

Tendencias en el desarrollo de software libre: un análisis bibliométrico

Alejandro Valencia-Arias, PhD¹, Sebastian Cardona-Acevedo², Andrés Felipe Rúa-Ortiz, Mg³, Christian Obando Ibarra, Mg¹, Anderson Trespalacios, Mg¹, Camila Bermeo-Giraldo, Mg⁴, and Hernan Cuervo Colorado, Mg³

¹Corporación Universitaria Americana, Colombia, jvalencia@americana.edu.co, cobando@americana.edu.co, amtrespalacio@americana.edu.co

²Instituto Tecnológico Metropolitano, Colombia, sebastiancardona272247@correo.itm.edu.co

³I.U Pascual Bravo, Colombia, andres.rua6757@myucw.ca, h.cuervo3128@pascualbravo.edu.co

⁴Institución Universitaria Escolme, Colombia, cies2@escolme.edu.co

Abstract– The development of free software includes information processing that generalizes variables according to the interaction and needs that they pose in a given area; it should also be noted that data visualization and interactions with them are the first candidates to evaluate the evolution and trends of a population; likewise, data analysis, data processing, data architecture, data protection and data management systematizes a set of information that communicates and connects the most representative variables of a population: trajectories, environment, traffic, speeds, technologies, synchronization of elements, among others.

Keyword: *free software, bibliometric analysis, trends*

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LEIRD2021.1.1.13>
ISSN: 2414-6390 ISBN: 978-958-52071-9-6

Tendencias en el desarrollo de software libre: un análisis bibliométrico

Alejandro Valencia-Arias, PhD¹, Sebastian Cardona-Acevedo², Andrés Felipe Rúa-Ortiz, Mg³, Christian Obando Ibarra, Mg¹, Anderson Trespalacios, Mg¹, Camila Bermeo-Giraldo, Mg⁴, and Hernan Cuervo Colorado, Mg³

¹Corporación Universitaria Americana, Colombia, jvalencia@americana.edu.co, cobando@americana.edu.co, amtrespalacio@americana.edu.co

²Instituto Tecnológico Metropolitano, Colombia, sebastiancardona272247@correo.itm.edu.co

³I.U Pascual Bravo, Colombia, andres.rua6757@myucw.ca, h.cuervo3128@pascualbravo.edu.co

⁴Institución Universitaria Escolme, Colombia, cics2@escolme.edu.co

Abstract– The development of free software includes information processing that generalizes variables according to the interaction and needs that they pose in a given area; it should also be noted that data visualization and interactions with them are the first candidates to evaluate the evolution and trends of a population; likewise, data analysis, data processing, data architecture, data protection and data management systematizes a set of information that communicates and connects the most representative variables of a population: trajectories, environment, traffic, speeds, technologies, synchronization of elements, among others.

Keyword: free software, bibliometric analysis, trends

I. INTRODUCCIÓN

El desarrollo de software libre permite no solo pensar en nuevas formas de crear códigos ejecutables, sino también, de innovar en las diferentes áreas del conocimiento donde los programadores aportan desde la ciencia soluciones prácticas a los procesos no automatizados y la accesibilidad de retos que podrían cambiar significativamente la forma de ejecutar variables en entornos empresariales, por lo que Wu (2020) define el código abierto como un beneficio para mejorar sustancialmente los proyectos asociados a los recursos empresariales.

Por esta razón, el software libre genera un ambiente de libertad sin restricciones en cuanto a la creación de nuevos aplicativos, resaltando variables importantes donde los creadores renuncian a los derechos de autor, se exponen a modificaciones y el código inicialmente desarrollado podría presentar características denominada “códigos maliciosos” en las versiones posteriores, es así como Cao (2019) presenta un proceso de verificación en los modelos construidos en los software libres, especificando la propiedad del aplicativo, el uso de la herramienta y los algoritmos de verificación que garantizan el control de los códigos creados.

En este sentido, el desarrollo de códigos abiertos impacta en campo de la innovación, dado que las necesidades de las organizaciones están concentradas en los ecosistemas cerrados y esto hace que los progresos tengan velocidades bajas en relación con la creación de nuevos conocimientos, también, es relevante mencionar la importancia que se tiene frente a la participación de personas que contribuyen con el desarrollo de códigos abiertos, lo que genera nuevas ideas, nuevos caminos para la solución de determinados problemas.

Existen cada vez más opciones de desarrollo de software libre que abordan temas de áreas científicas, como es el caso de MatLab y Python para cuantificar variables numéricas de fenómenos naturales que se asocian a comportamientos análogos, además, de permitir estructurar componentes a gran escala para estudiar fenómenos de tipo macro y micro, por lo que las publicaciones en el tema han venido especializándose y aumentando significativamente alrededor del mundo, el siguiente artículo delimita un análisis bibliométrico que da razón sobre los indicadores de calidad, cantidad y estructura decómo es el comportamiento del desarrollo de código abierto por medio de una revisión literaria basado en la construcción de una ecuación de búsqueda en Scopus.

II. MARCO TEÓRICO

En los últimos años (alrededor de la década de los 90) la cifra de dispositivos en las redes domésticas e industriales ha aumentado significativamente impactando el uso y la aplicabilidad en el internet de las cosas (IoT). Aun no es clara la cantidad de dispositivos que represente un diagnóstico claro con relación a la evolución del mundo frente a este tipo de tecnologías y el acelerado desarrollo de códigos y ejecutables que realizan una tarea determinada para un ambiente no industrializado (Bremner-Barr, Levy, & Yakhini, 2019).

Dada esta necesidad, los proyectos de desarrollo de software a partir de nuevos códigos (abiertos) se convierten en una tendencia para explorar nuevas fronteras en la construcción de metodologías que permiten ampliamente profundizar en proyectos de innovación que impacten de manera positiva las insuficiencias de las organizaciones en contextos nacionales e internacionales, así mismo, explorar y explotar el potencial de los desarrolladores (programadores) en las nuevas y mejoradas plataformas para adaptar los softwares en todas las áreas del conocimiento (Lee, Baek, & Oh, 2019).

Tal adaptación de software debe realizar procesos de priorización, en este caso de recursos informáticos (Craft y Leake, 2002), razón por la cual es indispensable mencionar que existen principios como la Ley de Pareto, que refiere que la mayoría de las consecuencias, son producidas por una minoría de causas (Juran, 1975), en una relación de 80%-20% respectivamente (Dunford, Su y Tamang, 2014).

En esta misma línea, Linåker (2020) plantea la importancia

de que las empresas contextualicen un ecosistema de desarrollos de software de característica multifacética, permitiendo no solo a la entidad ser innovadora, sino también, competitiva en el medio, que se adopten medidas para potencializar el uso, los beneficios y las consecuencias, además, que proyecte la importancia de nuevos desarrollos permitiendo al interior de la organización crear nuevos proyectos con miras en la estructuración de procesos analíticos que interactúen a través de códigos debidamente creados, compilados y ejecutados.

En este contexto, desde el punto de vista del software, el desarrollo de múltiples aplicaciones difiere de técnicas y características que están asociadas a los entornos de los lenguajes de programación. En este sentido, se hace necesario el término “código abierto” para iniciar una ruta de las diferentes aplicaciones que se desarrollan para un fin específico, por lo que es relevante contextualizar sobre los lenguajes primarios de programación de código abierto, que son utilizados de manera constante en el mundo para crear los repositorios, los códigos y los ejecutables en proyectos y aplicaciones usados en las industrias, y universidades (Corno, De Russis, & Saenz, 2020).

Así mismo, la figura 1 representa los principales lenguajes de programación en el desarrollo de aplicaciones, aquí se observa que los lenguajes más populares son JavaScript con tendencias altas en el desarrollo de páginas web con base en HTML y CSS, por otro lado Python que es un lenguaje basado en “data science” pero se ha expandido el desarrollo del mismo hasta alcanzar aplicaciones web y aplicaciones en la nube con tendencias cuantitativas; también se encuentra el lenguaje JAVA que es popular en múltiples aplicaciones, sobre todo en los móviles, la industria y orientado a objetos. Otro lenguaje es el C++ que fue desarrollado para crear objetos y tiene la característica de ser híbrido, es decir, adopta todo lo del lenguaje predecesor C y el lenguaje C que es a base de códigos con ambientes desarrollados a base de librerías y tiene un gran campo de aplicabilidad, entre ellos desarrollos en bases de datos y ambientes electrónicos (Odun-Ayo, Falade, & Samuel, 2018).

servicios basados en múltiples plataformas desarrolladas en diferentes lenguajes, además, de proveer información que es relevante tanto para las empresas como para las personas por lo que el almacenamiento masivo y los recursos informáticos se convierten en códigos fuentes que se adaptan a las necesidades humanas y pueden mejorarse y adaptarse en línea según sea la necesidad. Por otro lado, Slyadnev (2019) expone la importancia de describir los códigos fuentes para resolver problemas de ingeniería inversa, donde se abre una discusión de las principales características del software y los desarrollos posteriores en cada una de las etapas de los proyectos.

Khan (2019) incorpora ideas científicas con perfil tecnológico sobre las ventajas que se tiene frente al desarrollo de software libre a partir de lenguajes de programación debidamente estructurados, lo que permite integrar proyectos de ingeniería en las empresas con perfiles innovadores, sin embargo, el autor también hace hincapié a las desventajas que se tienen respecto al uso de OSS, y se focaliza en las vulnerabilidades de seguridad resaltando que no todos los desarrolladores de código abierto tienen la destreza de crear plataformas que imposibiliten a los hackers en alterar o violentar el proceso desarrollado.

Agregando a lo anterior, Mujica (2017) expone el aumento de aplicaciones tecnológicas en el mundo, principalmente, en el área de la salud y resalta los beneficios que se tienen frente a los desarrollos tecnológicos en la medicina permitiendo entre otras cosas visualizar, procesar, revisar, analizar e intercambiar información de tipo digital para diagnosticar procesos relacionados con el estado de los pacientes, pero marca una fuerte tendencia en la importancia de evolucionar con nuevos desarrollos recalando que una de las desventajas son los pobres desarrollos de códigos abiertos, puesto que los programas que maneja el área de la salud a través de empresas de software tienen elevados costos lo que hace poco asequible los beneficios de la salud para muchas poblaciones.

Ahora bien, el desarrollo del software libre se caracteriza por ser abierto donde cualquier ente pueda leerlo, analizarlo, modificarlo, mejorarlo y compartirlo a través de licenciamientos como Apache, GNU, Mozilla y Eclipse generando calidad, pruebas por los pares (ingenieros principalmente) del código, además, la verificación de la operabilidad y rendimiento en relación con un procedimiento delimitando las ventajas y desventajas que se tienen frente al código cerrado que certifican calidad y control en los sistemas operativos (Bahamdain, 2015).

En este mismo sentido, se tienen desarrollos similares en el tema de energías como lo expone Da Silva (2019) en un sistema fotovoltaico que requiere de software especializado para contribuir exponencialmente a sistemas que proporcionan energía, sin embargo, recalca que no se tiene disponibilidad de software para este tipo de eventos, por lo que se emplean desarrollos libres que permitan abordar estos problemas y presentan una solución a una comunidad, que para el caso, es de primera necesidad; el desarrollo del sistema se basa en MatLab software que tiene algunos licenciamientos gratis online y es de utilidad para desarrollos en el área de ingeniería, como es el uso de los scripts, Simulink y terminales

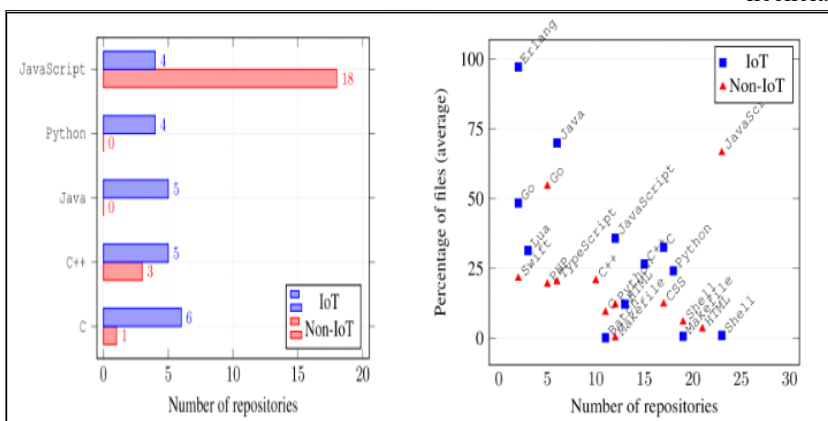


Figura 1. Lenguajes de programación en el Desarrollo de código abierto Fuente: tomado de (Corno et al., 2020)

En esa misma línea, Odun (2018) plantea que la computación en la nube es un paradigma que ofrece múltiples

de control para el manejo de sistemas energéticos en línea con velocidades ideales para la intervención de cada una de las variables.

Hobbs (2020) resalta la capacidad de desarrollar por medio de aplicaciones de software libre procesos de bajos costos, como es el caso de los sensores que devuelven datos relacionados con componentes sociales en el procesamiento de las mediciones ambientales, la ciencia ciudadana y la educación con instrumentos prácticos y de uso común como los celulares, la figura 2 muestra la construcción de un RPi Spectrometer bajoun código abierto e identificar propiedades de la luz para la identificación de materiales (1. RPi, 2. Espectrómetro, 3. Sensores para monitorear, 4. Raspberry, 5. Bateríaslocalizadas).

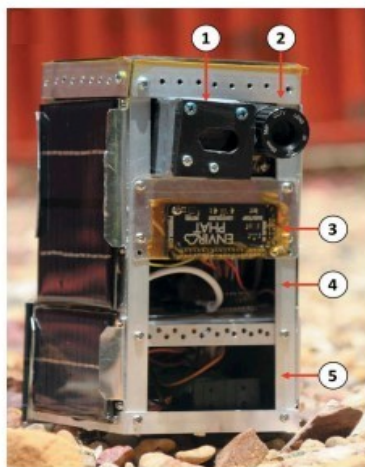


Figura 2. Overview of the cubesat and spectrometers
Fuente: tomado de (Hobbs et al., 2020)

Finalmente, Aleixo (2020) establece la importancia de recuperar objetos mediante la fotografía y explica mediante información cuantitativa la construcción de herramientas que permitan contextualizar procesamiento de datos en 2D y 3D adaptables en aplicaciones específicas, es decir, presenta una idea clara de la creación de software de código abierto (recomienda JAVA para la optimización de flujos de datos) y poder tener una adecuada calibración basadas en las características de las cámaras, la figura 3 relaciona la creación de lo propuesto por el autor.

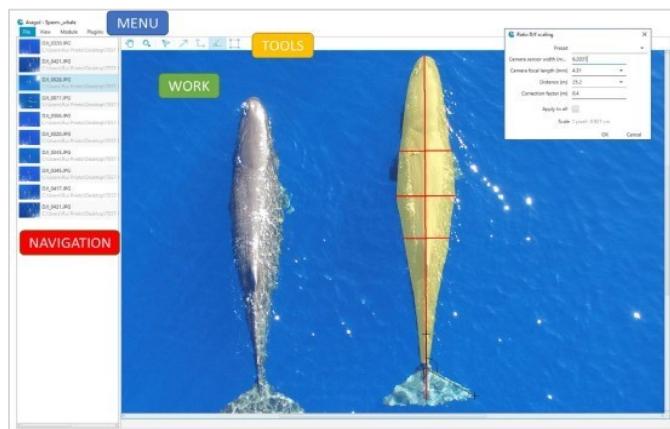


Figura 3. Graphical user interface of Arago
Fuente: tomado de (Aleixo et al., 2020)

III. METODOLOGÍA

El presente estudio en tendencias de desarrollo de software se caracteriza por ser el resultado de un análisis bibliométrico basado en un mapeo tecnológico a partir de una ecuación de búsqueda en la base de datos de Scopus, en la cual se identificaron aspectos de alta relevancia como los indicadores de calidad, cantidad y estructura relacionados con revistas, universidades, centros de investigación y autores que apuntan al desarrollo de esta temática. Desde esta perspectiva se presentan en el campo del desarrollo del software los impactos más relevantes frente a las investigaciones que se tienen en esta área. El análisis bibliométrico como metodología relaciona la evolución y las tendencias de los desarrollos de software que se presentan en la actualidad, donde el mercado de software libre ha tenido cambios significativos en la actualización de algoritmos, códigos, y entornos de desarrollo que generan impactos a nivel industrial.

VI. RESULTADOS

4.1. INDICADORES DE CANTIDAD

La bibliometría es una herramienta por medio de la cual se analiza la literatura académica y científica, con respecto a las publicaciones o investigaciones en un campo específico (Acevedo et al., 2017). Este análisis se presenta por medio de diferentes indicadores que permiten orientar la revisión de literatura y medir sus resultados (Aleixandre et al., 2017), donde, de los indicadores más importantes, se encuentran los de cantidad, los cuales, de acuerdo con Valencia, Arévalo y Rodríguez (2017), permiten cuantificar el número de publicaciones en un tiempo determinado, trasladando el análisis a términos de productividad académica, bien sea por años, por autores, por revistas o por países (Acevedo et al., 2017).

4.1.1. INDICADORES DE CANTIDAD POR AUTORES

De acuerdo con la Figura 4, alrededor de la temática del desarrollo de software libre, el autor que, en la actualidad, posee una mayor productividad científica es Kevin Crowston, investigador de la Syracuse University, quien, con un total de 18 publicaciones, orienta sus investigaciones hacia los nuevos modelos de organización que son posibles gracias al uso de la tecnología de la información, con un interés específico en el desarrollo de software de código abierto o software libre, de hecho, su investigación más representativa en el área es un análisis del éxito de los sistemas de información en el contexto de los software libres (Crowston, Howison y Annabi, 2006).

Por otro lado, el segundo autor que en la actualidad da cuenta de un mayor número de investigaciones alrededor del desarrollo de software libres, es el Dr. Walt Scacchi, con un total de 14 estudios; la publicación más relevante de este autor, asociado al Institute for Software Research, de la University of California, describe los procesos de mejora y práctica de un software, con énfasis en aquellos de código abierto o libre - F/OSSD (Sacchi et al., 2006). A su vez, otra autora que posee una cifra homóloga de productividad es la líder investigadora de la University of Oulu, Netta Livari, profesora de Sistemas de

información, y cuya principal investigación alrededor de esta temática de estudio, informa sobre la participación de los usuarios en el contexto del desarrollo de software libres o de código abierto – OSS (Livari, 2009).



Figura 4. Autores más productivos
Fuente: Elaboración propia a partir de Scopus

En términos estadísticos, por medio de este análisis bibliométrico se identificó que el 6,5% de los autores publican el 25% de la literatura existente, el 19,9% realiza la publicación del 50% de la literatura aproximadamente, mientras que, a su vez, el 25% de los autores da cuenta del 75% de las publicaciones. Por último, teniendo que el 56,4% de estos realiza el 80% de las publicaciones, se sabe que, por tanto, no se cumple con la Ley de Pareto.

4.1.2. INDICADORES DE CANTIDAD POR REVISTAS

Con relación a los previamente definidos Indicadores de Cantidad, por medio de la Figura 5, se analizan las 5 revistas que, en la actualidad, poseen una mayor productividad académica y científica con énfasis en el desarrollo de software libre; encontrando que la revista que encabeza este listado es la revista alemana Lecture Notes in Computer Science, revista que se enfoca en la divulgación académica de los nuevos desarrollos en la investigación informática y tecnologías de la información y cuya publicación más representativa con relación al desarrollo de software libre, se titula “Agile principles and open source software development: A theoretical and empirical discussion”, que discute, de acuerdo con Koch (2004), las similitudes en problemáticas y resultados entre el desarrollo ágil y el código abierto, en el contexto de diseño de software.

En segundo lugar, se encuentra la revista IFIP International Federation for Information Processing, revista estadounidense que tiene, dentro de sus principales temáticas de interés, la libertad de acceso a la información, la cual, de acuerdo con Oszlak (2017), encierra la discusión sobre el desarrollo de software libre. La publicación que en la actualidad posee un mayor impacto, es denominada “Continuous integration in open source software development”, que enmarca o contextualiza la discusión sobre la integración continua de las metodologías ágiles en el desarrollo de software de código abierto (Deshpande y Riehle, 2008).

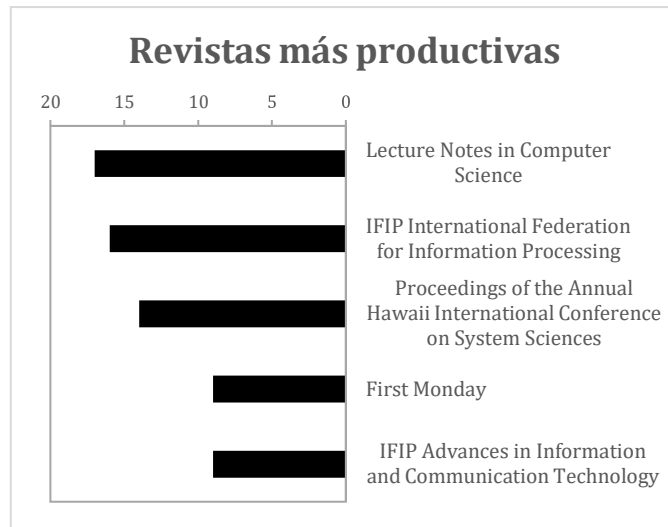


Figura 5. Revistas más productivas
Fuente: Elaboración propia a partir de Scopus

Posteriormente, se tienen las revistas Proceedings of the Annual Hawaii International Conference on System Science, First Monday y la IFIP Advances in Information and Communication Technology, con un total de 14, 9 y 9 publicaciones respectivamente. Por otro lado, con relación al análisis estadístico realizado por cuartiles, se halló que el 2,8% de las revistas realizan el 25% de publicaciones en la temática del desarrollo de software de código abierto; mientras que el 13,1% publica el 50% de las publicaciones científicas y que, a su vez, el 24,9% del total de revistas da cuenta del 75% de la producción académica. Finalmente, teniendo que el 46% de las revistas realiza el 80% del total de publicaciones e investigaciones en la temática, se sabe que, por tanto, no se cumple con la Ley de Pareto.

4.1.3. INDICADORES DE CANTIDAD POR PAÍSES

De acuerdo con la información identificada por medio de la Figura 6, los 5 países que, en la actualidad, poseen una mayor productividad científico-académica con relación a la discusión sobre las diferentes variables que componen el desarrollo del software libre o de código abierto, pertenecen al contexto de países desarrollados. En esta lista se puede identificar que Estados Unidos es, ampliamente, el país que da cuenta de un mayor número total o acumulado de investigaciones en el área, con un total de 141, donde, de esta cifra, 14 fueron publicadas entre los últimos 5 años.

Por otro lado, se puede ver que, en el contexto europeo, son los países Reino Unido (31), Finlandia (18) y Alemania (16), quienes continúan el listado de los países que dan cuenta de un mayor volumen de publicaciones alrededor de la temática central de investigación; a su vez, teniendo en cuenta la producción científica de estos tres países en conjunto, se identificó que 12 publicaciones fueron realizadas en los últimos 5 años, y teniendo en cuenta este intervalo de tiempo, la publicación más importante fue titulada “Development, deployment and applications of robot audition open source software HARK” que, realizada en el Reino Unido, buscaba explicar las principales características de un modelo de

software abierto enfocado al desarrollo de tecnologías de sonido en robótica (Nakadai, Okuno y Mizumoto, 2017).

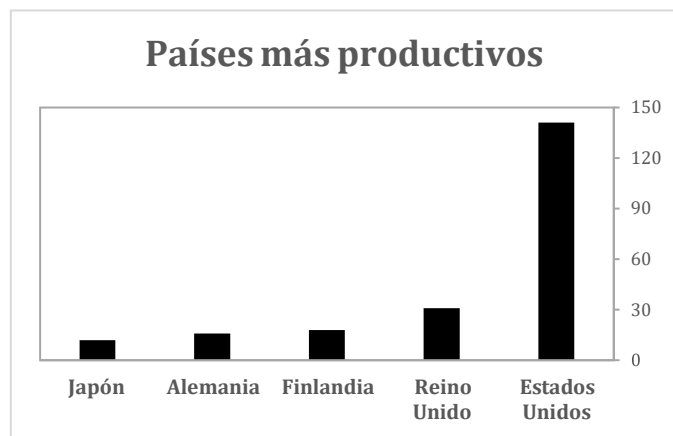


Figura 6. Países más productivos
Fuente: Elaboración propia a partir de Scopus

Por último, se identificó que, con relación al análisis estadístico de la productividad científica global, el 0,8% de los países dan cuenta del 25% de la literatura existente orientada al análisis de factores asociados al desarrollo de software libre; mientras que, por otro lado, el 2,3% del total de países da cuenta del 75% de las investigaciones existentes en el área. Además, teniendo presente que el 3,8% de los países da cuenta del 80% del total de trabajos, se tiene que no se cumple con la Ley de Pareto.

Figura 7. Top 5 autores más citados
Fuente: Elaboración propia a partir de Scopus

4.2. INDICADORES DE CALIDAD

De acuerdo con García y García (2021), los indicadores bibliométricos en general tienen un doble propósito: cuantificar la producción científica en términos de productividad (indicadores de cantidad), y evaluar su impacto dentro de la comunidad académica (indicadores de calidad). Estos últimos indicadores son una herramienta de medición de la calidad de las investigaciones (Cascón et al., 2020), y cuyo impacto puede ser cuantificado a partir del número total de citaciones (King, Llinàs y Améstica, 2020). En ese sentido, se presentan indicadores de calidad por autores, revistas y países.

4.2.1. INDICADORES DE CALIDAD POR AUTORES

La información contenida en la Figura 7, permite identificar los autores que, con relación al número total de citaciones, poseen un mayor impacto en la academia, específicamente alrededor de la temática central sobre desarrollo de software libre y sus variables conexas. En ese sentido, se puede apreciar que son los autores Audris Mockus, Roy Fielding y James D. Herbsleb quienes poseen un mayor número de citaciones en el campo, con un total de 1084. Estas citaciones corresponden a un único trabajo denominado “Two case studies of open source software development: Apache and Mozilla”, el cual, de acuerdo con los autores, se basa en un análisis comparativo entre la capacidad de obtención de éxito de un software de código abierto, contra los métodos

tradicionales de desarrollo comercial. (Mockus, Fielding y Herbsleb, 2002).

Posteriormente, en el top 5 de autores con mayor impacto o mayor número total de citaciones, se encuentran los autores David Agus y Matthew Chambers, coautores, junto con los autores Darren Kessner, Robert Burke y Parag Mallick, de una investigación que posee un total de 992 citaciones y que es titulada “ProteoWizard: Open source software for rapid proteomics tools development”, la cual tiene como objetivo, de acuerdo con Kessner et al. (2008), analizar el proyecto de software ProteoWizard, como conjunto modular y extensible de bibliotecas y herramientas multiplataforma de código abierto.

Con relación a la división del total de citaciones en cuartiles para la susceptibilidad al análisis estadístico, se obtuvo que el 0,02% de los autores da cuenta del 25% del total de citaciones en el campo; que el 0,04% de los mismos posee el 50% total de citas, mientras que, a su vez, el 0,17% de autores explica, aproximadamente, el 75% del total de citaciones. Adicionalmente, partiendo de que el 1,01% de autores justifica el 80% de citas, se sabe que, por tanto, no se cumple con la Ley de Pareto.

4.2.2. INDICADORES DE CALIDAD POR REVISTAS

Por otro lado, al analizar las 5 revistas que, en la actualidad, poseen una mayor cantidad de citaciones en investigaciones relacionadas al desarrollo de OSS, y de acuerdo con la Figura 8, se encontró que es la revista estadounidense ACM Transactions on Software Engineering and Methodology, revista que, con un total de 1084 citaciones, se encarga de publicar investigaciones sobre especificación, diseño, desarrollo y mantenimiento de software. Esta revista perteneció al cuartil Q1 en Scopus hasta el año 2015 y, en la actualidad, se encuentra en el cuartil Q2 y su impacto con relación a la temática de investigación se ve explicado por el trabajo publicado por Mockus, Fielding y Herbsleg, quienes son los autores más citados (ver Figura 7).

En segundo término, se encuentra la revista inglesa Bioinformatics, revista asociada a la Oxford University Press, y que cuenta con un total de 1015 citaciones en la actualidad. Esta revista, perteneciente al cuartil Q1 desde 1999 hasta la actualidad, se centra principalmente en la publicación de artículos académicos y científicos enfocados en los nuevos desarrollos en bioinformática del genoma y biología computacional. Sus dos publicaciones alrededor de la temática de investigación son, por una parte, el mencionado “ProteoWizard: Open source software for rapid proteomicstools development” que cuenta con un total de 992 citaciones, y por otra parte, otro artículo científico titulado “Biogem: An effective tool-based approach for scaling up open source software development in Bioinformatics” que cuenta con un total de 23 citaciones. Por último, el top 5 de revistas más citadas lo completan las revistas MIS Quarterly: Management Information Systems (785), Management Science (561) y Research Policy (507).

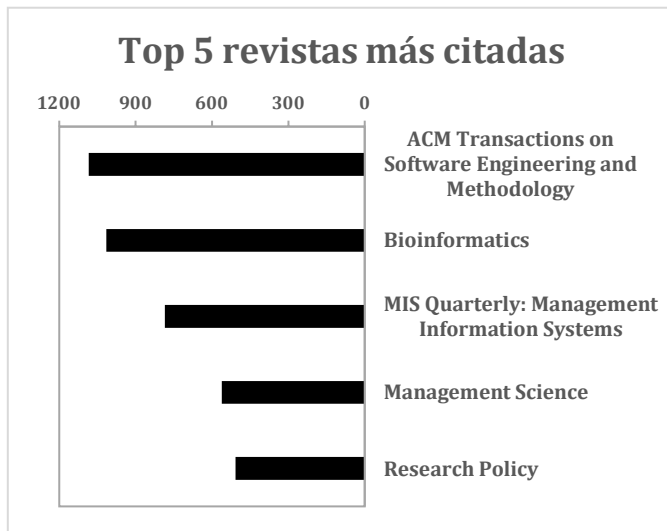


Figura 8. Top 5 revistas más citadas
Fuente: Elaboración propia a partir de Scopus

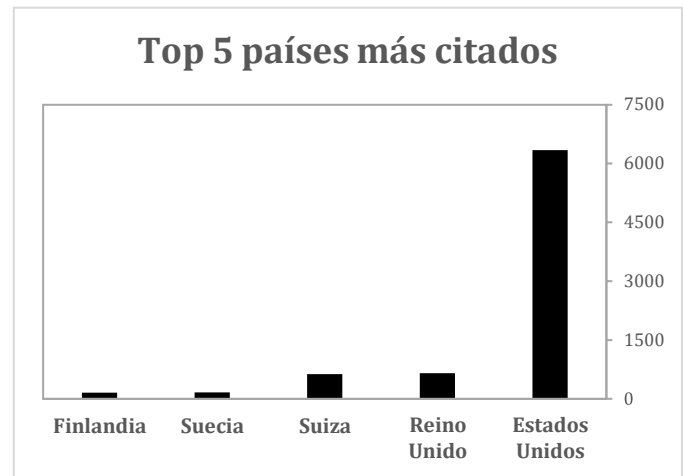


Figura 9. Top 5 países más citados
Fuente: Elaboración propia a partir de Scopus

Estadísticamente, se obtuvo que el 0,02% de las revistas posee aproximadamente el 25% del total de citaciones; mientras que el 0,05% da cuenta del 50% de citas en el área de investigación y que, a su vez, el 0,13% explica el 75% del total de citaciones. Por otro lado, partiendo de que el 0,27% de revistas da cuenta del 80% de citaciones, se observa que no se cumple con la Ley de Pareto.

4.2.3. INDICADORES DE CALIDAD POR PAÍSES

Sejante al análisis de los indicadores de cantidad por países (ver Figura 6), los indicadores de calidad por países presentan los principales resultados dentro del contexto de países desarrollados, donde, de acuerdo con la Figura 9, el país que da cuenta de un mayor número total de citaciones es Estados Unidos, con 6342 citaciones en la actualidad, contando entre sus registros con las publicaciones más importantes en términos de impacto alrededor de la temática del desarrollo de software libre o de código abierto y que han sido analizadas previamente. Seguido a Estados Unidos se encuentra el Reino Unido, contando con un número de citas de 661, y cuya principal publicación, con un total de 207 citaciones, es titulada “Code quality analysis in open source software development”, la cual expone un análisis comparativo entre el software de acceso libre, contra aquellos del modelo cerrado tradicional (Stamelos et al., 2002).

Posteriormente, se encuentran en la lista países como Suiza (633), Suecia (168) y Finlandia (161), completando el top 5 de países que dan cuenta de un mayor número acumulado de citaciones en este campo de estudio. Adicionalmente, en términos de análisis estadístico por cuartiles, se identificó que el 0,02% de países responden o explican, aproximadamente, el 75% de citaciones, es decir, no se tienen países posicionados dentro de los dos primeros cuartiles estadísticos. A su vez, se tiene que el 0,03% de los países dan cuenta del 80% de citaciones, razón por la cual no se cumple la Ley de Pareto.

4.3. INDICADORES DE ESTRUCTURA

4.3.1. COAUTORÍA DE AUTORES MÁS PRODUCTIVOS

De acuerdo con Alvarado y Arango (2018), una red de coautoría gesta cuando se da una colaboración entre dos o más autores, por medio de datos, equipos e ideas, para la publicación de un artículo académico. En ese sentido, de acuerdo con la Figura 4, se pueden identificar, por medio de 4 diferentes clústeres, la colaboración que se ha presentado entre algunos de los autores más productivos en el campo investigativo, con el principal clúster, caracterizado con el color rojo, compuesto por autores como Scacchi, quien es el segundo autor más productivo, Jensen, que es el séptimo, Noll, quien es el treceavo y Elliott, en la posición 44. El segundo clúster, de color azul, se ve compuesto por los autores Fitzgerald, quien se posiciona entre los 15 autores más productivos y el investigador Feller. Por medio del clúster verde, se encuentran los autores Robles y Capuluppi, quienes están ubicados al interior de la lista de 30 autores más productivos en la temática. Por último, mencionamos el clúster morado, compuesto por el autor Lundell B, también al interior de los 30 autores más representativos en términos de productividad académica.

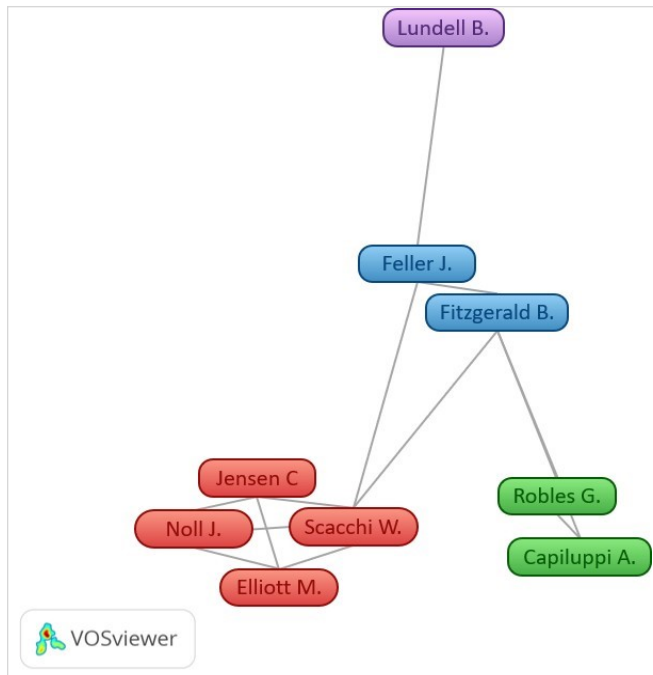


Figura 10. Red de autores más productivos
Fuente: Elaboración propia a partir de VOSviewer y Scopus

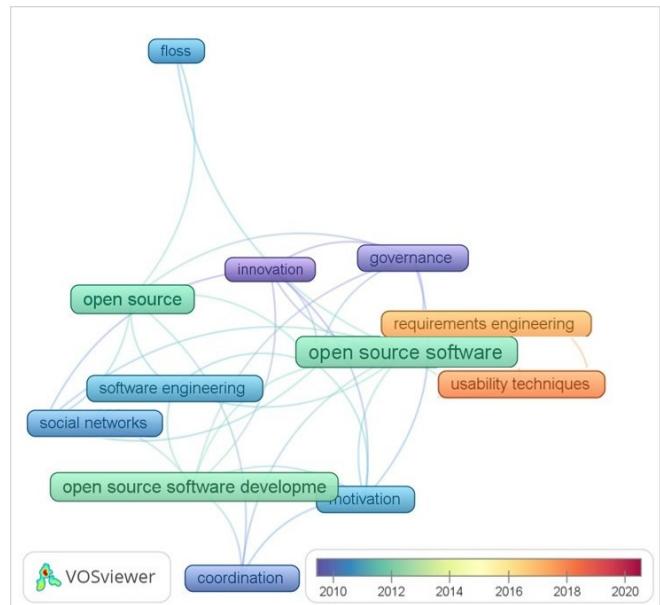


Figura 11. Red de coocurrencia de palabras clave
Fuente: Elaboración propia a partir de VOSviewer y Scopus

4.3.2. COOCURRENCIA DE PALABRAS CLAVE.

La red de coocurrencia de palabras clave es un factor clave dentro de todo análisis bibliométrico, puesto que éste, de acuerdo con Pérez et al. (2020), permite identificar los principales tópicos que responden a los propósitos de la investigación. En ese sentido, por medio de la Figura 5, se identifican las palabras clave que, de forma histórica o cronológica, se perfilan como las más representativas dentro de los estudios en desarrollo de software libre. Entre los resultados de esta red de coocurrencia, se puede ver que, entre las principales palabras claves conexas a la temática, las dos más recientes son “requirements engineering” (ingeniería de requisitos), y “usability techniques” (técnicas de usabilidad).

De acuerdo con Wang et al. (2019), la relación de la ingeniería de requisitos y el desarrollo de software de código abierto o de acceso libre, tiene que ver con el diseño e implementación de requisitos de seguridad en general, y en específico sobre la seguridad acarreada en aspectos relacionados a códigos, archivos adjuntos o enlaces con recursos externos. Finalmente, la relación de las técnicas de usabilidad y el desarrollo de OSS, se explica a partir del incremento de usuarios de este tipo de software sin ser necesariamente desarrolladores, lo cual implica limitaciones, desfavorabilidades y desventajas en cuanto al desconocimiento de las técnicas a usar en cada actividad del proceso de desarrollo, lo cual conlleva a un análisis de adaptabilidad o modificación en las técnicas de usabilidad de software (Llerena et al., 2019).

CONCLUSIONES

El desarrollo de software libre es una temática que le compete a toda la industria actual, desde el desarrollo de algoritmos de inteligencia artificial hasta el almacenamiento de información en plataformas de Big Data. Lo cual, desde un enfoque científico e investigativo, todas las empresas a nivel mundial están en la tarea de crear soluciones a partir de metodologías de innovación.

El desarrollo de software libre tiende a convertirse en un modelo industrial, dado que las empresas apuntan a este tipo de proyectos reflejando altos beneficios de acuerdo con la parametrización de modelos de innovación.

El análisis bibliométrico genera una relación directa con el aumento de conocimiento que se tiene en esta área, garantizando impactos fuertes en el desarrollo de software, además, muestra que es una temática que a pesar de ser muy popular en la sociedad tiene una alta demanda y cada día refleja nuevos y mejores desarrollos industriales con propósitos muy claros.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Acevedo, N. M., Jiménez, L. M., & Rojas, M. D. (2017). Análisis bibliométrico sobre indicadores de innovación. *Revista Espacios*, 38(08).
- [2] Alexandre, R., de Dios, J. G., Cogollos, L. C., Molina, C. N., Alonso, A., Vidal, A., & Lucas, R. (2017). Bibliometría e indicadores de actividad científica (II). *Indicadores de producción científica en pediatría. Acta Pediatr Esp*, 75(3-4), 44-50.
- [3] Aleixo, F., O'Callaghan, S. A., Ducla Soares, L., Nunes, P., & Prieto, R. (2020). AragoJ: A free, open-source software to aid single camera photogrammetry studies. *Methods in Ecology and Evolution*, 2020(January), 1–8. <https://doi.org/10.1111/2041-210X.13376>
- [4] Alvarado, R. U., & Arango, C. R. (2018). La red de co-autores en la bibliometría Mexicana. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, 23(51), 74-94.

- [5] Bahamdain, S. S. (2015). Open source software (OSS) quality assurance: A survey paper. *Procedia Computer Science*, 56(1), 459–464. <https://doi.org/10.1016/j.procs.2015.07.236>
- [6] Bremler-Barr, A., Levy, H., & Yakhini, Z. (2019). IoT or NoT: Identifying IoT Devices in a ShortTime Scale. <https://doi.org/1910.05647>
- [7] Cao, Z., Yin, J., Wang, Y., Li, Y., & Zhang, J. (2019). An Empirical Study of Open Source Flight Control Software Program Model Checking. *Proceedings - Companion of the 19th IEEE International Conference on Software Quality, Reliability and Security, QRS-C 2019*, 164–169. <https://doi.org/10.1109/QRS-C.2019.00043>
- [8] Cascón, J., Moral, J. A., Liao, H., & Cobo, M. J. (2020). Análisis bibliométrico de la Revista Española de Documentación Científica desde su inclusión en la Web of Science (2008-2018). *Revista Española de Documentación Científica*, 43(3), 267.
- [9] Corno, F., De Russis, L., & Saenz, J. P. (2020). How is Open Source Software Development Different in Popular IoT Projects? *IEEE Access*, 8, 28337–28348. <https://doi.org/10.1109/ACCESS.2020.2972364>
- [10] Craft, R. C. y Leake, C. (2002), The Pareto principle in organizational decision making. *Management Decision*, 40(8), 729–733. <https://doi.org/10.1108/00251740210437699>
- [11] Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, 11(2), 123–148.
- [12] Da Silva, J. R. C., & Pacheco, G. M. (2019). Open source software development in matlab for sizing photovoltaic systems. *2018 13th IEEE International Conference on Industry Applications, INDUSCON 2018 - Proceedings*, 256–262. <https://doi.org/10.1109/INDUSCON.2018.8627272>
- [13] Deshpande, A., & Riehle, D. (2008). Continuous integration in open source software development. In *IFIP International Conference on Open Source Systems*. 273–280.
- [14] Dunford, R., Su, Q., y Tamang, E. (2014) The Pareto Principle. *The Plymouth Student Scientist*, 7(1), 140–148.
- [15] García, C., & García, J. M. (2021). Indicadores bibliométricos para evaluar la actividad científica. *Radiología*, 63(3), 228–235.
- [16] Hobbs, S. W., Paull, D. J., Haythorpe, J., & McDougall, T. (2020). Developing a spectral pipeline using open source software and low-cost hardware for material identification. *International Journal of Remote Sensing*, 41(7), 2517–2543. <https://doi.org/10.1080/01431161.2019.1693075>
- [17] Iivari, N. (2009). “Constructing the users” in open source software development: An interpretive case study of user participation. *Information Technology & People*.
- [18] Juran, J. M. (1975). The non-Pareto principle; mea culpa. *Quality Progress*, 8(5), 8–9.
- [19] Kessner, D., Chambers, M., Burke, R., Agus, D., & Mallick, P. (2008). ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics*, 24(21), 2534–2536.
- [20] Khan, F. I., Javed, Y., & Alenezi, M. (2019). Security assessment of four open source software systems. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(2), 860–881. <https://doi.org/10.11591/ijeecs.v16.i2.pp860-881>
- [21] King, A., Llinàs, X., & Améstica, L. (2020). Caracterización de la producción científica sobre clasificaciones de universidades. *Un estudio bibliométrico desde 1988 a 2018. Formación universitaria*, 13(2), 53–62.
- [22] Koch, S. (2004, June). Agile principles and open source software development: A theoretical and empirical discussion. In *International Conference on Extreme Programming and Agile Processes in Software Engineering*. 85–93.
- [23] Lee, S., Baek, H., & Oh, S. (2019). The role of openness in open collaboration: A focus on open-source software development projects. *ETRI Journal*, 42(September 2018), 196–204. <https://doi.org/10.4218/etrij.2018-0536>
- [24] Linåker, J., Regnell, B., & Damian, D. (2020). A method for analyzing stakeholders’ influence on an open source software ecosystem’s requirements engineering process. *Requirements Engineering*, 25(1), 115–130. <https://doi.org/10.1007/s00766-019-00310-3>
- [25] Llerena, L., Rodríguez, N., Castro, J. W., & Acuña, S. T. (2019). Adapting usability techniques for application in open source software: A multiple case study. *Information and Software Technology*, 107, 48–64.
- [26] Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3), 309–346.
- [27] Mujika, K. M., Méndez, J. A. J., & De Miguel, A. F. (2017). Workstation versus free open source software in radiology. *ACM International Conference Proceeding Series*, Part F132203. <https://doi.org/10.1145/3144826.3145411>
- [28] Nakadai, K., Okuno, H. G., & Mizumoto, T. (2017). Development, deployment and applications of robot audition open source software HARK. *Journal of Robotics and Mechatronics*, 29(1), 16–25.
- [29] Odun-Ayo, I., Falade, A., & Samuel, V. (2018). Cloud computing and open source software: Issues and developments. *Lecture Notes in Engineering and Computer Science*, 2233, 140–145.
- [30] Oszlak, O. (2017). La noción de Estado abierto en el contexto de América Latina y el Caribe. In *Desde el gobierno abierto al Estado abierto en América Latina y el Caribe*. 209–229.
- [31] Pérez, L. G., Blanco, C. P., González, M. J. P., & Placeres, G. M. (2020). El movimiento iSchools en Iberoamérica: un análisis de su producción científica. *Revista Publicando*, 7(27), 10–22.
- [32] Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding free/open source software development processes.
- [33] Slyadnev, S. E., & Turlapov, V. E. (2019). To the Development of Open Source Software for the Reconstruction of CAD Models. *Programming and Computer Software*, 45(4), 202–212. <https://doi.org/10.1134/S036176881904008X>
- [34] Stamelos, I., Angelis, L., Oikonomou, A., & Bleris, G. L. (2002). Code quality analysis in open source software development. *Information systems journal*, 12(1), 43–60.
- [35] Valencia, D. C., Arévalo, J. B., & Rodríguez, D. (2017). Análisis bibliométrico sobre direccionamiento de los estudios en Riesgos Financieros. *Revista Venezolana de Gerencia*, 38(59), 2–13.
- [36] Wang, W., Mahakala, K. R., Gupta, A., Hussein, N., & Wang, Y. (2019). A linear classifier based approach for identifying security requirements in open source software development. *Journal of Industrial Information Integration*, 14, 34–40.
- [37] Wu, Y., Zhang, Y., Wang, T., & Wang, H. (2020). Characterizing the Occurrence of Dockerfile Smells in Open-Source Software: An Empirical Study. *IEEE Access*, 8, 34127–34139. <https://doi.org/10.1109/ACCESS.2020.2973750>