

# A debate on teaching Computing Science

Edsger W. Dijkstra

Communications of the ACM, Diciembre 1989

Edsger Dijkstra era matemático, físico y doctor en Computación. Nacido en Rotterdam, Holanda en 1930 y fallecido en 2002, es muy reconocido por sus numerosas contribuciones al área de Computación. Entre éstas se cuentan el algoritmo de Dijkstra para caminos mínimos en grafos y los semáforos para control de exclusión mutua. Por sus numerosas contribuciones en lenguajes de programación recibió el premio Turing en 1972. Fue uno de los pioneros en investigación en sistemas distribuidos, y el premio anual de la ACM para esta área lleva su nombre.

La publicación “A debate on teaching Computing Science” apareció en la publicación de la ACM en Diciembre de 1989 después de que meses antes Dijkstra diera una charla titulada “On the cruelty of really teaching Computing Science” en la conferencia de la organización. A continuación se resume el contenido de la publicación, que se divide en dos partes esenciales.

## 1. Novedades radicales

Cuando aprendemos algo en general tendemos a pensarlo en términos de algo que ya conocíamos anteriormente. Este mecanismo es apropiado cuando los cambios son graduales; cuando los nuevos conocimientos no se alejan demasiado de los anteriores.

Dijkstra postula que la computación es una **novedad radical**: se trata de un área que por sus características particulares está muy alejada de todas las que se conocían hasta el momento. Por lo tanto, el método de razonamiento por analogía no funciona, ya que las diferencias entre la computación y todo lo conocido anteriormente son demasiado grandes. Las diferencias que hacen a esta *radicalidad* son dos:

- **Las escalas involucradas**: La mayoría de las disciplinas atacan un rango pequeño de órdenes de magnitud. Ya sea que se trate de problemas de tamaño pequeño, mediano o grande, la mayoría de los especialistas solo trabajan en una sola de estas categorías. La computación, por otra parte, requiere que una única persona trate con jerarquías muy profundas, desde un diseño de muy alto nivel hasta las particularidades de *bits* y *bytes*.
- **La “digitalidad”**: La computadora es el primer dispositivo digital a gran escala en un mundo predominantemente analógico. En el mundo digital pequeños cambios no siempre tienen pequeñas consecuencias, y esto contradice la intuición convencional de la época. Si bien hoy por hoy la persona promedio interactúa regularmente con dispositivos digitales, hay que tener en cuenta que cuando se escribió el *paper* esto no era el caso.

## 2. Consecuencias científicas y educativas

Según Dijkstra, computar es manipular símbolos. En correspondencia con ésta visión, un programa es una fórmula que determina como debe realizarse dicha manipulación. Así, la computación puede verse como la **aplicación a gran escala de la lógica**.

Esta visión es un poco particular. El autor argumenta que las personas están muy mal predisuestas a adaptarse a las novedades radicales, ya que exigen un cambio de mentalidad que puede ser trabajoso. Por esta razón, insisten en sus propias visiones personales sobre lo que creen que es o debe ser la computación. Esto redundaba en que los contenidos enseñados en las universidades se ajustan a estas visiones, degradando la calidad educativa. Es en este punto en que Dijkstra se detiene a criticar la antropomorfización del software, que considera una infantilización.

Finalmente el autor propone una currícula para una materia inicial de computación, comparable a Algoritmos y Estructuras de Datos I.

## 3. Resumen ejecutivo

- Las computadoras constituyen una novedad radical, y la ingeniería de software es un intento de tratar a los nuevos problemas como si fueran los ya conocidos. Como tal, es difícil que tenga éxito.
- Computar es manipular símbolos; otras visiones resultan limitantes en varios aspectos. Por esta razón, la educación debe enfatizar este aspecto en lugar de intentar simplificarlo en términos de otros más simples.