychology. Cambridge, MA: Harvard University Press, 1972.
[23] T. de Camino Beck, “Interpretar con la máquina: Nuestra relación con la inteligencia artificial generativa,” CR Hoy, 26-Jul-2025.