Prefacio

Luis Ángel Meza Chavarría

Durante mucho tiempo, el software se ha entendido principalmente como un instrumento que genera utilidad: un conjunto de instrucciones diseñadas para automatizar tareas, optimizar procesos y resolver problemas prácticos con eficiencia y precisión. Desde esta perspectiva, la programación se presenta como una actividad puramente técnica, subordinada al rendimiento medible. Sin embargo, esta forma de entender el software es, cuando menos, incompleta. Más allá de su función instrumental, el software también puede concebirse como un medio de expresión creativa.

Esta idea no es del todo nueva. En su conocida conferencia *Computer Programming as an Art* (1974), Donald Knuth argumentó que la programación no debe reducirse a la mera corrección técnica, sino que puede aspirar a la belleza, el estilo y la elegancia. Su afirmación abrió la puerta a pensar en el código no solo como una herramienta funcional, sino también como una forma de creación artística. En este sentido, el software puede apreciarse estéticamente, tanto en su construcción interna como en las experiencias que posibilita. Un algoritmo puede admirarse por su elegancia, un programa por la claridad de su estructura y una obra digital por la originalidad de su diseño y su fuerza expresiva.

En las últimas décadas, esta dimensión artística del software se ha vuelto más visible. Los videojuegos, el arte generativo, las instalaciones digitales interactivas, el *live coding* y las imágenes y sonidos producidos algorítmicamente muestran que el software no es solo una herramienta para producir arte, sino también un medio de creación en sí mismo, igual que un lienzo o un libro. Como argumentan Cox y McLean (2013), la programación puede funcionar como una práctica estética y política, ya que el acto de hacer software nunca es neutral: organiza posibilidades, impone estructuras y expresa maneras de ver y moldear el mundo.

Por otra parte, la formación en ingeniería suele enfatizar la resolución de problemas, la optimización y la precisión técnica. Si bien estas dimensiones son esenciales, no son suficientes por sí solas para ser un buen ingeniero. La ingeniería también requiere imaginación, creatividad en el diseño y la capacidad de pensar críticamente sobre el significado e impacto humano de lo que se construye. Por lo cual, reconocer el software como arte no significa negar su rigor, sino afirmar que el rigor y la creatividad pueden coexistir.

En este contexto, el octavo número de la Revista invita a reflexionar sobre las dimensiones estéticas y culturales del software. La intención es explorar al software no solo como un producto tecnológico, sino como una práctica creativa capaz de generar belleza, significado, y nuevas formas de experiencia cultural.

Referencias

Cox, G., & McLean, A. (2013). *Speaking code: Coding as aesthetic and political expression*. MIT Press.

Knuth, D. E. (1974). Computer programming as an art. *Communications of the ACM, 17*(12), 667-673. https://doi.org/10.1145/361604.361612

Maeda, J. (1999). *Design by numbers*. MIT Press.

McCullough, M. (1996). *Abstracting craft: The practiced digital hand*. MIT Press.

Paul, C. (2015). *Digital art* (3rd ed.). Thames & Hudson.