

Guzmán Contreras Amabel Vanessa

Resumen del artículo:

"Los algoritmos de nuestras vidas"

El autor habla de como el software y los algoritmos, están detrás de todo ya que no solo son herramientas que usamos, sino que ahora son una parte fundamental de nuestra vida.

También menciona lo acostumbrados que estamos a que los algoritmos decidan qué vemos en redes y sobre por qué no nos cuestionamos sobre ello. Aunque ofrecen ventajas como recomendaciones útiles, también implican riesgos como la manipulación.

Dice que no debemos de ver el software solo como una herramienta, sino como una forma de "habla", ya que los algoritmos no solamente procesan datos, sino que también moldean nuestra realidad.

En conclusión el autor nos invita a cuestionarnos sobre la tecnología, a entender cómo funciona y no simplemente aceptarla. Y que si los algoritmos van a tomar el rumbo de nuestra vida, deberíamos al menos ser conscientes de ello.

Escogí este artículo porque me llamo la atención su título, ya que habla sobre los algoritmos y cómo influyen en nuestra vida diaria.

Link del artículo

<https://www.redalyc.org/pdf/935/93552794004.pdf>



CIC. Cuadernos de Información y
Comunicación

ISSN: 1135-7991

cic@ccinf.ucm.es

Universidad Complutense de Madrid
España

Manovich, Lev
Los algoritmos de nuestras vidas
CIC. Cuadernos de Información y Comunicación, vol. 22, 2017, pp. 19-25
Universidad Complutense de Madrid
Madrid, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=93552794004>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



Los algoritmos de nuestras vidas¹

Lev Manovich²

Propuesto: 10 de octubre de 2016 / Aceptado: 25 de octubre de 2016

Resumen. El autor reflexiona sobre la llegada de nuevas categorías de reflexión en el mundo de los estudios de los medios, y cómo el *software* y los códigos de programación del lenguaje digital se insertan en la cultura de masas para cambiar radicalmente nuestra capacidad de análisis y de comprensión crítica de la evolución de la cultura digital. Plantea la necesidad de pensar en los “actos” de *software* y su carácter performativo y dinámico, considerándolos un “habla” más que un lenguaje. Y apela a la necesidad de capacitarse para comprender críticamente las dimensiones de poder que este nuevo instrumento comunicativo tiene sobre nuestras vidas.

Palabras clave: lenguaje digital; *software*; algoritmo; actos de *software*; comunicación; internet; código.

The algorithms of our lives

Abstract. The author reflects upon the arrival of new categories of elements in media studies, and how software and programming codes of digital language have come to mass culture to change dramatically our ability to analyze and to criticize the evolution of culture. He pleads for the consideration of software as a performative act of speech, considering it dynamically. And he remarks the need to acquire the competence to understand critically the dimensions of power that this new instrument of communication has over our lives.

Keywords: digital language; software; algorithm; software acts; communication; the Internet; code.

Cómo citar: Manovich, L. (2017). Los algoritmos de nuestras vidas, en *CIC. Cuadernos de Información y Comunicación* 22, 19-25.

En 2002 residiendo en Colonia, en Alemania, visité una de las mejores librerías de la ciudad, especializada en títulos de humanidades y artes. Sin embargo, ni un solo

¹ Traducción de Eva Aladro Vico.

² Graduate Center de la City University de Nueva York <http://manovich.net/>. Lev Manovich es profesor de Humanidades Digitales e Informática en el Graduate Center de la City University de Nueva York, y director de la Software Studies Initiative en el Instituto de Telecomunicaciones y Tecnología de la Información de California (Calit2) y el Graduate Center. Su libro más reciente es *Software Takes Command. Extending the Language of New Media* (Bloomsbury Academic 2013) traducido al español en ese mismo año por la UOC y publicado como “softbook” de libre acceso en la web igualmente. Manovich es una de las mentes más creativas en el análisis de la revolución digital, aportando su vasta formación cultural a un fino enfoque de los nuevos fenómenos comunicativos. A él debemos la acuñación y justificación académica de la expresión “lenguaje digital” (en su clásico texto *El lenguaje de los nuevos medios de comunicación* (2001) publicado en España en 2005) y una ingente cantidad de aportaciones científicas en este sector que están accesibles en su web <http://manovich.net/>. El presente artículo se publica en 2013 en *The Chronicle*. Email: manovich.lev@gmail.com

libro trataba una de las claves directoras de la “era del ordenador”: el *software*. Me puse a buscar en todos los índices de libro tras libro: no aparecía “software” en ninguna parte.

Sin embargo en los años 90 todas las herramientas basadas en el *software* se habían incorporado a todas las áreas de producción y diseño de medios profesionales. En 2000 esos desarrollos habían conducido a centenares de millones de personas a escribir *blogs* o *twits*, subiendo fotos y vídeos, intercambiando textos en *Scribd* y usando herramientas gratuitas que 10 años antes hubieran costado decenas de miles de dólares.

Gracias a los desarrollos encabezados por *Google*, el mundo hoy opera en aplicaciones *web* que permanecen siempre en versiones *beta*. Pueden actualizarse en cualquier momento en servidores remotos sin que los consumidores deban hacer nada –y de hecho, *Google* está revisando su algoritmo de búsqueda aproximadamente unas 600 veces al año. Bienvenidos al mundo del cambio permanente –un mundo definido no por las pesadas máquinas industriales, que rara vez se modifican, sino por el *software* que fluye continuamente.

El *software* se ha convertido en un lenguaje universal, la interfaz entre nuestra imaginación y el mundo. Lo que fueron la electricidad o el motor de combustión para el siglo XX, lo es el *software* al comienzo del siglo XXI. Pienso en él como en una capa que permea las sociedades contemporáneas. Si queremos entender las técnicas actuales de comunicación, de representación, de simulación, de análisis, de decisión, de memoria, de visión, de escritura y de interacción, debemos comprender qué es el *software*.

Pero aunque los académicos y teóricos de los nuevos medios han cubierto todos los aspectos de la revolución de las tecnologías de la información, creando campos como el estudio de la cibercultura, los estudios sobre Internet, los estudios de los juegos, la teoría de los nuevos medios, o las humanidades digitales, han puesto comparativamente poca atención en el *software*, el invento que dirige prácticamente todo cuanto se estudia.

Ya es hora de que lo hagan.

Consideremos el moderno “átomo” de la creación cultural: un “documento”, es decir, contenido almacenado en una forma física, que es difundido a consumidores mediante una copia física (libros, películas, registros sonoros) o transmisión electrónica (televisión). En la cultura del *software*, ya no tenemos “documentos”. En su lugar, tenemos “actos de *software*”.

Si es usted académico y trabaja con *Google* o con *Facebook*, tendrá mucha ventaja sobre los colegas para entender esta distinción.

Uso el término “acto” porque lo que experimentamos se construye en tiempo real con el *software*. Tanto si exploramos una página *web*, como si jugamos con un videojuego, o usamos una aplicación del móvil para localizar a unos amigos cercanos o un lugar para comer, estamos interactuando con los resultados dinámicos de la computación.

Aunque usemos también documentos estáticos, un académico no puede simplemente consultar un PDF o un archivo JPEG del mismo modo en que los críticos literarios del siglo XX examinaban una novela, película o programa de TV. El *software* a menudo no tiene límites finitos. Por ejemplo, un usuario de *Google Earth* puede experimentar diferentes “Tierras” cada vez que ella o él abran la aplicación. *Google* actualiza cada poco las fotografías de satélite o añade nuevas “Street Views” y edi-

ficios en 3D. A cada instante, el usuario de la aplicación puede también cargar más datos geoespaciales creados por otros usuarios o compañías.

Google Earth no es simplemente un “mensaje”. Es una plataforma para usuarios que está en construcción permanente. Y aunque podamos establecer cierta continuidad con el trabajo de recreación dinámica de los medios comerciales del siglo XX –el *pop art* y la apropiación, la música, la fanficción y el video, y demás- las diferencias son más numerosas que las similitudes.

Incluso si un usuario está trabajando solamente con un archivo local en su ordenador, esa experiencia no puede definirse sencillamente por la organización y contenido del archivo. El usuario puede navegar el documento, escogiendo qué información leer y la secuencia en que hacerlo. (En *Google Earth*, puedo hacer zoom a mayor o menor distancia, cambiando de la vista de pájaro de un área, a sus detalles, y también puedo cambiar a diferentes tipos de mapas.)

Más esencial, el *software* no está anclado a ningún documento ni máquina. Las nuevas herramientas pueden añadirse fácilmente sin necesidad de cambiar los documentos mismos. Con un simple “clic”, puedo añadir botones de interacción a mi *blog*, permitiendo así nuevos modos de circular su contenido. Cuando abro un documento de texto en un Mac en *OS Preview* puedo subrayar, añadir comentarios y enlaces, dibujar y añadir burbujas con ideas. *Photoshop* me permite salvar mi edición en “capas de ajuste” separadas, sin modificar la imagen original. Y así sucesivamente.

Todo esto requiere un nuevo modo de analizar los medios y la cultura. Desde los primeros años de la década del 2000, algunos de nosotros (la mayoría provenientes de los estudios sobre medios y de arte digital) hemos trabajado en afrontar este reto. Hasta donde yo sé, he sido el primero en usar los términos “*estudios de software*” y “*teoría del software*” en 2001. El campo de los estudios de *software* gradualmente cobró forma en mitad de los 2000. En 2006, Matthew Fuller, autor del texto pionero *Behind the Blip: Essays on the Culture of Software* (Sagebrush Education Resources, 2003), organizó el primer Taller de Estudios de Software en Rotterdam. “El *software* es a menudo el punto ciego en la teorización y el estudio de los medios digitales y en red.” En cierto sentido, ahora cualquier trabajo intelectual es un “estudio de *software*”, en tanto es el *software* el que proporciona sus medios y su contexto, pero hay pocos sitios en donde su naturaleza específica, su materialidad, se estudien excepto como asunto de ingeniería.

En 2007, comenzamos a desarrollar la Iniciativa de Estudios de Software en la Universidad de California en San Diego, y en 2008 llevamos a cabo el segundo taller de estudios de *software*. MIT Press ofrece una serie de estudios en este campo, y un número creciente de libros en otros campos (teoría de medios, estudios sobre plataformas, humanidades digitales, estudios de Internet, estudios de los juegos) nos ayudan a comprender mejor el papel que juega el *software* en nuestras vidas. En 2011, Fuller y otros investigadores en Gran Bretaña comenzaron *Computational Culture*, revista académica de acceso abierto.

Hay mucho por hacer. Una cuestión que me interesa especialmente es cómo pueden contribuir los estudios de *software* al “big data” –analizar vastas cantidades de datos- en campos como las humanidades digitales, la ciencia social computacional, y la computación social. Aquí se dan una serie de cuestiones clave relacionadas con los grandes datos culturales que los estudios en torno al *software* podrían ayudar a responder:

¿Qué son los “datos” de medios interactivos? ¿El código de *software* tal y como se ejecuta, los registros de las interacciones de los usuarios (por ejemplo, los “clics” y movimientos de cursor), las grabaciones de video en la pantalla de un usuario, o la actividad cerebral de un usuario captada por un EEG o por un FMRI? ¿Todo ello, o algo más?

Para usar terminología de la lingüística, en lugar de estudiar el código de *software* como un lenguaje, debemos contemplarlo como un *habla*.

En estos pasados años un número cada vez mayor de académicos en humanidades digitales han empezado a usar herramientas informáticas para analizar grandes cantidades de artefactos culturales digitalizados, como por ejemplo las novelas del siglo XIX o las cartas de pensadores ilustrados. Siguen enfoques tradicionales de las humanidades –observando los objetos culturales (en lugar de la interacción de las personas con dichos objetos). Lo que ha cambiado es simplemente la escala, pero no el método.

El estudio de la cultura del *software* exige una metodología de humanidades diferente en su fundamento. Necesitamos poder registrar y analizar las experiencias interactivas, siguiendo a los usuarios individuales conforme navegan por las páginas o juegan con un videojuego, para estudiar a los diferentes jugadores, y no simplemente analizando nuestro propio juego como base del estudio, u observar a los visitantes de una instalación interactiva conforme exploran las posibilidades definidas por el diseñador –posibilidades que se convierten en sucesos reales solamente cuando los visitantes actúan con ellos.

En otras palabras necesitamos pensar cómo representar adecuadamente los “actos de software” en tanto “datos”. Algunas respuestas pueden provenir del campo de la interacción humano-ordenador, en el cual investigadores académicos y estudios de empresas privadas están analizando las interfaces del ordenador. Los objetivos de tal investigación, sin embargo, son normalmente prácticos: identificar los problemas de las nuevas interfaces y arreglarlos. Los objetivos de los estudios de humanidades digitales de los medios interactivos serán diferentes –entender cómo construyen las personas los significados en sus interacciones, y cómo sus experiencias sociales y culturales se ven mediadas por el *software*. Así que necesitamos desarrollar nuestros propios métodos de transcribir, analizar, y visualizar experiencias interactivas. Junto con el Experimental Game Lab, dirigido por Sheldon Brown, por ejemplo, mi laboratorio analizó las experiencias de cientos de usuarios de *Scalable City*, una gran instalación artística virtual compleja creada en el laboratorio de Brown. Uno de nuestros objetivos era ayudar a los futuros usuarios a conseguir experiencias interactivas más completas.

¿Quién tiene acceso a los registros en detalle de las interacciones con artefactos culturales y servicios en la web, y cuáles son las consecuencias de poder por primera vez analizar esos datos?

Desde los primeros días de las interfaces de interacción entre humanos y ordenadores, el seguimiento de la interacción de los usuarios con el software fue fácil. ¿Por qué? Porque el *software* continuamente registra las operaciones como los movimientos de ratón, la presión sobre teclas, la selección de los menús, los gestos del dedo sobre una superficie táctil, o los comandos de voz.

El cambio del ordenador de consola a la computación en red en los años 90 ha hecho que las posibilidades ya existentes de registrar y almacenar los datos de entrada de los usuarios se multipliquen y conviertan en componente fundamental del

“complejo *software-media*”. Dado que los sitios *web* dinámicos y los servicios en Internet (la tienda *online* de Amazon, los blogs personales que usan el sistema *Blogger*, los juegos en línea, etc) operan mediante *software* residente en los servidores de las compañías, es fácil almacenar los detalles de las interacciones del usuario. Cada servidor *web* conserva información detallada de todas las visitas a un sitio determinado. Una categoría separada de *software* y de servicios, ejemplificada por *Google Analytics*, emerge para ayudar a afinar en detalle el diseño de un *blog* o de una página *web*.

Las redes sociales y sus compañías dejan disponible para sus usuarios cierta información almacenada sobre interacciones de los visitantes a las páginas, *blogs* o cuentas de sus empresas, y proporcionan también visualizaciones interactivas para ayudar a la gente a conocer qué asuntos publicados son más populares, y de dónde vienen sus visitantes. Sin embargo, usualmente las empresas conservan para ellas los registros más detallados. Por ello, si usted es uno de los pocos sociólogos y expertos que trabaja dentro de un gigante como *Facebook* o *Google*, tendrá una enorme ventaja sobre sus colegas de la academia. Podrá preguntar cosas que los otros no pueden. Esto puede crear una brecha real en el futuro entre investigadores académicos y corporativos. Mientras estos últimos serán capaces de analizar patrones culturales y sociales tanto en niveles supermicro como supermacro, los primeros tendrán una “lente” normal, que no podrá acercarse demasiado ni alejarse lo suficiente hasta una visión planetaria.

¿Quién se beneficia del análisis de las actividades culturales de cientos de millones de personas? La ubicación automática de anuncios en las redes de *Google*, *Facebook* o *Twitter* ya usa tanto los textos de los *post* de los usuarios como los correos y otros datos, pero llegar a conocer cómo interactúan cientos de millones de personas con imágenes y con videos en las redes sociales no solamente sirve a los anunciantes para colocar mejor su publicidad visual, sino a los académicos para plantear nuevas cuestiones.

¿Podemos analizar el código de los programas de *software*? No es tan sencillo como se pueda pensar. El código mismo son “*big data*”.

Los programas del primer *software* como los videojuegos de los años 70 eran relativamente cortos. Sin embargo, cualquier servicio *web* comercial contemporáneo o sistema operativo tiene un código de programación tan largo y completo que no se puede leer e interpretar como si fuera un cuento corto. Si bien Windows NT 3.1 (1993) contenía aproximadamente cinco millones de líneas fuente de código, Windows XP (2001) tiene unos 40 millones. MAC OS es aún más grande, con OS X 10.4 (2005), llegando a los 86 millones de líneas. El número estimado de líneas en Adobe Creative Suite 3 (que incluye Photoshop, Illustrator, y otras aplicaciones populares de producción de medios) tiene 80 millones.

El gradual desplazamiento del *software* de aplicación a la red también trae un nuevo conjunto de consideraciones. Los servicios de la red, las aplicaciones, los sitios dinámicos, usan frecuentemente arquitecturas de programación multi-capas, en las que un número de módulos separados (por ejemplo un cliente *web*, un servidor de aplicación, y una base de datos) trabajan en conjunto. Especialmente en el caso de grandes sitios comerciales como *Amazon.com*, lo que el usuario percibe como una única página *web* implica seguramente continuas interacciones con docenas e incluso centenares de procesos de *software* separados.

La complejidad del *software* a gran escala y su arquitectura distribuida de gran complejidad plantea un serio desafío a nuestra idea de “leer el código”. Sin embargo,

incluso aunque un programa sea relativamente corto y si un crítico cultural comprende exactamente lo que el programa se supone que hace, esta comprensión de la estructura lógica del programa no puede traducir los términos de la visión que el usuario tiene de su experiencia.

La atracción de “leer el código” en el enfoque de las humanidades crea la ilusión de que tenemos ante nosotros un texto estático y definido que podemos analizar —es decir, una lista de programa—. Pero tenemos que aceptar la variabilidad fundamental de los verdaderos “actos de *software*”. Así, más que analizar el código como una entidad abstracta, debemos en su lugar observar cómo se ejecuta, o “actúa”, en las sesiones particulares de los usuarios. Para usar los términos de la lingüística, en lugar de pensar en el código como en un lenguaje, debemos estudiarlo como *habla*.

Algunos investigadores, como Mark Marino y otros, trabajando en “estudios críticos de código”, están promoviendo ideas sutiles y rigurosas teóricamente sobre lo que significa “leer el código”, así que mi crítica va solamente dirigida a la versión infantil de la idea que a veces he encontrado en las Humanidades.

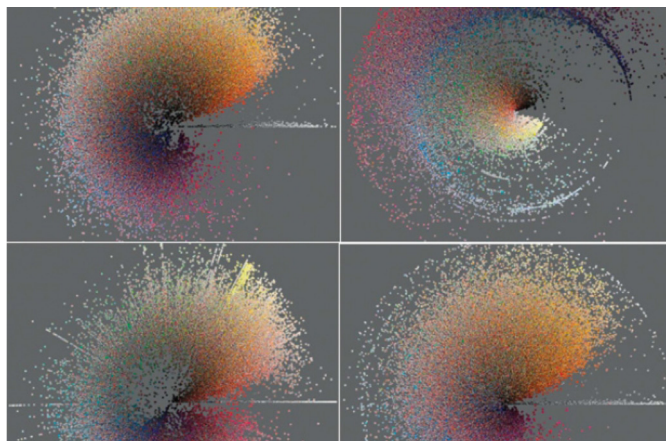
El desarrollo de métodos para estudiar el *software* contemporáneo de modo que pueda discutirse en artículos, conferencias y debates por parte de no-programadores, es una tarea clave para los estudios de *software*. Sin embargo, dada tanto la complejidad de los sistemas como el hecho de que, al menos actualmente, solo una pequeña parte de los investigadores de medios y cultura están familiarizados con la ingeniería de *software*, no creo que podamos resolver este problema en un corto plazo de tiempo.

Y sin embargo, confrontarlo es crucial, no solamente para la academia sino también para la sociedad en su totalidad. ¿Cómo podemos discutir públicamente las decisiones tomadas mediante algoritmos de búsqueda de *Google*, o los algoritmos de control de *Facebook*, sobre lo que se muestra o no se muestra de nuestras propias búsquedas de información? Incluso aunque estas empresas hicieran todo su *software* de fuente abierta, su tamaño y complejidad haría que la discusión pública se viera muy limitada.

Aunque algunos de los detalles de estas compañías populares de la red se publiquen en artículos académicos escritos por los investigadores que trabajan en ellas, solamente las personas con conocimientos en informática y estadística pueden entenderlos. Además, muchos servicios populares de *software* usan tecnologías de automatización que a menudo presentan soluciones de “caja negra” (aunque el *software* consiga resultados deseados, no sabemos las reglas que sigue).

Conforme más y más experiencias culturales, interacciones sociales y decisiones se ven gobernadas por los sistemas de *software* a gran escala, la habilidad de los no-expertos para discutir estos sistemas se va haciendo crucial. Si reducimos cada sistema complejo a una descripción de su algoritmo de una sola página, ¿podremos capturar y analizar con ello suficiente conducta del software? ¿O los matices de las decisiones particulares de cada caso particular del *software* se perderán?

El papel de los estudios de *software* no es responder a ésta u otras cuestiones sobre nuestro nuevo mundo interactivo, sino más bien articularlas y ofrecer ejemplos de cómo se deben de encarar. Y animar a los estudiosos de todas las disciplinas a que piensen sobre los cambios en el *software* que estudian y en cómo llevan a cabo ese estudio.



En el Proyecto *Phototrails* (phototrails.net),³ que he creado yo mismo, Nadav Hochman y Jay Chow, visualizamos patrones del uso de *Instagram* en 2.3 millones de fotos de 13 ciudades globales. En el artículo que desarrollamos con el proyecto, intentamos combinar dos lados en espejo de los estudios de *software*, reflexionando sobre las interfaces de *software* y cómo influyen en lo que hacemos y al mismo tiempo estudiando conductas a gran escala de muchos usuarios de *software*. Una de las preguntas que nos planteamos, fue cuántas diferencias es posible encontrar entre los usuarios de las diferentes ciudades, a pesar de que todos ellos usan la misma aplicación *Instagram* que viene con su propio y vigoroso “mensaje” (todas las fotos tienen la misma forma cuadrangular y el mismo tamaño, y todos los usuarios acceden al mismo conjunto de filtros de edición para hacer sus fotos más estéticas de las mismas maneras). Aunque encontramos pequeñas diferencias sistemáticas en las fotos de cada ciudad, el uso del *software* de Instagram permanecía notablemente similar en todos los sitios.

¿Cómo influye el software que usamos en lo que expresamos e imaginamos? ¿Debemos seguir aceptando las decisiones que los algoritmos toman por nosotros, si no sabemos cómo operan? ¿Qué significa ser un ciudadano de una sociedad dominada por el *software*? Estas y otras cuestiones importantes están esperando ser analizadas.

³ Phototrails.net. Corresponde a las imágenes de nuestra portada. Se representan las variaciones y semejanzas de los patrones de post en Instagram organizados por brillo y color, de ciudades como Tokio, Nueva York, Bangkok y San Francisco.