

## **Introducción**

El mundo matemático va en una dirección acelerada en tanto a comunicación y tecnología se refiere. Como consecuencia, esto ha traído la necesidad de una nueva generación de profesionales que tengan conocimientos en los dos campos antes mencionados.

## Capítulo I: La enseñanza de las matemáticas

Entre los intereses de los alumnos de educación básica, no es un secreto que el menos importante es la disciplina matemática; con justa razón, lo que genera el sistema educativo a través de los docentes no es persuasión, más bien, miedo por parte del alumnado hacia las matemáticas. Y existen distintas problemáticas dentro de este sistema que, con el debido tiempo y dedicación a la materia, traen como consecuencia los resultados en las pruebas PISA, en donde en 2018, solamente un 44% del alumnado obtuvo un nivel mínimo de competencias en matemáticas [1]. Englobar cada problemática que causa este índice tan lamentable en las pruebas PISA nos llevaría más de unas cuantas cuartillas, por ello, sólo mencionaré un par de esas problemáticas que van bien ligadas a la enseñanza de una matemática que no se sostiene ni se ajusta a la velocidad con la que el mundo actual va avanzando gracias la tecnología.

La primera de ellas es que las matemáticas hoy en día, más que una necesidad, se están convirtiendo en una unificación entre múltiples áreas. El problema aquí es que esa unificación no se ve reflejada si no hasta que se tiene la oportunidad de llegar a un ambiente laboral en donde se requieran de esos conocimientos, y la pregunta que puede surgir ahora es ¿cuántos logran tener esa oportunidad y qué los condiciona a no llegar a ella?. Pues fijando nuestra atención a «La nueva familia de Libros de Texto Gratuitos» proporcionados por la SEP [2], podemos darnos cuenta que el intento de «unificación» no es más que el simple revoltijo de temas ya planteados en anteriores sexenios, ¿por qué enseñar primero los números naturales, pero en seguida un tema de evolución y tecnologías verdes?, si lo que se quiere evitar, como decía Morris Kline en su libro «El fracaso de la matemática moderna» [3] es una saturación de información y conceptos, considero que además de que esto contribuye a la antítesis propuesta por Kline, es aún peor mezclar dos cosas que son completamente distintas.

Otra de las dificultades que considero, es de alta prioridad atender, es la falta de visualización geométrica en la matemática; según un estudio realizado por la Universidad Técnica de Manabí (2019) [4], señala que:

*«La problemática en la enseñanza de la geometría plantea un desafío a todos los involucrados en el proceso de enseñanza y aprendizaje de las Matemáticas para hallar alternativas de solución, pues la enseñanza de la geometría se puede desvirtuar y se han dejado de lado procesos de razonamiento, argumentación y **visualización**, los cuales son trascendentales para el aprendizaje.»*

*p.(22).*

Lo cual puede traer como repercusión una falta de entendimiento respecto a las materias como cálculo integral y diferencial y geometría analítica, pues es en estas materias en donde la visualización de las funciones y figuras forma parte indispensable tanto de los cálculos como para el análisis dentro de la resolución de problemas.

En un primer ensayo argumentativo, destacué la importancia de no remover la teoría de conjuntos y la lógica para los alumnos de educación básica. Se planteó la necesidad de considerar estas ramas de estudio en ese nivel para una preparación encaminada al uso de Excel, que se ha convertido en un programa demandado por los empleadores y su funcionamiento está basado en funciones, conjuntos y lógica. Ahora, a manera de conectar la visualización geométrica con el uso de software, haré énfasis en los recursos tecnológicos con los que contamos hoy en día, los cuales nos pueden ayudar a no dejar de lado las problemáticas antes mencionadas.

## **Capítulo II: La Resolución de Problemas como metodología de enseñanza y el uso de tecnologías**

Cuando se publicaron los problemarios a resolver por el alumnado durante el seminario, llegamos a concluir que la manera de abordar uno de los ejercicios es distinta para cada persona y lo que influye más al momento de tomar un camino para la resolución, es el entendimiento del ejercicio en sí. Sin embargo, gracias a las heurísticas propuestas por Polya en su libro «Cómo plantear y resolver problemas» [5], se logró establecer una serie de consideraciones antes, durante y después de la solución. En particular los estudiantes se centraron en los cuatro pilares que se necesitan según Polya, para resolver un problema, los cuales fueron: comprender el problema, concebir un plan, ejecución del plan y examinar la solución obtenida. Posteriormente durante la exposición de algunas soluciones, se destacó la importancia de la visualización y el uso de las heurísticas para su conclusión. A continuación, durante este capítulo y el siguiente, veremos como puede esto relacionarse con el uso de las TICs en la resolución de problemas.

Hasta antes de la accesibilidad económica para tener un computador en casa, era necesario hacer todo en papel, con cuentas tediosas y visualizaciones de gráficas poco precisas. Hoy en día, además de que es más accesible esta tecnología, se cuenta con arquitecturas para uso remoto como:

- Google Colab: Es un servicio alojado de Jupyter Notebook que no requiere configuración y que ofrece acceso gratuito a recursos de computación, como tarjetas gráficas y unidades de procesamiento tensorial. Es una solución especialmente adecuada para el aprendizaje automático, la ciencia de datos y la educación.
- Programiz: Es un sitio web que cuenta con ejercicios y tutoriales para aprender a programar en Python, R, Java y otros lenguajes.

- Mathematica: software orientado a la computación de problemas en las áreas científicas, de ingeniería, matemáticas y computacionales. No solo es un sistema de álgebra computacional, sino también un potente lenguaje de programación de propósito general.

Se hace mención de estas herramientas para dar a conocer la capacidad de cómputo que posee cada una, pero también para familiarizar al lector con ellas, pues pueden ser mencionadas de ahora en adelante.

Consideremos que en el ensayo anterior, titulado «Metodología inmarcesible», se hizo incapie en el uso de Manim para visualizaciones de construcciones matemáticas, pero a fecha de publicación de este ensayo, no existen estudios acerca del apoyo que ha dado Manim a estudiantes de matemáticas, consecuentemente no es posible formar una argumentación sostenible sobre el uso de esta herramienta, sin embargo podemos plantear un análisis casi análogo.

Notemos que las TICs en educación son las Tecnologías de la Información y la Comunicación que se emplean como recursos y herramientas para el aprendizaje y que almacenan, procesan y transmiten información digital. Por poner un ejemplo, Grisales (2018) [6] argumenta que un caso particular que permite una evidencia de trascendencia luego de hacer uso del software Mathematica 10, permitió a los estudiantes universitarios de precálculo un aprendizaje más sencillo gracias a una serie de herramientas desarrolladas en el software. Además, se declara que el punto de partida de esa investigación fue el hecho de que el modelo de aprendizaje tradicional no ofrece una experiencia que genere una comprensión sobre los temas, pues no se permite una interacción con el objeto de conocimiento en cuestión.

La idea de que la tecnología como la conocemos hoy sea un puente para conectar objetos matemáticos con propiedades, haciendo uso de conceptos manipulables dentro de los programas en lugar de objetos abstractos, está bien sustentada en el artículo de Vega(2015) [7]. De donde

parte un argumento más general a la hora de proponer que los límites y las funciones junto con sus operaciones ayuden no solamente a visualizar, si no también a manipular los valores de estos y consecuentemente, intuir un cambio gráfico y analítico como se menciona en «Diseño de herramientas que fomentan el aprendizaje de matemáticas con ayuda de Mathematica 10» [8].

De este modo se puede ver que Manim, como lo define el autor, «es un motor de animación para videos **explicativos** de matemáticas. Se utiliza para crear animaciones precisas mediante programación en Python», es una TIC per se, con el apoyo de grandes Sponsors y una comunidad en línea [9]. Y pese a no tener estudios de ventajas y crecimiento intelectual así como un ahorro en el tiempo de aprendizaje, su uso sí se ve reflejado por importantes universidades como la Universidad de Alcalá: Escuela Politécnica Superior [10], y también por grandes divulgadores de las matemáticas, por mencionar algunos: Mates Mike, BlueDot, 3Blue1Brown, que sin duda al observar el crecimiento obtenido video a video, junto con los excelentes comentarios de los usuarios, es claro que nuestro argumento general del párrafo anterior, conserva su esencia al estar haciendo uso de esta paquetería de Python.

Una vez comprendido nuestro panorama y visión del futuro de la enseñanza de las matemáticas, estamos seguros para proponer una metodología que encamine a los estudiantes a usar estas herramientas no sólo para la resolución y exposición de problemas, también para un sencillo entendimiento ante las soluciones propuestas por otros compañeros.

### Capítulo III: Propuesta de trabajo de la Resolución de Problemas

Un pequeño ejemplo que aborda las dificultades mencionadas en el primer capítulo, está contenido en la materia de cálculo integral a nivel bachillerato. Primero se nos es presentada una motivación llevada por la necesidad de calcular áreas debajo de curvas, para esto se pide considerar alguna función conocida  $f(x)$ , posteriormente se nos invita a pensar en la “mejor” figura para aproximar el área debajo de  $f$  y después de algunas propuestas, llegamos a que el rectángulo es la mejor de ellas, haciendo que el área bajo  $f$  sea representada como

$$A \approx \sum_{i=1}^n f(x_i) \Delta x = \sum_{i=1}^n A_i$$

donde  $A_i$  es el área de cada rectángulo y  $n$  es el número total de rectángulos. Seguido de esto y con ayuda del cálculo diferencial, se nos invita a pensar en

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n A_i$$

Que, usando la notación de Leibniz eso es igual a

$$\int_a^b f(x) dx$$

Siendo esta la noción de comparar sumas finitas con sumas infinitas en términos de áreas bajo curvas.

De aquí podemos identificar una de las problemáticas y es la forma de graficar esta comparación, pues para simular una aproximación en clase, comúnmente se hace uso del pizarrón, dibujando rectángulo a rectángulo la cantidad que convenga, pero lo que sucede es que se pierde tiempo al momento de querer dibujar hasta 20 rectángulos pequeños (recordemos que la idea es llevarlos hasta infinitos rectángulos), más aún es complicada la práctica de manejar y/o comunicar iteraciones en funciones sin ayuda de software. Además de que se puede limitar a un solo ejemplo de este tema tan importante.

Y es aquí donde presento como propuesta didáctica el uso de Manim para visualizar las iteraciones hasta (aparentemente)  $n$  rectángulos. Veamos que el código puede ser proporcionado por ChatGPT ejecutando un prompt como:

Genera una animación en manim que pueda ser usada como ejemplo de visualización para el entendimiento de sumas finitas y sumas infinitas respecto a áreas bajo la curva, es decir, hablamos de una función que en un principio, su área bajo la curva se pueda aproximar mediante rectángulos y después iterar la cantidad hasta que sean, aparentemente  $n$  rectángulos.

Y obtenemos un código extremadamente óptimo que bien puede ser usado para exponer:

```

1  from manim import *
2
3  class SumasFinitasEInfinitas(Scene):
4      def construct(self):
5          # Ejes y función
6          axes = Axes(x_range=[0, PI],
7                      y_range=[0, 1, 0.2],
8                      axis_config={"color": BLUE})
9          func = lambda x: np.sin(x)
10         graph = axes.plot(func, color=WHITE)
11
12         # Título
13         title = Text("Sumas Finitas e Infinitas", font_size=36).to_edge(UP)
14         self.play(Write(title))
15
16         # Mostrar los ejes y la gráfica
17         self.play(Create(axes), Create(graph))
18         self.wait(1)
19
20         # Aproximación con rectángulos
21         rects_group = VGroup()
22         for n in [4, 8, 16, 32]:

```




```

23         rects = axes.get_riemann_rectangles(graph,
24                                             x_range=[0, PI],
25                                             dx=PI/n,
26                                             stroke_width=0.5,
27                                             stroke_color=WHITE,
28                                             fill_opacity=0.5)
29         rects_group.add(rects)
30
31         # Mostrar las diferentes aproximaciones
32         for i, rects in enumerate(rects_group):
33             if i == 0:
34                 self.play(Create(rects))
35             else:
36                 self.play(Transform(rects_group[i-1], rects))
37             self.wait(1)
38
39         # Transformación final a una "suma infinita"
40         final_rects = axes.get_riemann_rectangles(graph,
41                                                  x_range=[0, PI],
42                                                  dx=PI/1000,
43                                                  stroke_width=0.5,
44                                                  stroke_color=WHITE,
45                                                  fill_opacity=0.5)
46         self.play(Transform(rects_group[-1], final_rects))
47         self.wait(2)
48
49         # Fin de la animación
50         self.play(FadeOut(rects_group),
51                 FadeOut(graph),
52                 FadeOut(axes),
53                 FadeOut(title))
54         self.wait(1)

```

Y obtenemos como output un video del resultado óptimo para la visualización que se necesitaba, el cual puede ser consultado en [youtube.com/output](https://youtube.com/output). O bien si no se desea depender tanto de la IA podemos construir nuestro propio código:

```
1 from manim import *
```

 Python

```

2
3 class Area(Scene):
4     def construct(self):
5         # Configuración de los ejes
6         ax = Axes(
7             x_range=[0, 5, 1],
8             y_range=[0, 6, 1],
9             axis_config={"include_numbers": True},
10        )
11
12        # Función a graficar
13        def func(x):
14            return 1 + x ** 2 * np.exp(-0.15 * x ** 2)
15
16        graph = ax.plot(func, color=BLUE)
17
18        # Texto de introducción
19        int_area_sym = MathTex(r"\int_{a}^b f(x)dx").shift(2 * UP)
20        area_mean_text = MathTex("""
21            \\text{Aproximaremos el área bajo la curva de}
22            \\hspace{.3cm} f(x) \\hspace{.3cm}
23            \\text{en la región} \\hspace{.3cm}
24            1 \\leq x \\leq 4""").scale(0.6).next_to(int_area_sym,
25            DOWN)
26
27        opening_text = VGroup(int_area_sym, area_mean_text)
28        self.play(Write(opening_text), run_time=4)
29        self.wait(2)
30        self.play(FadeOut(opening_text))
31
32        # Añadir ejes y graficar la función
33        self.play(Create(ax), Create(graph))
34        func_text = MathTex(r"y = f(x)").next_to(graph, UP)
35        self.play(Write(func_text))
36        self.wait(2)
37
38        # Límites verticales y etiquetas
39        min_lim = ax.get_vertical_line(ax.input_to_graph_point(1, graph),
40            color=YELLOW)

```

```

41     max_lim = ax.get_vertical_line(ax.input_to_graph_point(4, graph),
42                                   color=YELLOW)
43     self.play(Create(min_lim), Create(max_lim), run_time=0.5)
44     self.wait(2)
45
46     # Aproximación del área con rectángulos de Riemann
47     approx_text = Text("""El área puede ser aproximada con \n
48                        sumas de rectángulos pequeños""").scale(0.5).next_to(graph,
49                                                                              4 * UP)
50     self.play(Write(approx_text))
51     self.wait(2)
52
53     rects_initial = ax.get_riemann_rectangles(graph,
54                                                x_range=[1, 4],
55                                                dx=0.5,
56                                                stroke_width=0.1)
57     self.play(Create(rects_initial), run_time=2)
58     self.wait(2)
59     self.play(FadeOut(rects_initial), run_time=1)
60
61     rects_initial = ax.get_riemann_rectangles(graph,
62                                                x_range=[1, 4],
63                                                dx=0.1,
64                                                stroke_width=0.1)
65     self.play(Create(rects_initial), run_time=2)
66     self.wait(2)
67     self.play(FadeOut(rects_initial), run_time=1)
68
69     rects_initial = ax.get_riemann_rectangles(graph, x_range=[1, 4],
70                                                dx=0.01,
71                                                stroke_width=0.01)
72     self.play(Create(rects_initial), run_time=2)
73     self.wait(2)
74
75     conclude_text = Text("""Haciendo esto
76                           infinitamente, tenemos""").scale(0.5).next_to(graph, 4 * UP)
77     self.play(Transform(approx_text, conclude_text))
78     self.wait(3)
79

```

```
80     int_area_sym.next_to(graph, IN)
81     self.play(Create(int_area_sym), run_time=1)
82     self.wait(2)
83
84     # Haciendo los rectángulos más finos
85     rects_finer = ax.get_riemann_rectangles(graph,
86                                             x_range=[1, 4],
87                                             dx=0.001,
88                                             stroke_width=0.001)
89     self.play(Transform(rects_initial, rects_finer), run_time=2)
90
91     self.wait(3)
```

En donde solamente hay una diferencia de 30 líneas, líneas que bien fueron agregadas por pulir detalles, pero el resultado es todavía mejor para comunicarlo ([youtube.com/output2](https://youtube.com/output2)).

## Referencias

- [1] J. A. C. Gil, «Situación de la educación en México». Accedido: 27 de mayo de 2024. [En línea]. Disponible en: <https://www.iberofam.org/blog/situacion-de-la-educacion-en-mexico/>
- [2] Secretaría de Educación Pública, «Conoce tus libros». [En línea]. Disponible en: <https://conocetuslibros.sep.gob.mx/conoce>
- [3] M. Kline, *El fracaso de la matemática moderna, Por qué Juanito no sabe sumar*, 18a ed. siglo veintiuno editores, 1998.
- [4] R. C. M. Carlos Alberto Aray Andrade Orlando Francisco Párraga Quijano, «La falta de enseñanza de la geometría en el nivel medio y su repercusión en el nivel universitario: análisis del proceso de nivelación de la Universidad Técnica de Manabí.», *Revista de Ciencias Humanísticas y Sociales*, vol. 4, n.º 1, pp. 20-31, 2019.
- [5] G. Polya, *Cómo plantear y resolver problemas*, 15a ed. Editorial Trillas, 1989.
- [6] A. M. Grisales-Aguirre, «Uso de recursos TIC en la enseñanza de las matemáticas: retos y perspectivas.», *Entramado*, vol. 14, n.º 2, pp. 198-214, 2018, doi: 10.18041/1900-3803/entramado.2.4751.
- [7] J. C. V. Vega, «Enseñanza de las matemáticas básicas en un entorno e-Learning: un estudio de caso de la Universidad Manuela Beltrán Virtual.», *Revista Escuela de Administración de Negocios*, vol. 79, pp. 172-185, 2015, [En línea]. Disponible en: [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0120-81602015000200011](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-81602015000200011)
- [8] C. A. Ramírez, «Diseño de herramientas que fomentan el aprendizaje de matemáticas con ayuda de Mathematica 10.», *Revista Elementos*, vol. 5, pp. 65-78, 2015, [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=5179413>
- [9] Manim Community, «Manim Community». [En línea]. Disponible en: <https://www.manim.community/>

- [10] Á. Mesón, «Animaciones programáticas con Manim como soporte a la docencia universitaria», 2023. [En línea]. Disponible en: <https://ebuah.uah.es/dspace/handle/10017/55831>