



Mark C. Marino
Critical Code Studies
Cambridge, Massachusetts, London, England
The Mit Press
2020
288 páginas

PALABRAS CLAVE: MARK MARINO – ESTUDIOS CRÍTICOS DEL CÓDIGO
KEYWORDS: MARK MARINO – CRITICAL CODE STUDIES

Todo código es político: instrucciones para abrir (y leer) la caja negra

Alejandro Del Vecchio¹

No resulta frecuente encontrarse ante la lectura de un texto fundacional. *Critical Code Studies* –como el título mismo del volumen parece sugerir– sienta las bases para consolidar un campo disciplinar emergente en la era de la revolución digital: los “estudios críticos del código” (ECC), que el propio Mark Marino define como:

an approach to code studies that applies critical hermeneutics to the interpretation of computer code, program architecture, and documentation within a sociohistorical context. CCS holds that the lines of code of a program are not value-neutral and can be analyzed using the theoretical approaches applied to other semiotic systems, in addition to particular interpretive methods developed specifically for the discussions of programs (39).

¹ Magister en Letras Hispánicas. Ayudante graduado en “Literatura y cultura latinoamericanas II” y miembro del grupo de investigación “Literatura y cultura latinoamericanas” dirigido por la Dra. Mónica Marinone y codirigido por la Dra. Gabriela Tineo (Universidad Nacional de Mar del Plata-CELEHIS). Profesor asociado en “Gramática castellana 1” y “Gramática castellana 2” (UCAECE). Contacto: adelvecchio@mdp.edu.ar

[una aproximación a los estudios del código que aplica la hermenéutica crítica a la interpretación del código informático, la arquitectura del software y su documentación, en un contexto sociohistórico. Los ECC sostienen que las líneas de código de un programa no son ideológicamente neutrales y pueden ser analizadas utilizando enfoques teóricos aplicados a otros sistemas semióticos, además de métodos interpretativos particulares desarrollados específicamente para el análisis de programas.]²

La publicación de Marino (artista, profesor de escritura, especialista en literaturas electrónicas) cristaliza más de quince años de reflexiones y debates cuyo objeto central de indagación –normalmente escatimado al ojo humano y confinado a una caja negra– es el código fuente de los lenguajes de programación, concebido como un reino semiótico distintivo, susceptible de ser leído y analizado como texto de cultura.

Si el software aparece entretejido en la vida contemporánea –económica, cultural, creativa, políticamente– “de maneras tanto obvias como casi invisibles” (como afirma el prólogo a la colección de “Software Studies” de la que el texto que nos ocupa forma parte) el tono inquietante del adjetivo “invisibles” evoca (a la vez que reaviva) la sospecha sobre los códigos ocultos en dispositivos y plataformas, y los intereses a los que responden. Sin acceder al código, por lo tanto, el trabajo del analista queda restringido a considerar solo los efectos de las operaciones aparentes que un programa efectúa. “Crear el software o ser el software” / “programar o ser programado” son las fórmulas que elige Douglas Rushkoff para advertirlo en otros términos. O –como señala Bifo Berardi en el prefacio a *Speaking Code: Coding as Aesthetic and Political Expression* (2012)– si podemos decir que el código nos habla (impregnando y formateando nuestras acciones), también “hablamos código” de diversas maneras, ya que el poder está cada vez más inscripto en el código (Cox y McLean, 2012: 7).

En el capítulo introductorio de *Critical Code Studies*, Marino anticipa el movimiento argumentativo del texto a partir de la consideración de pequeños “casos testigo” cuyo análisis configura de forma gradual, inductivamente, las tesis que intenta sostener, a la vez que apela a cierto asombro del lector poco familiarizado con el código fuente: el avance –como si del tráiler de una serie se tratara– del llamado “Climategate” (caso con ribetes detectivescos desarrollado en profundidad en el capítulo 4); un letrero, escrito en pseudocódigo, en defensa del proyecto Jan Lokpal, redactado por activistas de la sociedad civil india con la misión de crear un organismo independiente para investigar casos de corrupción; la

² Todas las traducciones son mías.

competencia en una entrevista de trabajo entre dos postulantes desafiados a escribir un algoritmo para generar anagramas; una aplicación web alojada en el repositorio GitHub para celebrar el número creciente de mujeres programadoras.

Estas anécdotas relevadas por el ensayista no solo refieren diversos modos en los que el código circula –no siempre se oculta tras una interfaz³–, sino que ponen en escena su materialidad y su recepción. Como advierte Marino, el código *fuera-de-sí* del letrero de la mujer india constituye un *codework* más que un programa, en tanto carece de *ejecutabilidad*, rasgo que lo distancia de la aplicación referida de Tapasweni Pathak, cuyo impacto reside justamente en su performatividad. La hipotética entrevista laboral, en cambio, focaliza el riesgo de las decisiones estilísticas y retóricas asumidas por quien programa (por ejemplo legibilidad versus depuración).

A continuación, bajo el sugerente título de “A Manifesto”, el capítulo 2 actualiza un ensayo pionero (con inevitables, aunque parciales predecesores), publicado en 2006 en *Electronic Book Review*. Allí, el propio Marino proclamaba la necesidad de sacar el código de su caja negra para explorar el modo en que su complejo sistema semiótico produce significación, en tanto texto, mientras circula en contextos diversos:

As a new media scholar trained in literary theory, I would like to propose that we no longer speak of the code as a text in metaphorical terms, but that we begin to analyze and explicate code as a text, as a sign system with its own rhetoric, as semiotic communication that possesses significance in excess of its functional utility (39).

[Como estudioso de los nuevos medios formado en teoría literaria, me gustaría proponer que ya no hablemos del código como texto en términos metafóricos, sino que comencemos a analizar y explicar el código como texto, como sistema de signos con su propia retórica, como comunicación semiótica que posee significado que excede su utilidad funcional.]

Todo el libro de Marino puede pensarse como una expansión de esta tesis, neurálgica para los “estudios críticos del código” (ECC), cuyas implicancias teóricas para redefinir categorías clásicas como autor, lector, escritura, lectura (entre otras) son acaso inconmensurables. No obstante, en tanto “el código es a la vez lo que es y lo que hace” (51) [code is at once what it is and what it does], leerlo *barthesianamente* como texto no implica negar su especificidad originaria. En

³ Son anécdotas idóneas para evidenciar, además, lo que Rita Raley llama “impetu del *code art* contemporáneo” que consistiría, en pocas palabras, en “revelar códigos, hacer visibles al espectador los mecanismos de producción” (Raley, 2016).

cualquier caso, el significado del código está marcado por la tensión entre ambigüedad (porque es social y polisémico) y a la vez precisión (porque es tecnológico y su sintaxis estricta).

El capítulo 3, que inicia la demostración de algunos de los enfoques y métodos de los ECC, analiza un ejemplo de código escrito en Java para un proyecto de arte *hacktivista* denominado *Transborder Immigrant Tool* (TBT), una aplicación para teléfonos móviles diseñada con el objetivo de ayudar a migrantes, en el último tramo de sus viajes, a cruzar la frontera. La APP brinda instrucciones para encontrar agua, a la vez que reproduce poemas y consejos para sobrevivir a los peligros. Según Marino, explorar críticamente TBT permite entender formas en las que el código contiene simulaciones del mundo (con sus ideologías y contraideologías) y cómo el nombre en lengua natural de métodos, contantes, variables y otros elementos habilita una lectura textual del código.

Ante el capítulo 4, en el que Marino expone en detalle el escándalo de 2009 conocido como “Climategate”, el lector agradece la pulsión narrativa que rige su escritura. Como si se tratara de un relato policial, el ensayista documenta los avatares de un caso originado en la filtración de miles de correos electrónicos, archivos y –sobre todo– fragmentos de código de un software de modelado climático. El *misreading* (intencional o no) del código fuente condujo a un grupo de negacionistas del cambio a creer haber encontrado pruebas irrefutables de que los científicos manipulaban datos para engañar al público. Como concluye Marino, el “Climategate” revela cómo el código deviene herramienta del discurso político y, sacado de contexto, genera confusiones capaces de influir en el debate público, por lo que resulta imperiosa la formación de académicos con competencias adecuadas para leerlo críticamente.

La siguiente sección, “FLOW-MATIC”, examina proyectos artísticos que –a partir de la tensión entre código fuente y lenguajes naturales– proponen alternativas al predominio de herramientas de programación basadas en inglés, como forma de contrarrestar sus efectos imperialistas y coloniales sobre quienes programan. El sexto capítulo, asimismo, aborda vínculos entre praxis artística y teoría a partir de la exploración del código de un programa de gráficos por computadora llamado *raytracer* [“trazador de rayos”] escrito por Friedrich Kittler.

La relación código y poesía –que nos importa en particular– es indagada en el capítulo posterior, donde Marino analiza las reescrituras y remix del poema generativo *Taroko Gorge* de Nick Montfort para destacar, entre otros aspectos, la importancia del trabajo colaborativo en el arte digital. *Taroko Gorge* puede leerse, además, como síntoma de la emergencia de un “momento literario”, en el que poetas programadores interactúan con lenguaje y código para generar textos.

Por último, el ensayista dedica el capítulo final a desarrollar un panorama general de los ECC, enumerando algunas acciones clave para su futuro: enriquecer los planes de estudios de las carreras humanísticas y de las ciencias de la computación / apoyar la investigación en humanidades digitales / inspirar nuevos trabajos creativos basados en código fuente. En el apartado “How to Interpret Code” [Cómo interpretar código], el autor propone incluso una extensa lista preliminar –y por ende abierta– de enfoques y técnicas probadamente eficaces y fructíferos para la interpretación de código. No obstante, señala Marino:

Code itself is neither the end nor the beginning of this reflection, but as an expression of thought, as a trace record of labor and development history, as an artistic medium, and as a connection point in human-machine assemblages, code offers an opportunity to reflect on technoculture with symbols that are at once completely unambiguous and at the same time open to interpretation (239).

[El código en sí mismo no es ni el principio ni el final de estas reflexiones, pero como expresión del pensamiento, como registro de la historia del trabajo y el desarrollo, como medio artístico y punto de conexión en los ensamblajes hombre-máquina, el código ofrece una oportunidad para reflexionar sobre la tecnocultura, a través de símbolos que son completamente inequívocos y a la vez están abiertos a interpretación.]

De allí que la figura del “viaje de descubrimiento” (“journey into code”, dice Marino) resulte idónea para poner en escena la potencialidad de territorios aún apenas explorados y en constante expansión. En este sentido, desde una escritura que equilibra rigor académico y pulsión narrativa, el texto estimula y exhorta al lector a sumarse a una travesía marcada por la necesidad de horadar los códigos ocultos, ya que modelos económicos y sistemas de valores, entre otras facetas de la vida social, se cifran allí.

A modo de conclusión: *Critical Code Studies* constituye el grado cero en la teorización de una práctica, tensionada entre la lectura contextual y el *close-reading*, que se orienta a analizar fenómenos culturales a través del código y su sistema paratextual. El adjetivo “críticos”, por su parte, inscribe su arco conceptual en una genealogía vinculada a teorías capaces de deconstruir mecanismos opresivos, desde el feminismo al marxismo, y desde los estudios *queer* a los postcoloniales, entre otras vertientes.

Referencias bibliográficas

Cox, G. y Mclean, A. (2012). *Speaking Code: Coding as Aesthetic and Political Expression*. Cambridge, Massachusetts, London, England: The MIT Press.

Raley, R. (2016). “Code.surface| |Code.depth”. *Dichtung Digital. Journal für Kunst und Kultur digitaler Medien*. Nr. 36, Jg. 8.

Rushkoff, D. (2010). *Programa o serás programado. Diez mandamientos para la era digital*. Barcelona: Penguin Random House.