# Hitos matemáticos en la historia de la computación

**Marco Flores Nuñez**

19 de marzo de 2013

## 1. Álgebra booleana

El álgebra booleana fue introducida en 1854 por el matemático inglés George Boole en su libro *An Investigation of the Laws of Thought*.

George Boole falleció en 1864, pero sus ideas han vivido y evolucionado mucho más allá de lo que su creador pudo prever. Para aclarar, a pesar del nombre, el álgebra booleana como la conocemos hoy no es un aporte exclusivo del fallecido inglés, muchos académicos han contribuido a su ya mencionada evolución.

En la década de 1930, por ejemplo, Claude Shannon (cabe mencionar, entre los logros del señor Shannon, ser conocido como ."El padre de la teoría de la información") observó que se podían aplicar las ideas de Boole en un contexto de conmutación de circuitos, utilizando el concepto de puertas lógicas e introdujo el "álgebra de conmutaciones". Hoy en día la necesidad de considerar distintos tipos de álgebras booleanas es mínima, por lo tanto "álgebra de conmutaciones"z "álgebra booleana"se usan refiriéndose a la misma cosa. Todo esto está detallado en la tesis de maestría de Shannon, que William Poundstone, en su libro *Fortune's Formula: The Untold Story of the Scientific Betting System That Beat the Casinos and Wall Street* aclama como la tesis de maestría más importante de todos los tiempos.

## 2. Máquina de Turing

Una computadora es: ¿un conjunto de circuitos y cables conectados a un monitor? ¿un aparato capaz de llevar a cabo instrucciones formuladas por un programador? Claramente no es fácil definir el concepto. Alan Turing, quien vivió hasta el año 1954, nunca conoció una çomputadoraçomo nosotros la conocemos, pero Alan Turing, matemático británico, fue uno de los primeros en idear una máquina capaz de realizar computaciones (incluso si dicha máquina fuese totalmente hipotética).

Según Wikipedia, una máquina de Turing (no confundir con la banda de rock instrumental "Turing Machine") es un artefacto hipotético capaz de manipular símbolos inscritos en una tira de siguiendo un conjunto de reglas.

Citando al señor Turing en su ensayo de 1948 titulado *Intelligent Machinery*, en el cual se refería a la máquina que publicó en 1936:

«...una capacidad de memoria ilimitada conseguida usando una cinta infinita marcada en cuadrículas, en cada una de las cuales se imprime un símbolo. En cualquier punto del tiempo hay un símbolo en la máquina; se le llama el símbolo escaneado. La máquina puede alterar al símbolo escaneado y su comportamiento es parcialmente determinado por ese símbolo, pero los símbolos en otros lugares de la cinta no afectan al comportamiento de la máquina. Sin embargo, la cinta puede moverse hacia adelante o atrás en la máquina, esta es una de las operaciones básicas de la máquina. Cualquier símbolo en la cinta puede, entonces, ser escaneado.»

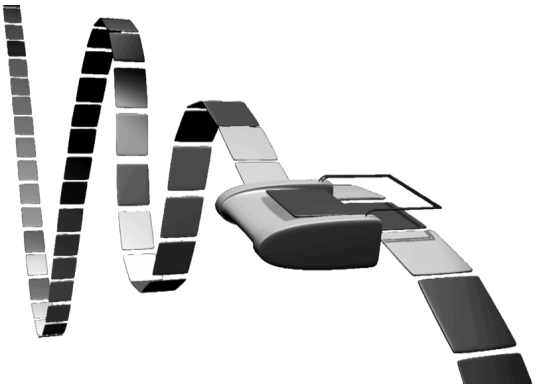

Figura 1: Máquina de Turing (no la banda).

## 3. Algoritmo de Dijkstra

Es difícil conversar con alguien interesado en algoritmos que no conozca el nombre de Edsgar Dijkstra (aunque es fácil encontrar a personas que no sepan su nombre de pila), después de todo él ideó uno de los algoritmos más conocidos en la ciencia de la computación.

Edsgar Dijkstra acuñó este conocido algoritmo en 1959, aunque la idea se le ocurrió en 1956. Ahondar en las vicisitudes técnicas del algoritmo estaría fuera del tono de este artículo así que basta con decir que solucionó el problema de encontrar el camino más corto entre dos nodos de un grafo tal que el peso de las aristas que los conectan es mínimo (en $O(V^2)$, para los interesados).

Las contribuciones de Dijkstra al campo son mucho mayores a la postulación de este algoritmo, entre ellas se incluyen avances en teoría de paralelimos y la sustentación de argumentos en contra del uso de la sentencia *GOTO* (que contribuyera a la proliferación del paradigma de lenguajes de programación estructurados, nuevamente, para los interesados). Tanto así que el premio ACM PODC de Artículo Influencial fue renombrado en su nombre en el año 2003, un año después de su muerte.

## 4. Hito futuro: $P = NP$

Una vez más, cae fuera del ámbito de este trabajo explicar los detalles del análisis de algoritmos, pero un resumen apropiado iría más o menos así: los problemas $P$ son aquellos que tienen una mejor solución que corre en tiempo polinomial, mientras aquellos que son $NP$ tienen una mejor solución de mayor complejidad. El problema de $P = NP$ busca encontrar una demostración formal de que cualquier problema $NP$ puede ser reducido a un problema $P$.

En agosto del 2010, Vinay Deolalikar demostró que $P \neq NP$, dejando en claro de una vez por todas que los problemas $NP$ no podían ser reducidos a una complejidad polinomial. Sin embargo muchos académicos han salido a retar su publicación, señalando errores fatales que probarían que su prueba es errónea (más información en la página relevante de PolyMath Wiki).

Por lo tanto, este problema queda abierto para el futuro y, si se llega a demostrar que $P = NP$ un cambio radical podría suscitarse en la ciencia de la computación y la disciplina del diseño de algoritmos.