# LA MATEMATICA DISCRETA COMO FORMACION BASICA

Sylvia da Rosa
Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
email: darosa@fing.edu.uy

**Resumen**
Desde las primeras calculadoras hasta los computadores modernos, no hay duda que se ha perseguido y se viene alcanzando el objetivo de realizar cálculos rápidos y seguros. El desarrollo de las interfaces también ha contribuído a obtener cada vez mejores representaciones gráficas y se puede preveer que la tecnología aplicada tanto a los problemas de cálculo como a lo visual brindará cada vez herramientas más potentes para asistir en la solución de problemas en diferentes áreas. Por otro lado, sabemos que los problemas que dependen del razonamiento humano, nunca podrán prescindir del mismo. En pocas palabras: las computadoras computan (calculan), pero nunca van a llegar a razonar (Hamilton,1978). Creemos que esta realidad, incide directa y profundamente sobre el sistema educativo, cambia radicalmente objetivos, contenidos y metodologías de lo que enseñamos, cómo lo enseñamos, para qué enseñamos, siendo imprescindible poner el acento en enseñar a razonar, abstraer, analizar, representar, lo cual, a nuestro juicio ha sido descuidado por el enorme peso que la enseñanza pone en desarrollar destrezas en el cálculo y en la demostración mecánica de teoremas.

**La actividad de computar**
La actividad de computar es una disciplina muy antigua, cuyos orígenes se remontan a civilizaciones como la griega, la babilónica y la egipcia.Los antiguos filósofos y matemáticos griegos contribuyeron enormemente en la sistematización del razonamiento y en la construcción de algoritmos, mientras que los egipcios y los babilónicos desarrollaron métodos computacionales destinados a facilitar el trabajo humano.
En todas las épocas han existido fuertes motivaciones para conseguir resultados avanzados tanto en la sistematización del razonamiento como en el diseño y construcción de dispositivos para realizar computaciones seguras y eficientes.
Leibniz (1646-1716) escribió: “Es lamentable que personas de excelencia deban desperdiciar horas como esclavos en una labor de calcular, que podría confiarse a otras personas si fueran utilizadas máquinas.[1]” (Tucker et al.,1995). La propuesta de David Hilbert (1862-1943) de encontrar un sistema axiomático lógico-matemático, del cual toda la matemática pudiera ser derivada, fue probada como imposible de realizar por Kurt Gödel (1906-1978), en 1931, estableciendo que hay problemas matemáticos que son inherentemente insolubles y revolucionando el punto de vista de los matemáticos sobre su disciplina. El trabajo de Gödel tuvo repercusiones prácticas inmediatas, planteando la cuestión de qué significa exactamente decir que se tiene un método para resolver un problema. Varias respuestas fueron propuestas, dos de las cuales tuvieron enorme impacto en el desarrollo posterior de los computadores digitales y de los lenguajes de programación:

[1] Traducción de la autora.

la de Alan Turing (1912-1954), que muestra que toda computación efectiva puede ser representada en una máquina abstracta, conocida como la máquina de Turing y la de A. Church conocida como la tesis de Church (1935), que expresa que un objeto es computable si se puede definir en el Cálculo Lambda (Hamilton, 1978; Barendregt, 1984). El desarrollo tecnológico ha actuado en cada momento histórico como freno o impulsor de las expectativas científicas . En las últimas décadas ese desarrollo ha influido enormemente en la expansión y profundización del estudio de problemas y teorías de antiguo origen, dando lugar a nuevas ciencias, enfoques y metodologías. Un ejemplo de ello lo proporciona la disciplina de computación, que se ha extendido y profundizado en diferentes direcciones y bajo diferentes nombres: Ciencia de la Computación (CC en adelante), Ingeniería en Computación, Informática, etc, cuya base común es, como siempre lo ha sido, el hecho de que se trata de una actividad matemática que ha tomado también forma y nombre como la rama de la matemática llamada Matemática Discreta (MD en adelante). Como su nombre lo indica, en MD se trabaja con conjuntos discretos, a diferencia de la Matemática Continua que trabaja con conjuntos continuos, como los números reales. El computador digital moderno es básicamente un sistema discreto y muchas de sus propiedades pueden ser entendidas y descritas modelándolas en un sistema matemático discreto. La MD comprende la lógica (o más correctamente, las lógicas): los computadores modernos no sólo nos liberan del "trabajo de esclavo" al que hacía referencia Leibniz, sino que juegan un papel activo en el desarrollo de pruebas, abriendo nuevos caminos y enfoques en el quehacer matemático (Giménez & Paulin-Mohring,1996).

**La Matemática Discreta y la Enseñanza Media**

En lo que respecta al sistema educativo, es imprescindible que éste se adapte a los cambios generados por los avances teóricos y las aplicaciones en los diferentes campos científicos y tecnológicos, garantizando la transmisión y supervivencia del conocimiento. Así como en su época el cálculo diferencial se vió impulsado por el desarrollo de la física y los problemas planteados por esta ciencia en el siglo XVIII, determinando fuertemente la orientación de la educación matemática , los cambios ocurridos en el siglo que acaba de terminar reclaman del sistema educativo otras orientaciones. La MD debe su intenso desarrollo de los últimos años a la comunidad científica relacionada con la CC y en lo que se refiere a la educación, los estudios terciarios en dicha ciencia han incorporado cursos de MD con alta prioridad. Sin embargo, fuera del área de la CC, la MD es prácticamente inexistente y esta situación es la que creemos que debe corregirse, ya que consideramos que los estudios en MD son importantes para la formación de cualquier estudiante, *aún de aquellos que no continúen estudios terciarios.* Hoy día, la Economía, las Ciencias Sociales, las Ciencias gerenciales, la Ingeniería eléctrica, la Física simbólica, por solo nombrar algunas, tienen necesidad de resolver problemas que se modelan utilizando herramientas de MD. En el mercado laboral, es cada vez más frecuente que las personas deban enfrentarse a situaciones que involucran toma de decisiones, procesos de abstracción y razonamientos lógicos, necesitando habilidades que raramente son desarrolladas en su pasaje por la Enseñanza Media. Más aún, consideramos anacrónico que no exista en la cultura general brindada por la Enseñanza Secundaria una base de conocimientos sobre algo tan ampliamente difundido en la sociedad actual como es la informática. Nombres como Turing, Church, Gödel, Dijkstra, y conceptos como tipo abstracto de datos, variable ligada, cuantificador, vinculados a la MD, son desconocidos para el ciudadano común e inexistentes en la Enseñanza Media. El cambio que proponemos implica un proceso de adaptación, incorporación y modificación que muchas veces se realizará en etapas de aproximaciones sucesivas. En este trabajo presentamos una propuesta para comenzar a actualizar la enseñanza de matemática a nivel de la Enseñanza Media. En pocas palabras, consiste en tomar de los programas actuales los temas de MD, como ser Teoría de

Conjuntos, Relaciones, Funciones, Combinatoria, Inducción Completa, Divisibilidad, e introducirlos con un enfoque alternativo que rescate la naturaleza discreta de los mismos y al mismo tiempo permita dedicarles mayor tiempo y profundidad, relacionándolos entre ellos. Al menos en nuestro país, constatamos que los temas mencionados como temas de MD, son subvalorados en la enseñanza: se dan en poco tiempo y a las apuradas, para poder dedicar prácticamente todo el tiempo de los cursos al estudio del conjunto de los números reales y al cálculo. Como consecuencia de esto, hemos constatado a través de tests realizados a estudiantes del curso de MD del primer año de la carrera de Ingeniería en Computación, que los estudiantes desconocen los conjuntos y sus propiedades (todo se reduce al conjunto de los reales) y aplican incorrectamente los métodos de prueba más elementales por falta de una sólida base en lógica.
Nuestro enfoque plantea asimismo una metodología para encarar el proceso de resolución de un problema dado, incorporando el método formal de una manera que puede hacerse desde el nivel secundario. Podemos resumirla en los siguientes pasos:

- Crear un algoritmo (método) para resolver el problema, expresándolo en algún lenguaje informal (por ejemplo, en español)
- Expresar el algoritmo matemáticamente
- Construir un programa que implemente el algoritmo definido matemáticamente
- Verificar la correctitud del algoritmo (o del programa)

El tercer punto planteado puede obviarse, sin que el enfoque sea alterado, en caso de que no sea posible incorporar en el curso un lenguaje de programación. En la siguiente sección ilustramos el proceso a través de un ejemplo.

**Las funciones como métodos**

El concepto de algoritmo está estrechamente vinculado a la resolución de problemas matemáticos desde sus orígenes: el propio nombre "algoritmo" proviene del nombre del matemático persa Abu Ja´far Mohammed ibn Mûsa al-Khowârizmî, que vivió aproximadamente en el año 825 A.C.. Por otro lado, el concepto de función es también un concepto matemático fundamental y que tiene aplicaciones en casi todas las ciencias y disciplinas. Ambos conceptos, el de algoritmo y el de función, raramente se presentan relacionados, a pesar de que lo están estrechamente. Tradicionalmente, se introduce el concepto de función como una relación que cumple determinada propiedad, es decir como un conjunto de pares. De esta forma, no es intuitivo ver el nexo existente entre función y algoritmo, ni tampoco formalizar la aplicación. En cambio, si consideramos a las funciones como métodos (Barendregt, 1984), es fácil ver que una función es un objeto matemático que representa un algoritmo, a la vez que aplicar una función es aplicar un método para obtener un resultado. Como ejemplo ilustrativo, presentamos brevemente el tratamiento del tema sobre el algoritmo de Euclides para hallar el máximo común divisor de dos naturales. Como partimos de un método ya creado, el primer punto consiste en especificar el método para resolver el problema en un lenguaje informal: Dados a y b $\in$ N, b > 0, para obtener el mcd de a y b, seguimos el siguiente método (algoritmo):

Dividimos a entre b obteniendo un resto r.
Si r es 0, devolvemos b,
Si no, volvemos a aplicar el método con b y r.

Observar que las variables a, b y r usadas en el algoritmo, designan los distintos valores con los cuales aplicamos el método reiteradamente hasta que obtenemos un valor de r igual a 0. Se puede observar también, que este enfoque facilita la introducción del concepto de

recursión y su comprensión, a partir de que el algoritmo especificado en lenguaje natural es recursivo, es decir, se consigue el resultado mediante aplicaciones del mismo a valores que de alguna manera se van reduciendo a un caso final (caso base).
Si llamamos mcd al algoritmo, podemos definir la siguiente función matemática que lo expresa (a mod b es el resto de la división entera entre a y b):

$$mcd : N \times (N - \{0\}) \rightarrow N$$

$$mcd\,(a,b) = \begin{cases} b, & \text{si (a mod b) es igual a 0} \\ mcd\,(b, (a \bmod b)), & \text{si no} \end{cases}$$

Si se cuenta con la posibilidad de utilizar un lenguaje de programación en el curso, el proceso continúa con la implementación de la función definida. Mostramos como hacerlo en Isetl (Dubinsky & Fenton, 1996):

```
mcd := func(x,y);
              if is_nat (x) and is_nat (y) and y > 0
              then
               r: = x mod y;
               if r = 0 then return y;
               else return mcd (y,r); end;
             end;
end;
```

Si no es posible o no se considera adecuado introducir la construcción de un programa, puede pasarse a la etapa siguiente que consiste en reponder la siguiente pregunta:
Cómo sabemos que efectivamente el algoritmo computa el máximo común divisor de dos naturales dados ? Es decir, lo que se plantea es demostrar que el algoritmo es correcto.
Usando la función *divisores* que toma un natural y devuelve el conjunto de sus divisores y la función *max* que toma un conjunto de naturales y devuelve el mayor de ellos, puede definirse el máximo común divisor de a y b como max (divisores(a) $\cap$ divisores(b)).
Para probar que el algoritmo es correcto, se establece la siguiente propiedad que expresa que la función mcd, definida para representar el algoritmo, debe satisfacer la definición:

$$\forall a,b \in N,\ b > 0,\ max\,(divisores(a) \cap divisores(b)) = mcd\,(a,b)$$

La posibilidad de olvidarnos de la necesidad de calcular, nos permite profundizar en conceptos esenciales y las relaciones entre ellos: la *especificación* de la solución del problema dado, la *correctitud de un algoritmo* que lo soluciona y la expresión matemática del algoritmo como *función*, la *propiedad* que establece el vínculo entre especificación y algoritmo y la *prueba* de la propiedad.
Todos estos conceptos aparecen continuamente en el estudio de la matemática, en cualquier nivel, los estudiantes los manejan separada e informalmente.
Nuestro enfoque presenta una metodología que les da coherencia y perspectiva, permitiendo el desarrollo de la capacidad de abstracción y el pensamiento algorítmico.
Queda claro que lo importante no es la realización de los cálculos (que puede hacerlos la computadora), sino comprender el proceso de abstracción por el cual representamos el método por una función matemática. Se puede estudiar las propiedades de dicha función, por ejemplo, la inyectividad y la sobreyectividad. De aquí surgen interrogantes interesantes, como ser: podemos representar cualquier método por una función matemática ?, toda función matemática representa un algoritmo ?, qué significa que los algoritmos (o las funciones que los representan) sean iguales o equivalentes ? Por supuesto, el tratamiento formal de estas cuestiones, escapa al marco de la educación media, sin

embargo, pueden introducirse informalmente como problemas relacionados que se plantean. También pueden presentarse problemas de aplicación relacionados con la CC, de la misma manera que muchas veces se plantean aplicaciones de la matemática a la física u otras ciencias, por ejemplo, las interrogantes mencionadas tienen que ver con el hecho de que hay funciones que no son computables, o algoritmos que si bien son computables, insumen tantos recursos que son inviables en la práctica.
Como ejemplos de aplicación, se puede ver que un programa es la implementación de una función parcial, que el dominio son los datos de entrada, ejecutar el programa es aplicar la función y los elementos del codominio son los datos de salida. La etapa de construcción del programa, si bien no es estrictamente necesaria en una primera aproximación, tiene muchas ventajas desde el punto de vista educativo. Entre ellas, un lenguaje de programación obliga a ser riguroso en las definiciones, lo cual muchas veces permite clarificar conceptos, el lenguaje de programación permite experimentar los algoritmos con mayor número de casos o casos más grande o "raros", ayuda a detectar errores y a corregirlos y además, algo nada despreciable ... hace la clase mucho más divertida !

**Resultados obtenidos**

Las ideas expuestas en este trabajo fueron estructuradas en la forma de un curso dictado para docentes de matemática del ciclo superior de Enseñanza Secundaria de Montevideo, Uruguay. Este ciclo comprende los dos últimos años, luego de los cuales el estudiante está habilitado para ingresar en la Universidad. El curso estuvo basado en el libro "Discrete Mathematics with Isetl" de Ed Dubinsky y William Fenton y se utilizó el lenguaje de programación Isetl. Como evaluación, los integrantes debieron realizar un trabajo sobre cómo exponer un tema de su elección con el nuevo enfoque y utilizando el lenguaje Isetl. En particular, dos grupos de docentes realizaron la experiencia con sus alumnos en clase, uno de ellos introduciendo el tema “Inducción” y el otro el tema “Divisibilidad”. Si bien este hecho no brinda datos suficientemente amplios como para extraer conclusiones generales, los resultados obtenidos son alentadores: se constató en la práctica la falta de validez de prejuicios tales como "los estudiantes de secundaria no pueden programar", "programar es difícil", etc, y las ventajas mencionadas arriba sobre la inclusión de un lenguaje de programación fueron empíricamente comprobadas. De los resultados observados, se desprende claramente, por ejemplo, que la posibilidad de programar permite a los estudiantes construir la definición de una función , a partir de abstraer de casos particulares, mejorando enormemente la comprensión del concepto de función. Para lograr este tipo de resultados, los docentes llevaron a cabo una ardua labor de elaboración de problemas, ejercicios y preguntas, de modo de adecuar la presentación y desarrollo de los temas a la metodología propuesta.

**Conclusiones**

Esta experiencia ha reafirmado nuestro convencimiento en dos sentidos que sintetizamos en las siguientes conclusiones: 1) la actualización de contenidos, objetivos y metodologías en la enseñanza de Matemática de nivel pre-universitario, incorporando estudios en Matemática Discreta, no solamente es una necesidad imperiosa sino que es una tarea *posible* de llevar a cabo en las condiciones actuales. La actualización de la educación matemática en el sentido que proponemos redundará en un beneficio para todos los estudiantes y la formación de los estudiantes que seguirán estudios terciarios en CC se verá enormemente favorecida. Este hecho merece ser tenido muy en cuenta, fundamentalmente por dos razones: en primer lugar, en nuestro país las carreras en CC son las que más han aumentado el número de estudiantes en los últimos 20 años. Solamente en la Faculatd de Ingeniería de la Universidad de la República, que imparte 9 títulos de grado, la mitad de los estudiantes ingresa a Ingeniería en Computación donde *recién comienzan* a recibir la base adecuada en su formación matemática. En segundo lugar, proporcionar una formación mejor a los informáticos, desde el punto de vista de la utilización de métodos formales en la construcción de software, es un requerimiento cada vez más exigido por el mercado laboral y la sociedad (da Rosa & Cirigliano, 1998).

2) el modelo seguido en nuestra experiencia es el más adecuado, a saber: la etapa de formación de los docentes de la Enseñanza Media, debe realizarse en *estrecha colaboración* con docentes de Ciencia de la Computación de la Universidad. Este punto, que consideramos condición ineludible para el éxito de la tarea, brinda además una posibilidad concreta de fomentar el intercambio entre los docentes de los dos ámbitos del sistema educativo, necesidad muchas veces sentida por los docentes y pocas veces satisfecha.

**Referencias Bibliográficas:**


Allen Tucker et al. (1995). *Fundamentals of Computing I. Logic, Problem Solving, Programs and Computers.* McGraw-Hill Series in Computer Science, New York.

Ed Dubinsky & William Fenton. (1996). *Introduction to Discrete Mathematics with Isetl.* Springer-Verlag, New York.

A.G. Hamilton. (1978). *Logic for Mathematicians.* Cambridge, USA: Cambridge University Press.

H.P.Barendregt. (1984). *The Lambda Calculus. Its syntax and semantics*. Studies in Logic and the Foundations of Mathematics. Vol 103. Amsterdam, North-Holland.

Eduardo Giménez & Christine Paulin-Mohring, (1996). *Types for Proofs and Programs.* LNCS nr. 1512. Types´96. Springer-Verlag, Berlin.

Sylvia da Rosa & Gustavo Cirigliano. (1998). Matemática y Programación. Anales del VI Congreso Iberoamericano de Educación Superior en Computación. CIESC'98. Pontificia Universidad Católica del Ecuador, Quito, Ecuador.