

# La colaboración en la cultura del software: presentación del problema, resultados intermedios y exploración inicial de un estudio de caso en la cultura “maker”

Luis Fernando Medina Cardona

□

## RESUMEN

Este artículo discute brevemente el proyecto de doctorado "Prácticas de colaboración abierta en la cultura del software y su impacto en la sociedad en red". Para ello presenta su justificación, preguntas iniciales, cómo se ha abordado metodológicamente y algunos hallazgos intermedios. Dado el carácter teórico del proyecto, dichos resultados obedecen a las relaciones discursivas que se han encontrado con el propósito principal de mapear prácticas colaborativas de la cultura del software en otros campos. Con este fin se ha seleccionado un estudio de caso que exhibe el modo en que se trasladan las llamadas "metáforas de software"; específicamente, se escogió la "cultura maker" como un ejemplo que no corresponde al desarrollo del software, pero que presenta una fuerte influencia de los medios digitales aplicados a la fabricación de prototipos físicos de objetos. Dicha elección revela el carácter inestable del software y aborda directamente la dicotomía material/inmaterial.

*Palabras Clave*—Colaboración en el software, estudios de software, software abierto o de código abierto, teoría de medios.

## I. INTRODUCCIÓN

A medida que el software —y el código mediante el cual este se hace objeto y procedimiento simultáneamente— se convierte en un medio transversal a nuestras vidas, más disciplinas se interesan en caracterizarlo desde una perspectiva académica y científica [1]. Así, el software como artefacto tecnológico ha trascendido el área exclusivamente de las ingenierías o de las ciencias de la computación, donde la preocupación principal radica en su eficiencia y metodologías de construcción, para incidir en otras visiones más holísticas, como la antropológica, sociológica y cultural. Esta travesía interdisciplinar se hace necesaria al observar dos hechos notorios: el primero es que vivimos en una sociedad donde la mayoría de las facetas de nuestro acontecer diario dialogan —en distintos niveles de transparencia— con dispositivos controlados por software, alimentándolos con información y siendo afectados por el

resultado de estos dispositivos [2]. El segundo es que cualquier disciplina aislada se torna incompleta para explicar, caracterizar e intervenir el software como objeto de estudio[3].

En este marco general se encuentra el proyecto actual de doctorado titulado “Prácticas de colaboración abierta en la cultura del software y su impacto en la sociedad en red” el cual podría ubicarse, de manera más precisa, dentro de una subárea relativamente reciente de la teoría de medios que lleva por nombre teoría de software [4]. El proyecto actualmente está saliendo de su parte inicial, habiendo pasado la fase de estado del arte y recolección bibliográfica. A pesar de ser de talante exclusivamente teórico, arroja unas primeras reflexiones que pueden ser consideradas pequeños resultados, debido a su importancia como posibles ejes rectores de la narrativa definitiva del proyecto. El proyecto se pregunta sobre la colaboración en una esfera amplia denominada cultura del software [5] y su impacto en lo que se ha llamado sociedad en red. Por colaboración debe entenderse metodologías de desarrollo, artefactos (software, lenguajes de programación, herramientas de apoyo a la ingeniería de software) y patrones de corte social (esquemas de coordinación, mecanismos de comunicación, distribución del trabajo) que son empleados en cualquier etapa del proceso de desarrollo de software moderno. Si bien el proyecto tiene un especial interés en lo que se ha agrupado bajo la sigla Floss (free/libre open source software – software libre y de fuente abierta) [6], razones históricas apuntan hacia la preferencia de una noción más general como la de colaboración. Así, su propósito es indagar la naturaleza y evolución de la colaboración en el software y la transformación de muchas de sus prácticas en una especie de metáfora que ha sido trasladada a otros entornos asociados a la cultura digital [7]. Justamente, el impacto hace referencia a este proceso de extrapolación de sentido en una sociedad que cuenta con redes, particularmente Internet, como uno de sus medios de trabajo más importantes y que será analizado a la luz de la selección de un estudio de caso que permita visualizar las inestabilidades propias de todo proceso de traducción; sobre todo en un entorno cultural que, como el del software, no es susceptible de transformaciones isomórficas deterministas. En palabras más sencillas, cuando se habla de colaboración modelada por las conductas de software en otras disciplinas siempre habrá asunciones preestablecidas que provocan que cada actor fabrique su propia versión de la colaboración enmarcándola con mayor o menor dificultad como “una

□

Autor: lfmedinac@unal.edu.co, Profesor, Escuela de Cine y Televisión, Universidad Nacional de Colombia. Estudiante de Doctorado en Artes y Medios en la Universidad de Artes y Medios de Colonia, Alemania (Kunsthochschule für Medien - Köln). Se agradece al servicio de intercambio académico alemán DAAD, a la Universidad Nacional de Colombia y a Colfuturo por el financiamiento.

colaboración inspirada por la cultura del software”. Es de notar que el estudio de caso seleccionado es de gran actualidad y cuenta con la ventaja de revelar muchas de las llamadas “inestabilidades” del software [8], como se explicará brevemente más adelante.

## II. HIPÓTESIS

A partir de lo enunciado en la introducción, puede formularse la pregunta de investigación así: ¿cómo evolucionaron las prácticas, metodologías y artefactos que permiten la colaboración en el desarrollo de software hasta que esta tomó la forma de la actual metáfora de lo libre y abierto y cuáles son los problemas de trasladar esta metáfora y su ética de trabajo a un campo externo al software? Debe verse que la pregunta asume la existencia de dicha metáfora; no obstante, a lo largo del desarrollo del trabajo y, en especial, del estudio de caso se confirmará su existencia. La hipótesis central subyace en la siguiente afirmación: la colaboración en la cultura del software, esto es, específicamente en el desarrollo y distribución de programas informáticos, siempre existió; sin embargo, adquirió su forma contemporánea a partir de la conjunción entre la cultura del software y la cultura de las redes, la cual sucedió desde mediados de los años 80 y se afianzó con la masificación de Internet en los años 90. Dos afirmaciones o subhipótesis complementan el juicio anterior. Primero, la colaboración es inestable. Al igual que el software, la colaboración en el desarrollo de programas puede calificarse como tal y no obedece a principios exclusivamente deterministas; si bien la aparición de la colaboración delata orígenes prácticos aprovechados casi por accidente por las compañías de software como estrategia de mercadeo [9], también hay una vertiente casi opuesta al pragmatismo capitalista, que incorporó la tradición hacker con el activismo político [10]. Segundo, y de manera complementaria, las redes de computadores son un ejemplo primigenio de la metáfora de la colaboración extrapolada al campo de las telecomunicaciones. La articulación entre software y redes generó un círculo armonioso que fomentó la colaboración, ya que la cultura de estándares abiertos que triunfó en Internet es heredera directa de la cultura del software, facilitando así la combinación de software y redes en un motor de creación y distribución de software compartido [11]. El estudio de caso dará cuenta también de la inestabilidad misma, no solo del software, sino del traslado de sus metáforas a otros ámbitos, aspecto que controvierte la superficialidad con la que a veces es tratado el tema.

## III. METODOLOGÍA

La metodología escogida reúne una serie de aproximaciones que buscan abordar tanto la parte técnica como cultural, así como un mapeo por medio de un estudio de caso. Antes de proceder a una descripción de los elementos principales de dicha metodología, es necesario detallar la contribución específica buscada, dada la serie de estudios que en los noventa florecieron generosamente sobre software libre. Como ya se ha aclarado, aunque forma una importante parte de la

motivación, el denominado Floss no es el centro del estudio, ya que la consolidación del concepto solo ocurrió en los años ochenta, obedeciendo a toda una serie de fuerzas de mercado, tecnológicas y sociales [12]. Por este motivo, el proyecto no es otro recuento de la saga del software libre y sus productos, sino más bien un análisis histórico y comparativo de los desarrollos, hitos y productos que provocaron la aparición del software libre como tal. Existen dos narrativas bien generalizadas al respecto, las cuales aparecen dependiendo de la perspectiva del autor y, muchas veces, de la característica que se quiera resaltar. De un lado, algunas narrativas señalan que el concepto del software libre no es nuevo y que las prácticas que lo precedían describían una especie de edén del código donde este se compartía libremente, relegando al software propietario al papel de una “anomalía” del mercado [13]. De otro lado, otros enfoques promueven el software libre como una revolución sin precedentes, muestra de las lógicas de la sociedad interconectada donde los métodos de producción de software cambiaron radicalmente [4]. Debe ser evidente que estas inconsistencias más que revelar un error, exhiben al software como una materia de estudio altamente opaca, que requiere la interacción de varias aproximaciones que se crucen y dialoguen entre sí. Como corolario queda que este proyecto busca una contribución que tome como ventaja la capacidad de analizar los hechos en perspectiva (algo que por lo reciente de los mismos estudios en los años noventa no pudieron hacer) y el asentamiento también de la teoría de medios que ya ha considerado al software como un medio más, incluso un meta-medio dotado de características de imitación y reproducción de los demás [2]. Por ello, los tres componentes metodológicos principales son:

\* Historia de la tecnología: la historia de los computadores y del software es estudiada tomando como partida la concepción misma del computador digital [14], el posterior surgimiento del software tal como lo conocemos hoy en día, y los hitos que han representado cambios tecnológicos específicos. Como ventana de tiempo puede considerarse el periodo que va desde 1945 al 2005, limitado aproximadamente por la aparición del computador de programa almacenado y la generalización de las prácticas de producción de software distribuidas por Internet. Más que ofrecer un recuento cronológico de eventos, el objetivo es armar un escenario de contextos e interacciones relevantes para el software y sus prácticas colaborativas. El análisis de documentos históricos (manifiestos tecnológicos, relatos de vida de pioneros de varias prácticas de software, el análisis de algunas fuerzas comerciales y la evolución misma del software como objeto de estudio) son ingredientes esenciales de este enfoque.

\* Ingeniería de software: esta disciplina hasta el momento es la más reconocida desde el campo de la ingeniería en la búsqueda de formalizar el proceso de producción de software como algo medible, estimable y controlable. Y aunque, como lo ha señalado incluso la literatura más estándar del tema, dicho propósito ha sido siempre esquivo [15], los avances de la ingeniería de software ofrecen un buen contrapunto para asir los recuentos históricos desde un punto más técnico.

Como se ha mencionado, de particular interés es la evolución de metodologías de producción de software y como estas han implementado artefactos, mecanismos de comunicación, canales de distribución, planificación y división de actividades, entre otras [16]. Complementando el punto anterior, existe una historia de la ingeniería de software que ofrece varios elementos de interés para interpretar el objeto de estudio desde esta perspectiva tecnológica.

\* Estudio de caso: aunque este proyecto se ubica en la intersección entre la tecnología y las humanidades, nada más lejano del objetivo fundamental del mismo que intentar esbozar una teoría general o absoluta de la colaboración en el software y su impacto en otros campos. No obstante, se hace necesario, para comprender el funcionamiento de las llamadas metáforas de colaboración y su desenvolvimiento en un campo ajeno a su origen inicial (el software), la selección de un ejemplo para observar su comportamiento. Por esta razón se ha escogido como estudio de caso la llamada “cultura maker” [17], la cual propone espacios de colaboración para el prototipado y fabricación de objetos físicos por medio de instrumentos tales como cortadoras láser, impresoras 3D, entre otros. Dicho estudio de caso ofrece dos grandes ventajas: el tratamiento que los medios masivos hacen de la cultura maker está rodeado de un halo similar al del Floss hace unos años y el que el aspecto físico de la fabricación parecería ir en contra de la supuesta inmaterialidad del software, una de las narrativas principales que se piensa abordar. De igual manera, y de modo muy conveniente, el discurso de la colaboración hace parte fundamental de los cimientos con los cuales se está construyendo este movimiento. Así, la cultura maker es un ejemplo con cualidades muy específicas para observar el traslado de las metáforas de la colaboración y abordar la inestabilidad del software, aún en soportes supuestamente “no inestables” como lo tangible o físico.

\*Técnicas etnográficas: el estudio de caso será explorado mediante técnicas de campo provenientes de la etnografía. Este enfoque ha sido ampliamente usado tanto para la caracterización de comunidades de hackers y programadores, como para enunciar requerimientos en ingeniería de software. Además, incluye herramientas como los cuadernos de campo, las entrevistas y, sobre todo, la observación participativa donde los límites entre objeto de estudio y observador se difuminan al convertirse el investigador en parte activa de una comunidad. Por ello, el uso de técnicas participativas se ha dividido en tres niveles: en el primero, una encuesta general sobre la colaboración en el movimiento maker será enviada a varios de los grupos de acción que se sientan identificados o se listen dentro de estas etiquetas que generalmente se asocian al mismo (como fablab, makerspace, hackerspace, etc.). El segundo nivel consta de entrevistas semiestructuradas con actores claves de este medio. Finalmente, un trabajo de campo donde se hará parte activa de uno de estos espacios permitirá no solo participar y colaborar *in situ* y desde adentro, sino también experimentar personalmente confirmaciones o posibles contradicciones con lo expresado en los dos niveles anteriores.

## IV. RESULTADOS ESPERADOS

### A. Resultados parciales

A pesar de su estado aún incipiente, importantes reflexiones han sido obtenidas tras la organización bibliográfica inicial previa al desarrollo propio del trabajo. Estas, como ya se ha señalado, no solo pueden tomarse como pequeños resultados o descubrimientos, sino que serán vectores transversales para la explicación de todo el trabajo.

Lo primero que llama la atención es la constante aparición de la dicotomía entre material e inmaterial del software [4], [8], [3]. A pesar de que el origen del término quería mostrar algo diferente de la maquinaria electrónica referida por la palabra “hardware”, la oposición no era manifiesta. El surgimiento propio de la palabra en los años cincuenta denota “software” como el conjunto de instrucciones para ejecutar tareas que empleaban el hardware disponible, formando un nivel de abstracción más maleable que el primero [14]. No obstante, por muchos años la industria del software estuvo supeditada al hardware como un elemento adicional que fomentaba las ventas de equipos, que era donde se suponía radicaba el negocio [9]. Independientemente de sus orígenes y de que la situación haya dado un giro completo (hoy en día es el software el motor económico muy por encima del hardware) persiste la percepción extendida de considerar al software como algo “inmaterial”, como lo reflejan términos que hacen referencia, por ejemplo, al trabajo inmaterial globalizado de los programadores, o al mundo “virtual” de las redes sociales. De esta manera, se configura la creencia sobre la cual se construyen dos grupos opuestos y excluyentes de lo material-real-físico y lo inmaterial-virtual-no tangible, y en este último grupo se ubica el software. Sin embargo, tanto la falsedad como la inutilidad de dicha dicotomía ha sido denunciada por diversas investigaciones de la disciplina de los estudios de software, e incluye rigurosas argumentaciones como la de la teórica Federica Frabetti quien, al tomar elementos de la lingüística (al considerar el código como un lenguaje per sé) y de la teoría de la deconstrucción de Derrida, logra una juiciosa argumentación del código y el software como algo bastante material [4]. Justamente, es a partir de la deconstrucción que el software se presenta como un elemento inestable, esto es, a pesar de estar basado en una tecnología supuestamente determinista como la lógica binaria de los computadores, está sujeto a incompatibilidades, errores, promesas sociales fallidas o muchos otros elementos que preciso lo hacen interesante de estudiar. En este contexto la colaboración se presenta como otro elemento inestable, con sus ventajas al oxigenar los modelos de desarrollo de software, pero con las problemáticas inherentes a todo proceso colaborativo en la red [18].

Como ya se mencionó, la inestabilidad conforma otro de los ejes rectores del trabajo. La historia tanto del software como de la ingeniería de software, constan de suficiente evidencia en la que se puede apreciar no solo el carácter oscuro del software como objeto de estudio, sino también las problemáticas que lo rodean. La misma disciplina de la

ingeniería de software como empresa formalizadora de una práctica que era considerada desordenada y más parecida al arte, padece lo que se conoce como la “crisis del software” [19], que se presenta en la constante incapacidad de producir software libre de errores, bajo los costos y tiempos trazados. Siguiendo esta premisa son varios los autores que presentan casos de estudio particulares de objetivos nunca completamente realizados en la historia del software. Por mencionar algunos casos, están las fallas en la proposición de estándares (como el caso del lenguaje Cobol) [9]; de proveer productos de uso universal independientes de la plataforma (Java) [8]; o la promesa de la llegada masiva del sistema operativo GNU/Linux al usuario regular [20]. La importancia de esta inestabilidad se observa a su vez de manera indirecta al considerar otras disciplinas que pueden haber sido influenciadas por las metáforas de la colaboración y lo abierto y que por sí mismas no han podido plantear esquemas completamente claros. Uno de los casos más patentes es el de los contenidos culturales y la ley de derechos de autor, pero el proyecto se enfocará en el estudio de caso de la cultura maker, con el fin de exhibir simultáneamente posibles inestabilidades de la fabricación y aportar a la valoración del software y la cultura que lo rodea como algo material, y no solamente a través del carácter agencial del código (es decir, que es a la vez algo que se escribe, pero también que se ejecuta), sino de posibles grietas que salgan a la luz en la promesa de una fabricación descentralizada y moldeada por patrones de comportamiento y producción presentes previamente en los procesos de colaboración del software.

### *B. Estudio de caso: la cultura maker*

Como se ha dicho, parte de la motivación del trabajo surgió de un interés, tanto teórico como activista, en la cultura de la colaboración en el software extrapolándola históricamente en el tiempo y culturalmente a otras esferas del conocimiento. Para ello se hacía necesario seleccionar un estudio de caso que no solo tuviese un carácter similar al software y los procesos colaborativos que se deseaban abordar, sino que también conservara el carácter muchas veces activista y político con el cual, especialmente en el Floss, es entendida dicha colaboración. Por tal motivo la cultura maker se constituyó en una elección idónea, tanto por el reto teórico de discutir patrones surgidos en un entorno digital supuestamente inmaterial y abordarlos en la materialidad (y con ello borrar los límites entre esta distinción), como por la decidida naturaleza activista de sus discursos, que promueven muchas veces una supuesta tecnoutopía por medio de la fabricación personal distribuida.

Hasta el momento, dos importantes consecuencias se desprenden específicamente de los textos consultados. En primer lugar, se encuentra un cambio en el paradigma económico que se atribuye a la cultura maker. La disposición de modelos de elementos en la red y el acceso creciente a dispositivos caseros de fabricación, convertirían el diseño, producción y distribución de bienes de consumo en una experiencia localizada pero interconectada que revolucionaría

la sociedad como la conocemos. La personalización de los bienes de consumo, la reducción de la contaminación por la eliminación de procesos de transporte y el diseño colaborativo son tendencias que se asocian a este resultado.

En segundo lugar, que incluye quizás el aspecto más relevante para este trabajo y es la consolidación de espacios de interacción de la cultura maker como los llamados makerspaces y fablabs como sitios de promoción de procesos educativos para todas las edades [17]. Dichos espacios son el punto de entrada para una inmersión en la cultura digital con la ventaja adicional de permitir la elaboración de objetos [21]. Existen múltiples experiencias de metodologías por proyectos donde queda demostrado que en el proceso de modelar y construir un objeto el participante no solo se empapa de tecnologías de punta, sino que las humaniza, las vuelve cotidianas y les conecta un significado de utilidad que sobrepasa la esfera del software. Aunque se puede estar de acuerdo con juicios entusiastas como el del escritor de ciencia ficción Cory Doctorow que dice ver las prácticas del Floss “materializarse” en la cultura maker (con lo que se vuelve a la dicotomía), una evaluación más juiciosa y organizada se hace necesaria. Lo importante, sin embargo, es que la cultura maker se percibe como un nuevo espacio de colaboración que hace un puente entre lo exclusivamente digital y simbólico y lo cotidiano y objetual.

## V. CONCLUSIONES Y TRABAJO FUTURO

Una de las conclusiones principales es la dificultad de emprender un proyecto como este en el campo académico. Su ubicación a medio camino entre las humanidades y la tecnología propone grandes retos. Sin embargo, es aquí donde reside la posible innovación teórica que depende del éxito de lograr articular discursos disímiles en su origen e intención, pero que al abordar un mismo objeto de estudio, presentan pequeños puntos de conexión que deben ser encontrados y explotados. Adicionalmente, esta es una consecuencia esperada dada la transversalidad actual del software, el cual despierta el interés tanto de múltiples disciplinas como la aparición de sus respectivas lecturas específicas y en sus códigos sobre este fenómeno de nuestra sociedad. Como se mencionó en la introducción, el área de los estudios de software busca ofrecer un enfoque mancomunado que albergue todas estas miradas. Una afirmación de los estudios de software que no se ha señalado y que aporta una ventaja de procedimiento y un apoyo para una posible argumentación de la colaboración en el software como elemento beneficioso para el desarrollo del mismo, reside en la observación de las ventajas del Floss para el estudio teórico. A diferencia del llamado software comercial o privativo, cuya operación interna es un secreto comercial que solo puede estudiarse a partir de sus efectos, el software libre, o de código abierto, cuenta con la entera disponibilidad tanto de su código fuente como de su documentación asociada, confiriéndole una indudable ventaja para análisis y estudios tanto técnicos como más interdisciplinarios, como este proyecto. Finalmente, a pesar del rechazo a la dialéctica material/inmaterial del

software, es indiscutible que existe una diferencia con el hardware. Esto se muestra en narrativas bastantes difundidas que manifiestan la brecha creciente entre el software y el hardware; donde el primero aún padece los mismos problemas desde que empezó a masificarse en los años sesenta (como ya se mencionó, estas problemáticas yacen en el origen de la ingeniería de software), mientras que el segundo avanza a pasos agigantados de manera más o menos ajustada a la famosa previsión de Gordon Moore en su conocida “Ley de Moore” [11] (la miniaturización de los microprocesadores es muestra fehaciente). Un posible alivio a la dicotomía es el manejo de los términos “abstracto” y “concreto”, no como conceptos opuestos, sino como distintas etapas de un mismo flujo del ámbito simbólico que construyen de manera simbiótica lo que conocemos como realidad. Esta solución entra en plena conformidad con las teorías del informático y pionero del arte digital Frieder Nake, quien a partir de sus ejercicios semióticos divide las interacciones entre superficie (lo visible) y lo interno (los símbolos manipulados por un computador), partes articuladas de una misma realidad [17].

Para concluir este recuento, se indican posibles trabajos futuros. Más que completar la redacción del proyecto tras la culminación de las labores aquí superficialmente descritas, existen algunas actividades que superan el alcance del trabajo de investigación pero que se perciben como de vital importancia. En primer lugar, el proyecto busca contribuir al fortalecimiento de las disciplinas académicas que giran entorno a la teoría de medios en Colombia. Sin bien esta es todavía incipiente en el país, existe una creciente tendencia a articular varios discursos que giran alrededor de las interacciones entre el arte, la ciencia, la tecnología y la cultura desde una perspectiva local que dialogue con los escenarios internacionales pero no pierda de vista nuestro contexto local. En esta dirección, no sobra repetir la idoneidad del Floss y de las teorías sobre la colaboración, los estándares abiertos y libres que, ya sea desde enfoques teóricos o prácticos, ofrecen una gran oportunidad para generar conocimiento de una manera transparente, horizontal y soberana, adaptándose a las condiciones locales. En segundo lugar, y como una exploración particular del proceder que se acaba de describir, estaría el estudio de expresiones autóctonas que se equiparen o sean similares a los objetos de estudio mencionados, específicamente el caso de la cultura maker. De manera más sencilla, la presencia de términos como “maker”, “hackerspace”, “fablab”, entre otros, aun cuando representan fenómenos globales, también revelan el carácter anglosajón y eurocéntrico de las prácticas y teorías relacionadas. Esto puede servir de base para establecer relaciones con prácticas locales que, como la reparación o la cultura del “cacharreo”, podrían encarnar expresiones vernáculas donde se podrían revisar las metáforas de la colaboración.

#### AGRADECIMIENTOS

El autor desea agradecer al Dr. Prof. Georg Trogemann, tutor del trabajo y al programa de doctorado en artes y medios en la Universidad de Artes y Medios de Colonia Alemania

(Kunsthochschule für Medien - Köln), donde realiza actualmente el mismo.

#### REFERENCIAS

- [1] R. Kitchin, M. Dodge. *Code/Space. Software in everyday life*. Cambridge, EUA: MIT Press, 2011.
- [2] L. Manovich. *Software takes command: extending the language of new media*. Nueva York: Bloomsbury, 2013.
- [3] M. Fuller (2008). *Introduction*. In: Fuller, M. (Ed.), *Software Studies. A lexicon*, The MIT Press.
- [4] F. Frabetti. *Software Theory. A cultural and philosophical study*. Londres: Rowman & littlefield, 2015.
- [5] D. M. Berry. *The Philosophy of Software. Code and Mediation in the Digital Age*. Londres: Palgrave Macmillan, 2011.
- [6] D.M. Berry. *Copy, rip, burn. The politics of copyleft and Open Source Software*. Londres: Pluto Press, 2008 .
- [7] M. T. Schäfer. *Bastard Culture! How User Participation Transforms Cultural Production*. Amsterdam: Amsterdam University Press, 2011.
- [8] A. Mackenzie. *Cutting Code. Software and sociality*. New York: Peter Lang Publishing Inc., 2006.
- [9] M. Campbell-Kelly. *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. Cambridge, EUA: MIT Press, 2004.
- [10] N. C. Taubert, 2006. *Produktive Anarchie?*. Bielefeld, Alemania: Transcript, 2006.
- [11] P. E. Ceruzzi. *Computing. A concise history*. Cambridge, EUA: The MIT Press, 2012.
- [12] S. Weber. *The success of open source*. Cambridge, EUA: Harvard University Press, 2004.
- [13] G. Moody. *Rebel code. Inside linux and the open source revolution*. Nueva York: Basic Books, 2002.
- [14] M. R. Williams. *A History of Computing Technology*. Nueva Jersey, EUA: Wiley-IEEE Computer Society, 1997 .
- [15] N. Wirth, “A Brief History of Software Engineering”, *Annals of the History of Computing-IEEE*, Vol. 30, pp. 32-39, Julio-2008.
- [16] I. Mistrík , J. Grundy, Van der Hoek A., Whitehead J. (Editores). *Collaborative Software Engineering*. Berlín: Springer Verlag, 2010.
- [17] Walter-Hermann J. Büchling C (Editores). *FabLab of machines, makers and inventors*. Berlín: Transcript, 2013.
- [18] J. Whitehead ,”Collaboration in Software Engineering: A Roadmap”, en *Future of Software Engineering FOSE '07*, Minneapolis, 2007, pp. 214-225.
- [19] R. Gehl, R. .S. Bell (2012; Septiembre). Heterogeneous Software Engineering: Garmisch 1968, Microsoft Vista, and a Methodology for Software Studies, *Computational Culture* (web), Volumen 2, Disponible en: <http://computationalculture.net/article/heterogeneous-software-engineering-garmisch-1968-microsoft-vista-and-a-methodology-for-software-studies> .
- [20] V. Grassmuck. *Freie Software. Zwischen Privat - und Gemeineigentum*. Bonn: Bundeszentrale für politische Bildung, 2004 .
- [21] J. Adenauer, J. Petruschat. *Prototype! physical, virtual, hybrid, smart. Tackling new challenges in design & engineering*. Berlín: Form+zweck, 2012.