



Artículo de Revisión de la Literatura

Recursos Software para Implementar Sistemas de Soporte a la Decisión Multicriterio: Una Revisión Sistemática de Literatura

Software Resources for Implementing Multi-Criteria Decision Support Systems: A Systematic Literature Review

Juan Francisco Figueroa Pérez¹, Juan Carlos Leyva López², Edgar Omar Pérez Contreras¹ y Herman Geovany Ayala Zúñiga¹

¹Facultad de Ingeniería Mochis, Universidad Autónoma de Sinaloa, Los Mochis, 81210, México

²Departamento de Ciencias Económico-Administrativas, Universidad Autónoma de Occidente, Culiacán, 80020, México

juanfco.figueroa@uas.edu.mx, juan.leyva@uado.mx, edgar.perez@uas.edu.mx,
hayala@uas.edu.mx

INFORMACIÓN

Historial del Artículo

Recibido: Abril 30, 2025

Aceptado: Noviembre 19, 2025

Publicado: Diciembre 23, 2025

Palabras Clave

Revisión sistemática
de la literatura

Análisis de decisiones
multicriterio

Sistemas de apoyo a la
toma de decisiones
multicriterio

Herramientas de
desarrollo de software

Recursos de software

RESUMEN

El análisis de decisiones multicriterio es una rama de la investigación operativa que ofrece una amplia variedad de métodos para resolver problemas en diversas áreas del conocimiento. En la última década, se han desarrollado y descrito en la literatura diversos recursos de software para la programación de sistemas de soporte de decisiones multicriterio. Esta revisión busca determinar el estado del arte mediante la revisión de diferentes fuentes para obtener un inventario de estos recursos. Se realizó una revisión sistemática de literatura de investigaciones publicadas entre 2012 y 2024. De los 190 estudios relevantes encontrados, 29 cumplieron con los criterios de inclusión y exclusión y fueron seleccionados. Para analizar los artículos y el software, se realizó un análisis contextual y un análisis conceptual formal. Nuestros hallazgos muestran que los recursos de software más desarrollados fueron los marcos para dominios de problemas específicos, y los métodos multicriterio más utilizados fueron AHP y TOPSIS. Windows, Linux y Mac fueron los sistemas operativos más compatibles. Un buen número de los recursos revisados son gratuitos y proporcionan código fuente, aunque la mayoría no están orientados a su uso más allá del trabajo de investigación en el que se propusieron.

©2025. Este es un artículo de acceso abierto distribuido bajo la licencia CC BYNC 4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>)

Cómo citar:

Figueroa Pérez, J. F., Leyva López, J. C., Pérez Contreras, E. O., & Ayala Zúñiga, H. G. (2025). Recursos software para implementar sistemas de soporte a la decisión multicriterio: una revisión sistemática de literatura. *Revista Lince de Ciencias Sociales, Humanidades y Tecnologías*, 1(2), 102–137. <https://doi.org/10.63622/RLI/2025.02/05>

MANUSCRIPT INFO

Article History

Received: April 30, 2025

Accepted: November 19, 2025

Published: December 23, 2025

Keywords

Systematic literature review

Multicriteria decision analysis

Multicriteria decision support systems

Software development tools

Software resources

ABSTRACT

Multicriteria decision analysis is a branch of operations research that offers a wide variety of methods for solving problems in diverse areas of knowledge. In the last decade, various software resources for programming multicriteria decision support systems have been developed and described in the literature. This review seeks to determine the state of the art by reviewing different sources to obtain an inventory of these resources. A systematic literature review of research published between 2012 and 2024 was conducted. Of the 190 relevant studies found, 29 met the inclusion and exclusion criteria and were selected. To analyze the articles and the software, a contextual analysis and a formal conceptual analysis were performed. Our findings show that the most developed software resources were frameworks for specific problem domains, and the most used multicriteria methods were AHP and TOPSIS. Windows, Linux, and Mac were the most compatible operating systems. A good number of the reviewed resources are free and provide source code, although most are not intended for use beyond the research work in which they were proposed.

1. Introducción

El Análisis de Decisiones Multicriterio (MCDA, Multiple Criteria Decision Analysis), es una de las áreas de investigación más conocidas en la toma de decisiones, con aplicación en diversos campos (Ozsahin et al., 2021). Abarca el análisis de diversas opciones disponibles en situaciones de la vida cotidiana, la ciencia, las ciencias sociales, la ingeniería, la medicina y muchas otras (Figueroa-Perez et al., 2020; Muruganantham & Gandhi, 2019; Ozsahin et al., 2021; Sharma et al., 2018; Yang et al., 2018).

En las últimas décadas, los avances del MCDA y la informática han propiciado el desarrollo de un nuevo tipo de Sistemas de Apoyo a la Decisión (DSS, Decision Support System), conocido como Sistema de Apoyo a la Decisión Multicriterio (MDSS, Multicriteria Decision Support System), el cual constituye una potente herramienta para analizar problemas complejos y no estructurados, además de proporcionar un soporte integrado para la toma de decisiones en tiempo real.

Los MDSS son una subcategoría de DSS que presenta sus principales características, asumiendo que al menos uno de los modelos que implementan pertenece al campo del MCDA (Leyva López et al., 2008). El desarrollo de MDSS se remonta a principios de la década de 1970, cuando aparecieron sus primeras implementaciones. Estos sistemas se desarrollaron utilizando las limitadas tecnologías de la información disponibles en aquel momento. Desde entonces, se han desarrollado cinco generaciones de MDSS, que se han utilizado en problemas del mundo real aprovechando el gran progreso de las tecnologías de la información (TI) y las mejoras del MCDA (Siskos & Spyridakos, 1999). Durante este período, se han desarrollado diversas aplicaciones, desde las que tenían una interfaz de usuario basada en consola, en sus inicios, hasta las de escritorio en los últimos años. Si

bien estas aplicaciones han quedado en desuso con el tiempo, proporcionaron las bases necesarias para comprender las particularidades del proceso de desarrollo de DSS.

Por otro lado, un recurso software es cualquier conjunto coherente de artefactos o productos de trabajo que resuelven un problema o conjunto de problemas específicos que se presentan en el ciclo de vida del desarrollo de software (SDLC, Software Development Life Cycle) (Zhao et al., 2010). En este sentido, los sistemas informáticos, incluyendo los MDSS, rara vez se construyen completamente desde cero y generalmente recurren a la reutilización de elementos construidos a priori. Es aquí donde el uso y la integración de recursos de software previamente creados y probados se convierte en un factor fundamental para lograrlo.

Investigaciones recientes muestran el desarrollo de diversos MDSS en los que los investigadores utilizaron recursos software como bibliotecas, frameworks, servicios web, paquetes, kits de desarrollo de software (SDK, Software Development Kit), interfaces de programación de aplicaciones (API, Application Programming Interfaces), kits de herramientas, aplicaciones y algoritmos tanto en la dimensión teórica como en la práctica. De este modo, para avanzar en el campo del desarrollo de los Recursos de Software para la Programación de Sistemas de Soporte de Decisiones Multicriterio (SRPMDS, Software Resources for Programming Multicriteria Decision Support Systems), es útil determinar el estado actual del arte mediante la identificación, evaluación e interpretación de la investigación relevante llevada a cabo hasta la fecha relacionada con los mismos.

El objetivo de este trabajo es realizar una Revisión Sistemática de la Literatura de investigaciones que abordan específicamente los SRPMDS, con base en las directrices propuestas por Kitchenham et al. (2015). Si bien todos los recursos de software presentan ventajas y desventajas, su análisis queda fuera del alcance de este trabajo, que se centra en los métodos multicriterio implementados más importantes, los sistemas operativos compatibles, los lenguajes de desarrollo de software utilizados, la información sobre licencias y el código fuente disponible, proporcionando un inventario de estos recursos desarrollados entre 2012 y 2024 en el contexto de SRPMDS e identificando posibles brechas en la literatura para sugerir áreas de investigación futura.

El resto de este documento se organiza de la siguiente manera: la [Sección 2](#) presenta los fundamentos de los recursos de software y el trabajo relacionado. La [Sección 3](#) describe la metodología de revisión. La [Sección 4](#) muestra la revisión sistemática de la literatura. La [Sección 5](#) describe la discusión y la agenda de investigación. La [Sección 6](#) presenta las conclusiones y las perspectivas futuras.

2. Fundamentos de los Recursos de Software y el Trabajo Relacionado

La reutilización de software es el proceso que consiste en crear sistemas software a partir de software existente, en lugar de construirlos desde cero. Durante el SDLC, se crean numerosos productos para reutilización, desde algoritmos hasta documentos. El código fuente es el más comúnmente reutilizado y, junto con la reutilización del diseño, se ha popularizado (Sametinger, 1997). En relación con lo anterior y para los fines de este trabajo, un recurso de software se define como cualquier conjunto coherente de

artefactos que resuelven un problema específico o un conjunto de problemas encontrados en el SDLC, y puede ser cualquier producto de trabajo o actividad relacionada con el software (Zhao et al., 2010). La Tabla 1 muestra algunos ejemplos de estos tipos de recursos de software:

Tabla 1. Tipos de recursos de software.

Tipos de recursos de software	
Componentes	Una unidad de composición con interfaces especificadas contractualmente y únicamente dependencias de contexto explícitas. Puede implementarse de forma independiente y está sujeto a la composición de terceros (Szyperski, 2003).
Patrones	Una solución general repetible para un problema común en el diseño de software (Source Making, 2019).
Casos de prueba	Un grupo de condiciones bajo las cuales un evaluador determina si una aplicación de software funciona según los requisitos del cliente o no (GeeksforGeeks, 2022).
Bibliotecas	Un conjunto de datos y código de programación que se utiliza para desarrollar programas y aplicaciones de software (Sterling et al., 2018).
Frameworks	Una plataforma concreta o conceptual donde los desarrolladores o usuarios pueden especializar o anular selectivamente código común con funcionalidad genérica (Amazon, 2024a).
Servicios web	Una aplicación autónoma, modular, distribuida y dinámica que puede describirse, publicarse, ubicarse o invocarse a través de la red para crear productos, procesos y cadenas de suministro (IBM Corporation, 2024).
Paquetes	Un grupo de programas informáticos que pueden licenciarse, descargarse o suscribirse como un paquete de productos relacionados (Geeks for Geeks, 2024).
Kits de desarrollo de software	Un conjunto de herramientas y programas de software proporcionados por proveedores de hardware y software que los desarrolladores pueden utilizar para crear aplicaciones para plataformas específicas (Amazon, 2024b).
Interfaces de Programación de Aplicaciones	Un conjunto de reglas definidas que permiten que diferentes aplicaciones se comuniquen entre sí (Goodwin, 2024).
Kits de herramientas	Un único programa de utilidad, un conjunto de rutinas de software o un conjunto completo e integrado de utilidades de software que se utilizan para desarrollar y mantener aplicaciones y bases de datos (Nevagi, 2021).

Aplicaciones	Un tipo de programa informático que realiza funciones específicas (Halwai, 2021).
Algoritmos	Un tipo de diseño informático que realiza funciones para resolver un problema específico (Upadhyay, 2022).
Documentación	La documentación de software proporciona información sobre un programa de software para todos los involucrados en su creación, implementación y uso (Lutkevich, 2022).

Fuente: Elaboración propia.

Para el propósito específico de esta revisión, nos centramos en los recursos de software asociados directamente con la actividad de programación del Proceso de Desarrollo de Software (SDP, Software Development Process), que incluyen bibliotecas, frameworks, servicios web, paquetes, SDK, API, kits de herramientas, aplicaciones y algoritmos.

No se encontraron revisiones de literatura relacionadas específicamente con SRPMDS, pero sí un artículo de una Revisión Sistemática de la Literatura (SLR, Systematic Literature Review) para la toma de decisiones multicriterio y dos repositorios de código en los sitios web del Consorcio Decision Deck y de la Sociedad Internacional para MCDM. Los tres están relacionados con el tema estudiado al presentar algunos SRPMDS y se describen brevemente en los párrafos siguientes.

Inicialmente, se presenta una SLR para la toma de decisiones multicriterio en el desarrollo de software en (Zhou, 2015). Sin embargo, esta investigación no se centra en los recursos de software multicriterio disponibles en relación con ninguna actividad de SDP en particular, sino que explora lo que se ha hecho en general. Presenta un resumen del estado del arte del MCDA en SDP, identificando los métodos multicriterio utilizados para resolver problemas de decisión en actividades como requisitos, calidad, gestión de proyectos, diseño y componentes, mediante la estructuración y el análisis sistemáticos de la literatura sobre estos temas. A diferencia de esta revisión sistemática de la literatura, nuestro trabajo se centra en la actividad de programación del SDP.

Por otro lado, aunque los repositorios web revisados no presentan formalmente revisiones de la literatura sobre SRPMDS como tales, es importante incluirlas y comentarlas, ya que recopilan, describen y publican un amplio conjunto de recursos de software que implementan algoritmos de diferentes métodos multicriterio.

En primer lugar, el sitio web del Consorcio Decision Deck, una asociación francesa que desarrolla de forma colaborativa herramientas de software de código abierto para apoyar la toma de decisiones multicriterio, ofrece un amplio conjunto de algoritmos MCDA disponibles en línea como servicios web. Estos utilizan el estándar de datos XMCDA, una propuesta XML estructurada para representar objetos y datos emitidos desde el campo de MCDA, para intercambiar información e incluyen implementaciones algorítmicas de diversos métodos multicriterio, como AHP, TOPSIS, ELECTRE y UTA, entre otros (Decision Deck Consortium, 2022). Se puede acceder a los servicios web a través de la aplicación Diviz de Decision Deck o mediante solicitudes SOAP desde aplicaciones de terceros (Meyer & Bigaret, 2012).

En segundo lugar, la Sociedad Internacional para MCDM, ofrece una lista de diferentes tipos de software, como aplicaciones, servicios web y paquetes, que incluyen

implementaciones de algoritmos como MAUT, MAVT, NIMBUS y AHP. Algunos de ellos pueden integrarse y utilizarse para crear nuevas aplicaciones multicriterio. Están disponibles para descarga gratuita o de pago (MCDM Society, 2025).

Con base en lo anterior y dada la ausencia de una revisión de literatura sobre el uso de recursos de software para programación en el SDP, el objetivo de esta investigación es proporcionar una SLR sobre SRPMDS para obtener un inventario de lo que se ha hecho en este contexto en el periodo de 2012 a 2024 e identificar posibles brechas en la literatura para proponer áreas de investigación futura.

Este trabajo se centra en (i) artículos que presentan recursos de software que implementan algoritmos de métodos multicriterio que pueden incorporarse en la creación de nuevas aplicaciones y (ii) repositorios web de entidades confiables que alojan y delinean recursos de software de este tipo.

3. Metodología de la Revisión

Para realizar esta SLR, nos basamos en las directrices propuestas por Kitchenham et al. (2015) compuesta por cinco fases: 1) preguntas de investigación, 2) proceso de búsqueda, 3) selección de estudios, 4) extracción y 5) síntesis de datos. Entre las actividades realizadas durante estas fases se encuentran el establecimiento de las preguntas de investigación y sus objetivos; los términos de búsqueda y las fuentes seleccionadas; los criterios de inclusión y exclusión; y un método para la extracción y síntesis de la información.

3.1. Preguntas de investigación

Tabla 2. *Preguntas de investigación.*

Preguntas de investigación	
Pregunta	Descripción
RQ-1: ¿Qué tipos de recursos de software se han desarrollado para facilitar la creación de aplicaciones multicriterio? (Decision Deck Consortium, 2022; Figueroa Pérez et al., 2022a, 2022b)	Identificar los tipos de recursos desarrollados con este fin, como componentes, patrones, casos de prueba, bibliotecas, frameworks, servicios web, paquetes, kits de desarrollo de software, interfaces de programación de aplicaciones, kits de herramientas, Aplicaciones, algoritmos o documentación.

RQ-2: ¿Qué métodos multicriterio implementan los recursos de software desarrollados para la creación de aplicaciones multicriterio? (Figueroa Pérez et al., 2022b)	Identificar los métodos multicriterio implementados por recursos de software desarrollados, como AHP, SAW, TOPSIS, ELECTRE, PROMETHEE, SMART, por mencionar algunos.
RQ-3: ¿Qué sistemas operativos son compatibles con los recursos de software desarrollados para la creación de aplicaciones multicriterio? (Figueroa Pérez et al., 2022c)	Conocer qué sistemas operativos son compatibles con los recursos de software desarrollados.
RQ-4: ¿Qué herramienta de desarrollo se utilizó para crear el software?	Identificar la herramienta de desarrollo utilizada para crear el recurso de software, como C/C++, Java, Excel u otros.
RQ-5: ¿Qué tipo de licencia se utiliza para los recursos de software desarrollados para la creación de aplicaciones multicriterio? (Figueroa Pérez et al., 2022a)	Para saber si el recurso es gratuito o de pago.
RQ-6: ¿Está disponible el código fuente?	Para conocer si el tipo de licencia es de código abierto o comercial.

Fuente: Elaboración propia.

3.2. Proceso de búsqueda

Debido a la naturaleza de la revisión, la restringimos a artículos revisados por pares publicados en revistas, libros, capítulos de libros o actas de congresos. También incluye sitios web de instituciones de investigación y empresas privadas que presentan este tipo de software.

El área central de investigación en la que se pueden encontrar los artículos o el software pertinentes determina los términos de búsqueda específicos. Dado que el objetivo de la investigación es identificar los diferentes tipos de recursos de software creados para desarrollar aplicaciones multicriterio, se consideraron algunos términos alternativos para “multicriterio” y “recurso de software”. Por consiguiente, para la

conformación de la cadena de búsqueda, se tuvieron en cuenta sinónimos, grafías alternativas y términos relacionados. La cadena de búsqueda se armó de la siguiente forma: [parte 1] + [parte 2], donde [parte 1] puede contener los términos “multicriteria”, “mcda”, “mcdm”, “madm” o “modm” y [parte 2] los términos “software resource”, “software library”, “software framework”, “software web service”, “software package”, “software SDK”, “software API”, “software toolkit”, “software application” o “software algorithm”. Algunos ejemplos de las cadenas usadas incluyen “multicriteria software resource”, “mcda software API”, “mcdm software application”, etc. Las cadenas armadas se usaron en los buscadores exactamente como se describen en los ejemplos anteriores.

Las fuentes incluyeron bases de datos académicas en línea, herramientas de búsqueda comunes y sitios web seleccionados. Las fuentes seleccionadas para realizar la búsqueda se muestran en la [Tabla 3](#).

Tabla 3. Fuentes electrónicas usadas.

Fuentes electrónicas	
Fuente	URL
Google Scholar	https://scholar.google.com/
Scopus	https://www.scopus.com/
Springer	https://link.springer.com/
ACM Digital Library	https://www.acm.org/
IEEE	https://ieeexplore.ieee.org/Xplore
Software resources available on the Web	http://www.decision-deck.org/project/ (Decision Deck Consortium, 2022) http://decisionarium.tkk.fi/ (Hämäläinen, 2013) https://project-flo.de/ (Gunther et al., 2015) https://github.com/MCDALab/MEM (MCDALab, 2014)

Fuente: Elaboración propia.

3.3. Selección de estudios

Los criterios de inclusión y exclusión que se muestran en la [Tabla 4](#) se definieron de acuerdo con las preguntas de investigación que se muestran en la [Tabla 2](#). Se utilizaron para evaluar los datos recuperados, analizar e interpretar la literatura para realizar la selección de los estudios primarios y el software.

Tabla 4. Criterios de inclusión y exclusión.

Criterios de inclusión y exclusión	
Inclusión	Exclusión
IC1: Estudios o software publicados o lanzados entre 2012 y 2024	EC1: El estudio o software es una versión anterior de un trabajo más reciente
SIC2: El estudio o software analiza, discute o presenta cualquier recurso para el desarrollo de aplicaciones multicriterio	EC2: El estudio o software no se concentra en recursos para el desarrollo de aplicaciones multicriterio
IC3: El estudio o software responde al menos a una pregunta de investigación	EC3: Estudios o software duplicados

Fuente: Elaboración propia.

La aplicación de los criterios de inclusión y exclusión para la obtención de los estudios se organizó en tres etapas. Las etapas ayudaron a reducir el número de hallazgos tratando de conservar los trabajos relevantes para su posterior análisis.

- Etapa 1:
 - IC1: Estudios o software publicados o lanzados entre 2012 y 2024.
 - EC1: El estudio o software es una versión anterior de un trabajo más reciente.
- Etapa 2:
 - IC2: El estudio o software analiza, discute o presenta cualquier recurso para el desarrollo de aplicaciones multicriterio.
 - EC2: El estudio o software no se concentra en recursos para el desarrollo de aplicaciones multicriterio.
- Etapa 3:
 - IC3: El estudio o software responde al menos a una pregunta de investigación.
 - EC3: Estudios o software duplicados.

Para incluir un artículo o software en la revisión, se añadió un proceso de evaluación de calidad para garantizar su compatibilidad con los objetivos de la investigación. Para ello, se utilizó la lista de verificación de calidad de (Dybå & Dingsøyr, 2008), la cual se adaptó eliminando algunas preguntas que no contribuyen a la evaluación de este trabajo. Cada artículo o software se evaluó con la lista de verificación que se muestra en la Tabla 5. Las posibles respuestas a las preguntas pueden ser afirmativas o negativas. La calificación general de calidad de cada trabajo se calcula mediante la suma de todos los puntos. El rango permitido varía de 0 a 10 puntos, y 6 puntos es la puntuación mínima para considerar un trabajo aceptable. Tabla 5: Preguntas para la evaluación de la calidad.

Tabla 5. Preguntas para la evaluación de la calidad.

Preguntas para la evaluación de la calidad		
Número	Pregunta	Valor
QA1	¿Existe una declaración explícita de la pregunta de investigación?	1.0 pt.
QA2	¿Es adecuado el método de investigación del estudio para responder a la pregunta de investigación?	1.0 pt.
QA3	¿Se realizó una descripción del contexto de la investigación?	0.5 pt.
QA4	¿Se desarrolló el recurso de software?	2.0 pt.
QA5	¿Se probó el recurso de software?	2.0 pt.
QA6	¿Se abordaron los hallazgos de acuerdo con los objetivos de la investigación?	1.0 pt.
QA7	¿Se validaron adecuadamente los resultados?	0.5 pt.
QA8	¿Es el estudio valioso para la investigación de recursos de software para la programación de sistemas de apoyo a la toma de decisiones multicriterio?	2.0 pt.

Fuente: Elaboración propia.

3.4. Extracción de datos

La estrategia de extracción de datos consistió en extraer los datos de cada estudio según el formato que se muestra en la [Tabla 6](#). Los datos D1-D7 contienen información general de cada estudio y los datos D8-D14 incluyen información que ayuda a responder las preguntas de investigación.

Tabla 6. Formato de extracción de datos.

Formato de extracción de datos	
Número	Pregunta
D1	Título
D2	Autor(es)
D3	Año
D4	Fuente

D5	Tipo de publicación
D6	Palabras clave
D7	Resumen
D8	Preguntas de investigación relacionadas
D9	Tipo de recurso desarrollado
D10	Método multicriterio implementado
D11	Sistemas operativos soportados
D12	Herramienta de desarrollo empleada
D13	Tipo de licencia
D14	Disponibilidad de código fuente

Fuente: Elaboración propia.

3.5. Síntesis de datos

Para analizar las características de los recursos de software, primero realizamos un análisis contextual básico inspeccionando los elementos encontrados asociados a las preguntas de investigación y, posteriormente, se llevó a cabo una revisión exhaustiva de los mismos mediante el método de Análisis Formal de Conceptos (FCA, Formal Concept Analysis) (Wille, 2005). El FCA se utiliza principalmente para el análisis de datos estructurados en elementos que son abstracciones formales de conceptos, lo que permite una interpretación comprensible y significativa. Este método es un enfoque de agrupamiento conceptual que proporciona descripciones intencionales de los conceptos abstractos que produce.

4. Revisión Sistemática de la Literatura

4.1. Análisis de los datos de publicaciones

Tras recopilar los resultados de la búsqueda, solo se consideraron los trabajos pertinentes que contenían elementos de las cadenas de búsqueda antes mencionadas. Inicialmente, se obtuvieron 190 documentos o software, que, tras aplicar los criterios de inclusión y exclusión, se redujeron a un conjunto final de 29 documentos o software. La Tabla 7 muestra el proceso completo de selección de estudios por etapas y fuente.

Tabla 7. Proceso de búsqueda y selección de estudios.

Proceso de búsqueda y selección de estudios				
Fuente	Resultados	Etapa 1	Etapa 2	Etapa 3
Google Scholar	66	55	42	12
Scopus	34	31	8	2
Springer	11	10	6	3
ACM Digital Library	32	31	8	1
Software resources available on the Web	19	17	8	0
Total	190	167	95	29

Fuente: Elaboración propia.

A continuación, se muestra la [Tabla 8](#) de los 29 documentos o software seleccionados con su año de publicación, título y tipo de recurso software presentado.

Tabla 8. Trabajos relacionados.

Trabajos relacionados			
ID	Año	Título	Tipo de recurso
S01	2012	A Fuzzy Logic Multi-Criteria Decision Framework for Selecting IT Service Providers (Karami & Guo, 2012)	Framework
S02	2013	Global space for decision support (Hämäläinen, 2013)	Aplicación
S03	2013	A Recommendation Framework Based on the Analytic Network Process and its Application in the Semantic Technology Domain (Radulovic et al., 2013)	Framework
S04	2014	MCDA Lab (MCDALab, 2014)	Algoritmo

Continúa en la siguiente página

Trabajos relacionados			
ID	Año	Título	Tipo de recurso
S05	2014	Interactive TOPSIS algorithms for solving multi-level nonlinear multi-objective decision-making problems (Baky, 2014)	Algoritmo
S06	2014	MS Excel based software support tools for decision problems with multiple criteria (Jablonsky, 2014)	Kit de herramientas
S07	2015	PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments (Siraj et al., 2015)	Aplicación
S08	2015	Development of tools for solving (vector, set) optimization problems (Gunther et al., 2015)	Kit de herramientas
S09	2015	DEXi: A Program for Multi-Attribute Decision Making Version 5.05 (Bohanec, 2015)	Aplicación
S10	2015	Decerns: A Framework for Multi-Criteria Decision Analysis (Yatsalo et al., 2015)	Aplicación
S11	2016	Development of a Comparative Multiple Criteria Framework for Ranking Pareto Optimal Solutions of a Multiobjective Reservoir Operation Problem (Bozorg-Haddad et al., 2016)	Framework
S12	2016	A new web-based framework development for fuzzy multicriteria group decision-making (Hanine et al., 2016)	Framework
S13	2017	Supporting the multi-criteria decision aiding process: R and the MCDA package (Bigaret et al., 2017)	Biblioteca
S14	2017	The vector linear program solver Bensolve - notes on theoretical background (Löhne & Weißing, 2017)	Algoritmo

Continúa en la siguiente página

Trabajos relacionados			
ID	Año	Título	Tipo de recurso
S15	2017	A New Method for Optimizing a Linear Function over the Efficient Set of a Multiobjective Integer Program (Boland et al., 2017)	Algoritmo
S16	2018	A State-of-the-Art tool for multi-criteria decision analysis and decision tree modeling with imprecise information (MAD Preference, 2018)	Aplicación
S17	2018	An integrated framework for multi-criteria optimization of thin concrete shells at early design stages (Gomes et al., 2018)	Framework
S18	2018	FRiCS: A Framework for Risk-driven Cloud Selection (Cabarcos et al., 2018)	Framework
S19	2019	FLINTSTONES 2.0 an Open and Comprehensive Fuzzy Tool for Multi-criteria Decision Analysis (Labella & Martínez, 2019)	Aplicación
S20	2019	Framework for Social Media Analytics based on MultiCriteria Decision Making (MCDM) model (Muruganantham & Gandhi, 2019)	Framework
S21	2020	A Framework for Welding process selection (Omar & Soltan, 2020)	Framework
S22	2020	Green and Regional? A multi-criteria assessment framework for the provision of green electricity for electric vehicles in Germany (Fabianek et al., 2020)	Framework
S23	2021	DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization (Misitano et al., 2021)	Framework
S24	2021	An MCDM-based framework for the selection of renewable energy system simulation tool for teaching and learning at university level (Ighravwe et al., 2021)	Framework

Continúa en la siguiente página

Trabajos relacionados			
ID	Año	Título	Tipo de recurso
S25	2021	International Society on MCDM (MCDM Society, 2025)	Aplicación
S26	2022	The Decision Deck project collaboratively develops Open Source software tools to support the Multi-Criteria Decision Aiding (MCDA) process (Decision Deck Consortium, 2022)	Servicio Web
S27	2022	A multi-objective supplier selection framework based on user-preferences (Toffano et al., 2022)	Framework
S28	2023	Pymcdm The universal library for solving multi-criteria decision-making problems (Kizielewicz et al., 2023)	Biblioteca
S29	2024	Version [1.1]- [pyFDM: A python library for uncertainty decision analysis methods] (Więckowski & Sałabun, 2024)	Biblioteca

Fuente: Elaboración propia.

Utilizando los estudios y software mostrados en la [Tabla 8](#) anterior, procedimos, en primer lugar, a realizar un análisis contextual básico que dé respuesta a las preguntas de investigación. Posteriormente, se realizó un análisis más profundo mediante FCA, generando lattices que nos permitieron identificar y comprender relaciones más complejas entre los elementos encontrados.

4.1.1. Artículos y software por año.

Los artículos y software publicados por año (hasta abril de 2024) se muestran en la [Figura 1](#). En ella se observa que la investigación relacionada con este tema se ha mantenido en los últimos años. Destacan los años 2015 con 4 trabajos y 2017, 2018 y 2021 con 3.

Figura 1: Artículos y software publicados por año.

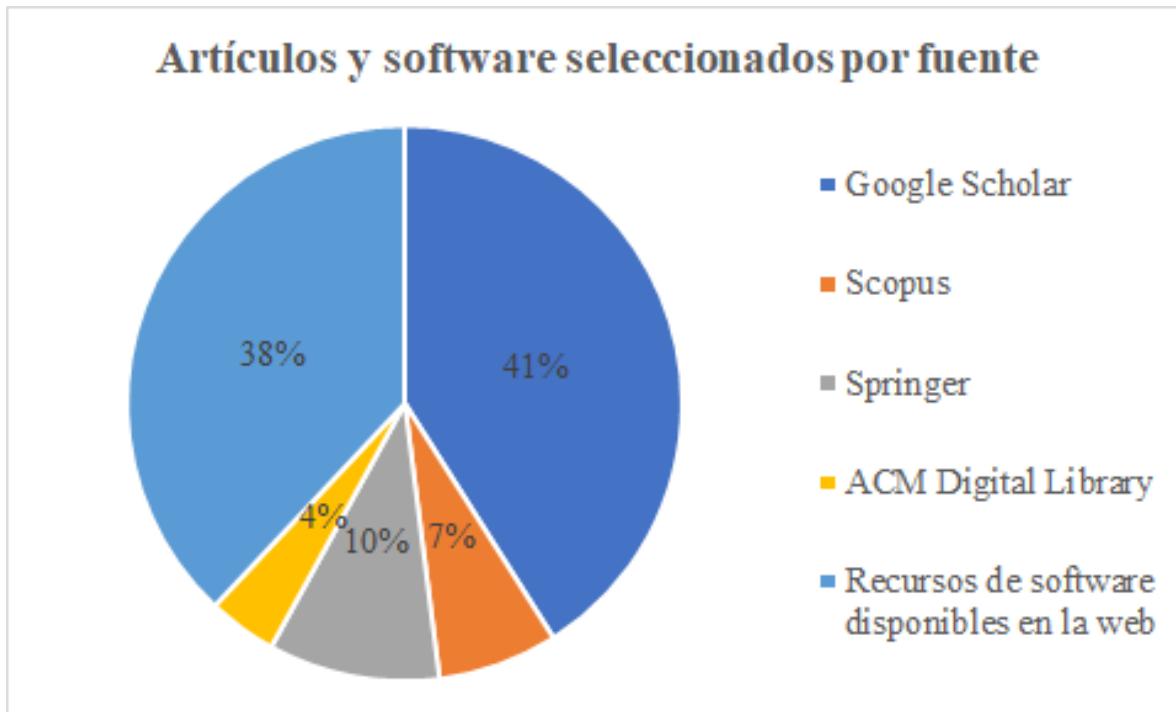


Fuente: Elaboración propia.

4.1.2. Artículos y software por fuente

Los artículos y software por fuente se muestran en la Figura 2. El 41 % de los trabajos fueron artículos que se encontraron en el buscador de Google Académico y 38 % fueron software encontrados en la web. El resto de los trabajos se distribuyó en diferentes bases de datos académicas como se puede observar en la imagen.

Figura 2: Artículos y software seleccionados por fuente.

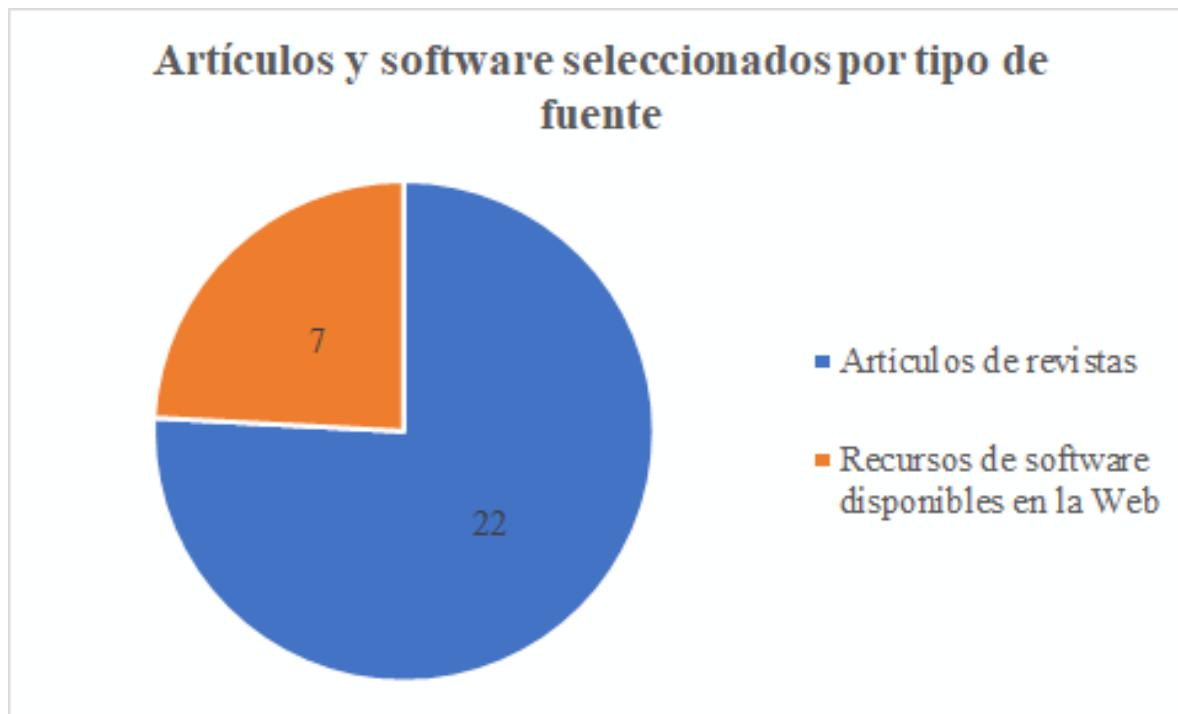


Fuente: Elaboración propia.

4.1.3. Artículos y software por tipo de fuente

Los artículos y software por tipo de fuente se muestran en la [Figura 3](#). De estos 22 fueron artículos de revistas y 7 recursos software disponibles en la web.

Figura 3: Artículos y software seleccionados por tipo de fuente.



Fuente: Elaboración propia.

4.2. Respuestas a preguntas de investigación

4.2.1. RQ-1: ¿Qué tipos de recursos de software se han desarrollado para facilitar la creación de aplicaciones multicriterio?

La Tabla 9 muestra los tipos de recursos de software desarrollados para facilitar la creación de aplicaciones multicriterio. Se observa que se encontraron 12 frameworks y 7 aplicaciones, siendo estos los tipos de recursos software más desarrollados. Se encontraron 4 algoritmos y 3 bibliotecas y 2 kits de herramientas. No se encontraron SDK's, API's, paquetes ni no especificados.

Tabla 9. Tipos de recursos de software desarrollados para ayudar a crear aplicaciones multicriterio.

Tipos de recursos de software desarrollados para ayudar a crear aplicaciones multicriterio	
Tipo de recurso	Cantidad
Biblioteca	3
Framework	12
Servicio Web	1

SDK	0
API	0
Kit de herramientas	2
Paquete	0
Aplicación	7
Algoritmo	4
No especificado	0

Fuente: Elaboración propia.

4.2.2. RQ-2: ¿Qué métodos multicriterio implementan los recursos de software desarrollados para la creación de aplicaciones multicriterio?

La Tabla 10 muestra los métodos multicriterio implementados por los recursos de software encontrados. Entre los métodos implementados, destacan TOPSIS y AHP con 12 y 11 apariciones respectivamente. Le siguen PROMETHE con 5 y ELECTRE con 4. DEA, COPRAS, MAVT y VIKOR aparecen 3 veces y finalmente 20 métodos aparecen una vez.

Tabla 10. *Métodos multicriterio implementados por recursos de software.*

Métodos multicriterio implementados por recursos de software					
Método	Cantidad	Método	Cantidad	Método	Cantidad
AHP	11	ELECTRE	4	TOPSIS	12
DEA	3	ANP	1	MEM	1
WSA	1	ORESTE	1	PROMETHE	5
MAPPAC	1	DEX	1	MAVT	3
FLOWSORT	1	MAUT	1	ProMAA	1
FMAT	1	FMAA	1	VIKOR	3
CP	1	COPRAS	3	WASPAS	2
VLP	1	MOIP	1	NIMBUS	2
NAUTILUS	1	DEMATEL	1	COMET	1
RUBIS	1	UTA	1	MILP	1

Continúa en la siguiente página

Métodos multicriterio implementados por recursos de software					
Método	Cantidad	Método	Cantidad	Método	Cantidad
No especificado	3				

Fuente: Elaboración propia.

4.2.3. RQ-3: ¿Qué sistemas operativos son compatibles con los recursos de software desarrollados para la creación de aplicaciones multicriterio?

La Tabla 11 muestra qué sistemas operativos son compatibles con los recursos software encontrados. Aquí, Windows tiene 25 apariciones, seguido de Linux con 15 y Mac con 13. Los sitios web y las aplicaciones Android e IOS también están presentes, aunque en cantidades menores con 5, 2 y 2 apariciones respectivamente.

Tabla 11. Sistemas operativos compatibles con recursos de software.

Sistemas operativos compatibles con recursos de software	
Sistema operativo	Cantidad
Windows	23
Linux	15
Mac	13
Android	2
iOS	2
Web	5
No especificado	1

Fuente: Elaboración propia.

4.2.4. RQ-4: ¿Qué herramienta de desarrollo se utilizó para crear el software?

La Tabla 12 muestra las diferentes herramientas de desarrollo utilizadas para crear los recursos de software. La información encontrada indica que Java y Python se utilizaron 6 veces, C/C++ Excel y Matlab 4 veces y 5 herramientas aparecen 1 vez.

Tabla 12. Herramientas de desarrollo utilizada para crear el recurso de software.

Herramientas de desarrollo utilizada para crear el recurso de software	
Herramienta de desarrollo	Cantidad
C/C++	4
Java	6
Excel	4
MatLab	4
C#	1
Javascript	1
PHP	1
R	1
Python	6
Delphi	1
No especificado	2

Fuente: Elaboración propia.

4.2.5. RQ-5: ¿Qué tipo de licencia se utiliza para los recursos de software desarrollados para la creación de aplicaciones multicriterio?

La Tabla 13 muestra el tipo de licencia de los recursos de software revisados. Se encontraron 21 recursos gratuitos, 3 de paga y 6 no lo especificaban.

Tabla 13. Tipo de licenciamiento de los recursos de software.

Tipo de licenciamiento de los recursos de software		
Gratis	Paga	No especificado
21	3	6

Fuente: Elaboración propia.

4.2.6. RQ-6: ¿Está disponible el código fuente?

La Tabla 14 muestra si el código fuente del recurso está disponible. Se encontró que 15 si cuentan con él, 10 no y 5 no lo especifican.

Tabla 14. Código fuente disponible.

Código fuente disponible		
Si	No	No especificado
15	10	5

Fuente: Elaboración propia.

4.3. Resultados del análisis formal de conceptos

Utilizamos el FCA para analizar en mayor profundidad la distribución de trabajos revisados en los atributos SRPMDDSS. Este método permite agrupar los trabajos en una red que incluye los diferentes atributos abordados para visualizar la similitud y el nivel de tratamiento en los trabajos revisados. Cada nodo identifica un atributo y el número de trabajos que lo abordan y las líneas indican una relación de subconjunto (Wille, 2005). El análisis se realizó con el software FCART Formal Concept Analysis Research Toolbox 0.9.5 (Formal Concept Analysis Research Toolbox (Formal Concept Analysis Research Toolbox (FCART, s.f.)). Esta herramienta permite la construcción, visualización y manipulación de redes de conceptos, proporcionando una amplia variedad de funciones de visualización, como diferentes etiquetas para atributos, objetos, colores y tamaños. En la [Tabla 15](#) se muestra un ejemplo ilustrativo para diez artículos con cuatro atributos. Los valores de los atributos son 1 y 0 según si el atributo es abordado o no en el artículo, respectivamente.

Tabla 15. Ejemplo de 10 documentos con 4 atributos.

Ejemplo de 10 documentos con 4 atributos				
Documento	A1	A2	A3	A4
Paper 1	1	0	0	1
Paper 2	1	0	0	0
Paper 3	0	1	1	1
Paper 4	1	1	1	1
Paper 5	1	0	0	0
Paper 6	1	0	0	0
Paper 7	0	1	0	1
Paper 8	0	0	1	0
Paper 9	0	0	0	1

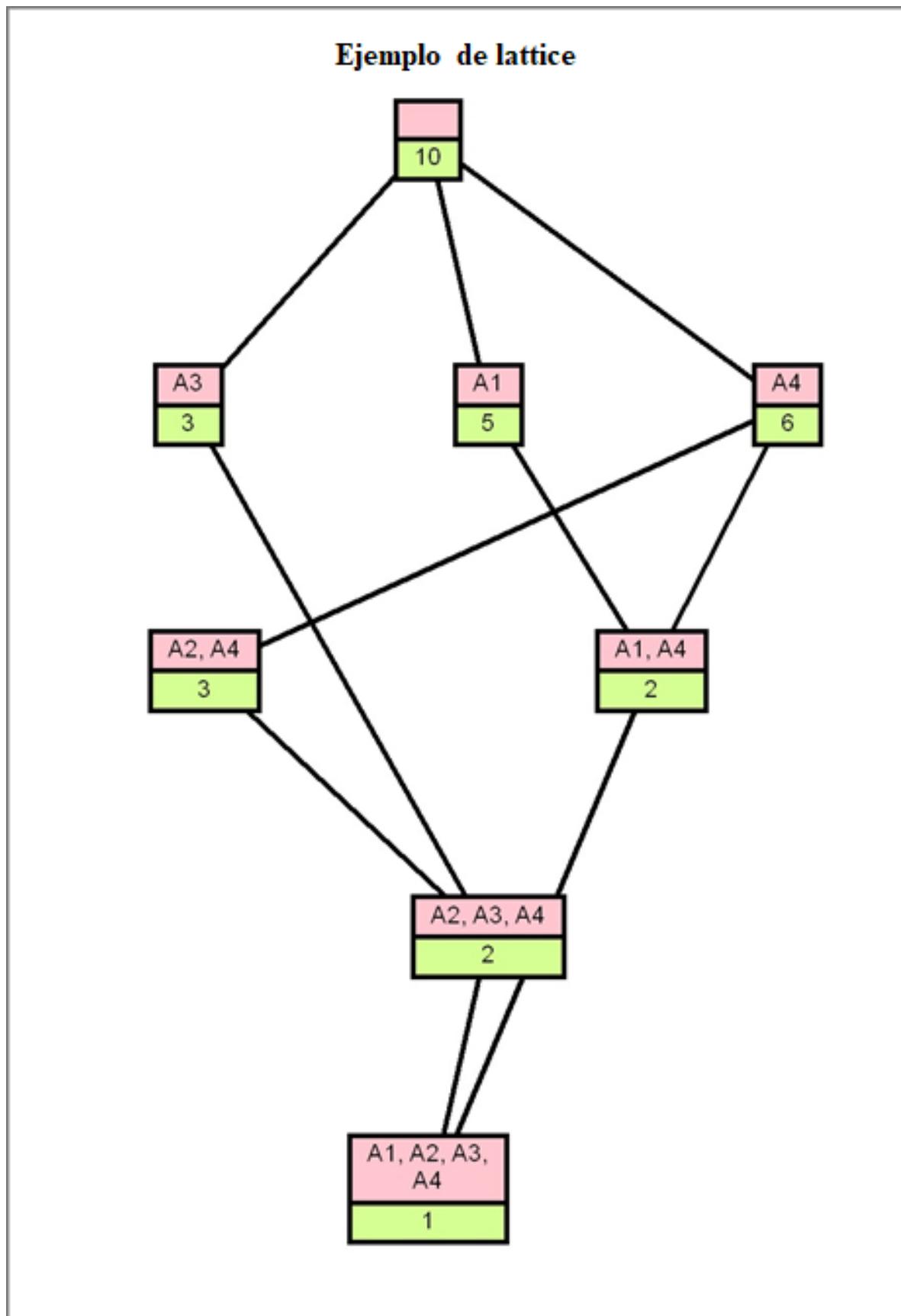
Continúa en la siguiente página

Ejemplo de 10 documentos con 4 atributos				
Documento	A1	A2	A3	A4
Paper 10	0	0	0	1

Fuente: Elaboración propia.

El ejemplo de red generado en la [Figura 4](#), generado a partir de la [Tabla 15](#), muestra que cada artículo aborda los atributos de su nodo y todos los atributos de los nodos ascendentes. En la [figura](#) se puede apreciar que 6 artículos abordan el atributo A4 (papers 1,3,4,7,9 y 10 en la [tabla](#)); 3 artículos, el A3 (papers 3,4 y 8 en la [tabla](#)); 5 artículos, el A1 (papers 1,2,4,5 y 6 en la [tabla](#)) y el A2 3 artículos (papers 3,4 y 7). Además de ello, se pueden visualizar en las intersecciones los atributos que son abordados por varios artículos. Así, 3 artículos abordan el A2 y el A4; 2 artículos, el A1 y el A4; 2 artículos, el A2, el A3 y el A4; y, finalmente, 1 artículo, el A1, el A2, el A3 y el A4. La red se visualiza con recuentos de objetos, lo que permite ver fácilmente la frecuencia con la que se aborda un atributo.

Figura 4: Ejemplo de lattice.

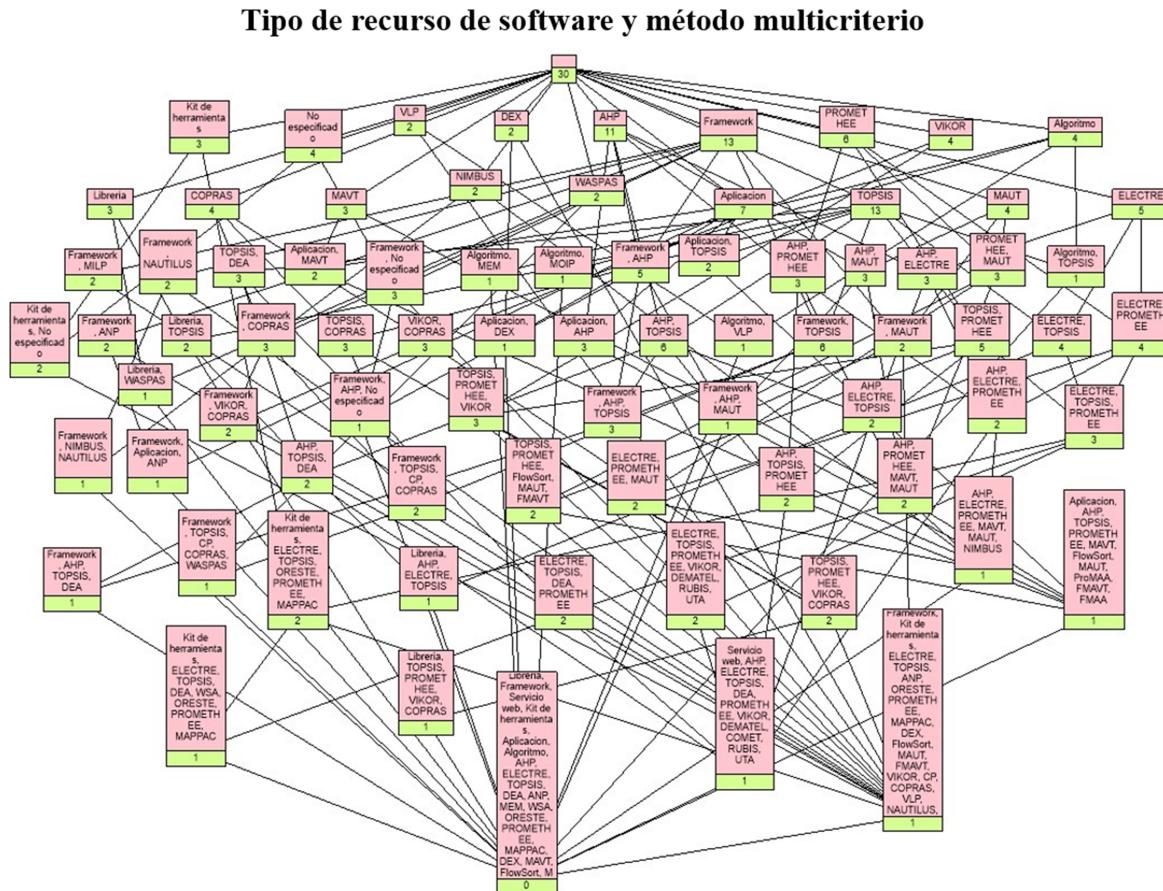


Como se puede observar en las secciones anteriores de la [Tabla 9](#), los Tipos de Recursos de Software Más Desarrollados fueron Frameworks (12), Aplicaciones (7), Algoritmos (4), Bibliotecas (3), Kits de Herramientas (2) y Servicios Web (1), representando el 100 % de los trabajos revisados. Por lo tanto, para reducir el tamaño del lattice generado y facilitar su lectura y análisis, los utilizamos como base, excluyendo aquellos tipos de software para los que no se presentó desarrollo (SDK, API y Paquete). En las siguientes subsecciones, profundizamos en la información recopilada por FCA para identificar algunas relaciones interesantes que no son fáciles de detectar a simple vista.

4.3.1. Lattice 1: Tipo de recurso de software y métodos multicriterio (RQ-1 y RQ2)

El lattice de la [Figura 5](#) analiza la relación entre MDTSR y los métodos multicriterio implementados. Muestra que, en conjunto, los métodos más utilizados fueron TOPSIS y AHP, con 12 y 11 respectivamente (6 de ellos en conjunto). Individualmente, 5 frameworks utilizaron TOPSIS y 5 AHP; 3 aplicaciones implementan AHP y 2 TOPSIS; solo 1 algoritmo utilizó TOPSIS. Además de lo anterior, los nodos de la red en la jerarquía muestran un conjunto de métodos que incluyen PROMETHEE, NIMBUS, ELECTRE, MAVT, VIKOR, COPRAS, MEM, VLP, DEX, MILP, MAUT, NAUTILUS, MOIP, ANP, CP y WASPAS, con una o dos apariciones por tipo de software y, en algunos casos, implementados en combinación con otros métodos.

Figura 5: Tipo de recurso de software y método multicriterio.

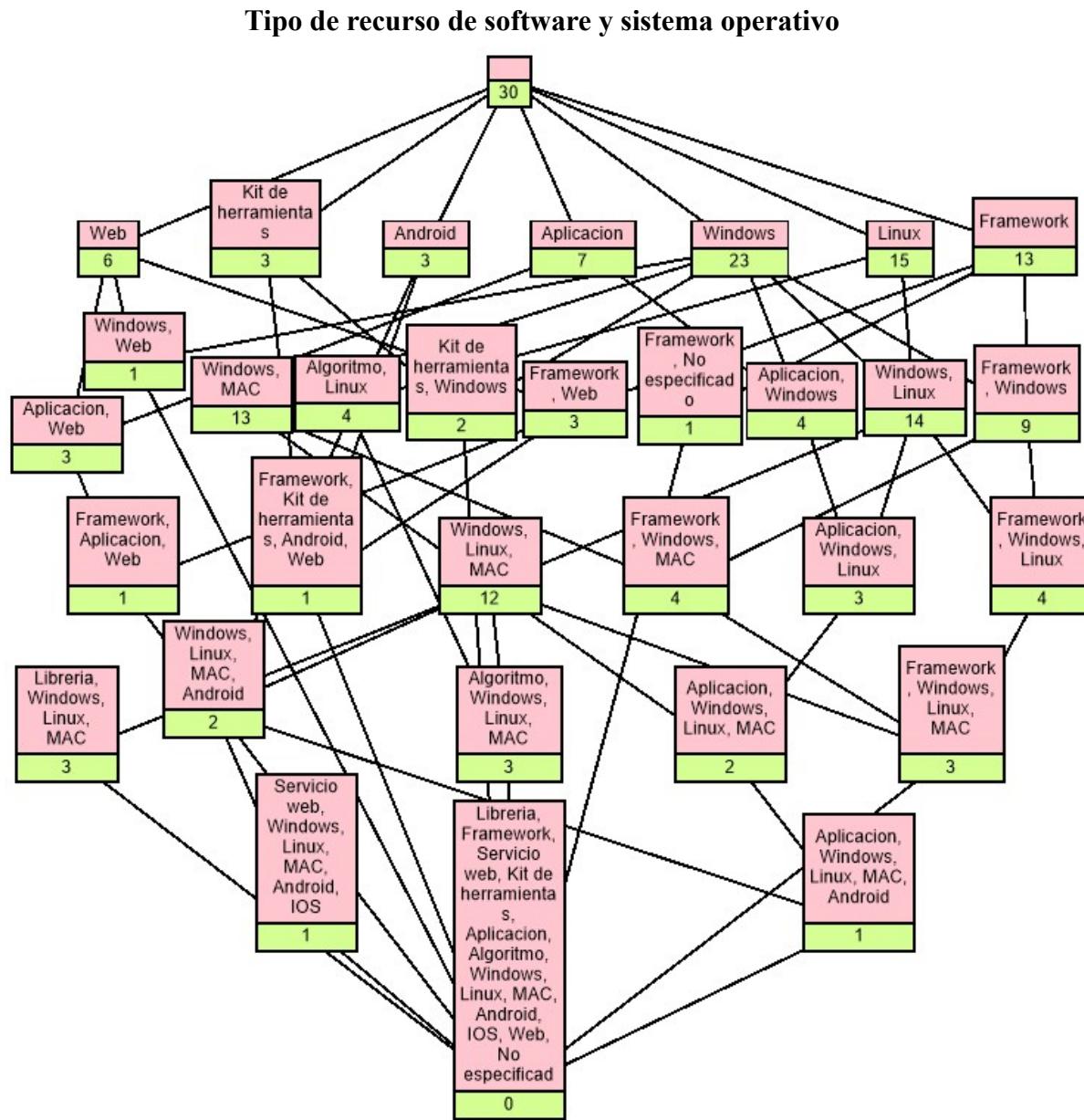


Fuente: Elaboración propia.

4.3.2. Lattice 2: Tipo de recurso de software y sistemas operativos (RQ-1 y RQ3)

El lattice de la [Figura 6](#) analiza la relación entre MDTSR y los sistemas operativos compatibles. Se observa que, en total, 23 recursos de software funcionan en Windows, 15 en Linux, 13 en Mac, 5 en la web y 2 en Android. De estos, 9 frameworks funcionan en Windows, 4 en Windows+Linux, 4 en Windows+Mac, 3 en Windows+Linux+Mac y 2 en la web. En cuanto a las aplicaciones, 4 funcionan en Windows y 2 en Windows+Linux+Mac. En cuanto a los algoritmos, 4 funcionan en Linux y 3 en Windows+Linux+Mac.

Figura 6: Tipo de recurso de software y sistema operativo.



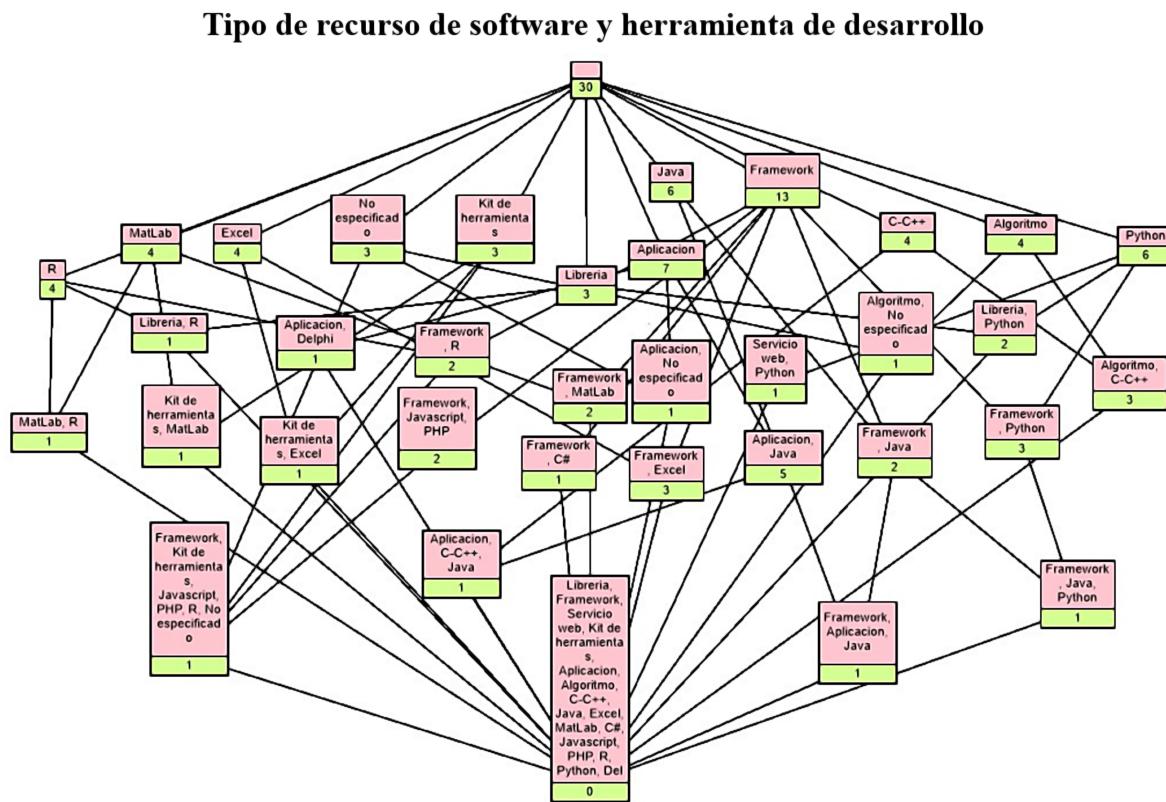
Fuente: Elaboración propia.

4.3.3. Lattice 3: Tipo de recurso de software y herramienta de desarrollo (RQ-1 y RQ-4)

El lattice de la Figura 7 analiza la relación entre MDTSR y la herramienta de desarrollo de software utilizada para su desarrollo. Muestra que no hay una herramienta claramente dominante, aunque Java y Python destacan ligeramente con 6 cada una. En concreto, 3 frameworks se desarrollaron con Python, 3 con Excel, 1 con R, 2 con Matlab, 2 con Java, 1 con C# y 3 con otras herramientas. En cuanto a las aplicaciones, 5 se desarrollaron con Java, 1 con Delphi y 1 con una combinación de C/C++ y Java. Respecto a los algoritmos,

3 se desarrollaron con C/C++ y 1 no se especificó.

Figura 7: Tipo de recurso de software y herramienta de desarrollo.

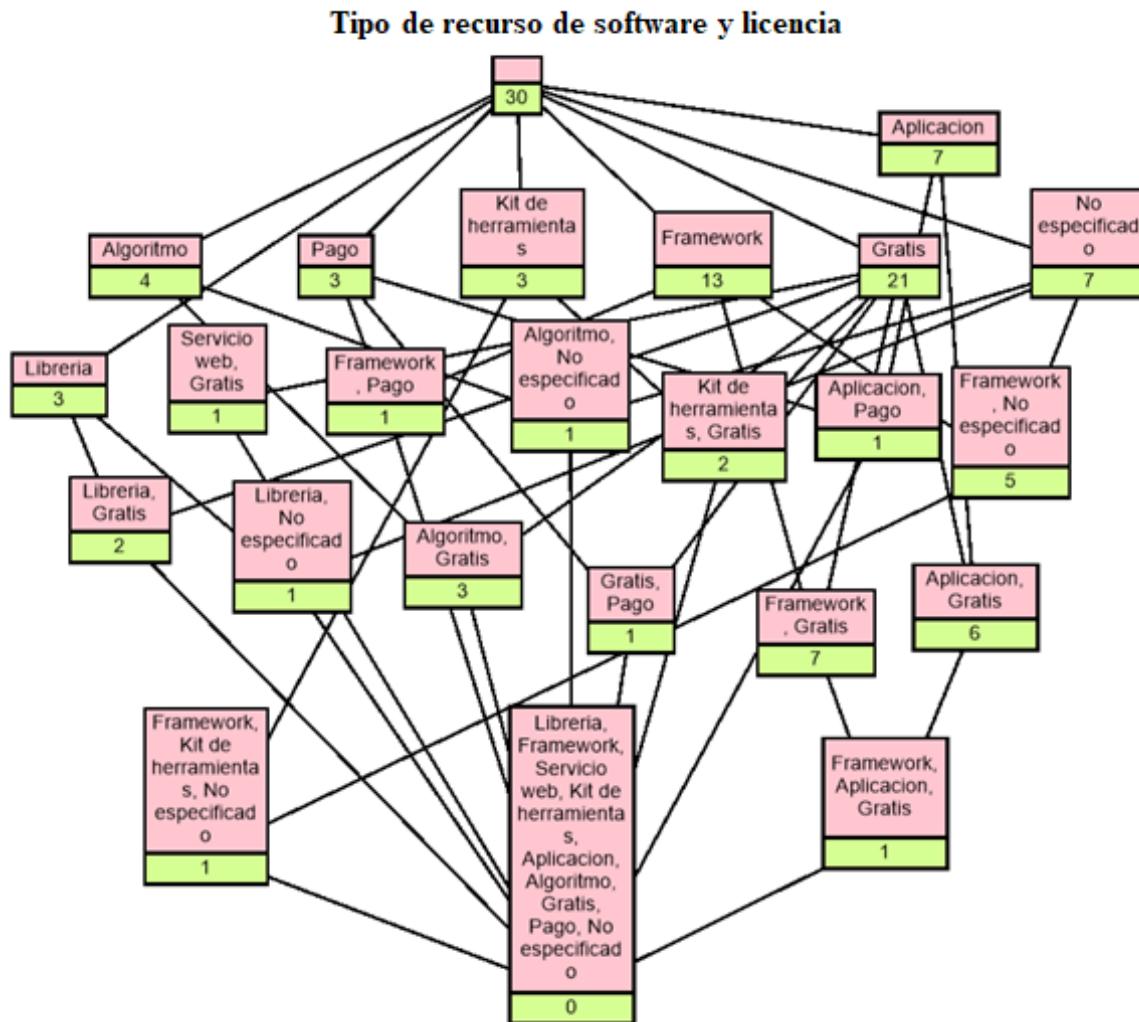


Fuente: Elaboración propia.

4.3.4. Lattice 4: Tipo de recurso de software y licencias (RQ-1 y RQ-5)

El lattice de la Figura 8 analiza la relación entre MDTSR y sus licencias. Muestra que 21 son gratuitas, 3 son de pago y el resto no las especifica. En concreto, 7 frameworks son gratuitos y 1 tiene una versión de pago, 6 aplicaciones son gratuitas y 1 de pago, y 3 algoritmos son gratuitos.

Figura 8: Tipo de recurso de software y licencia.

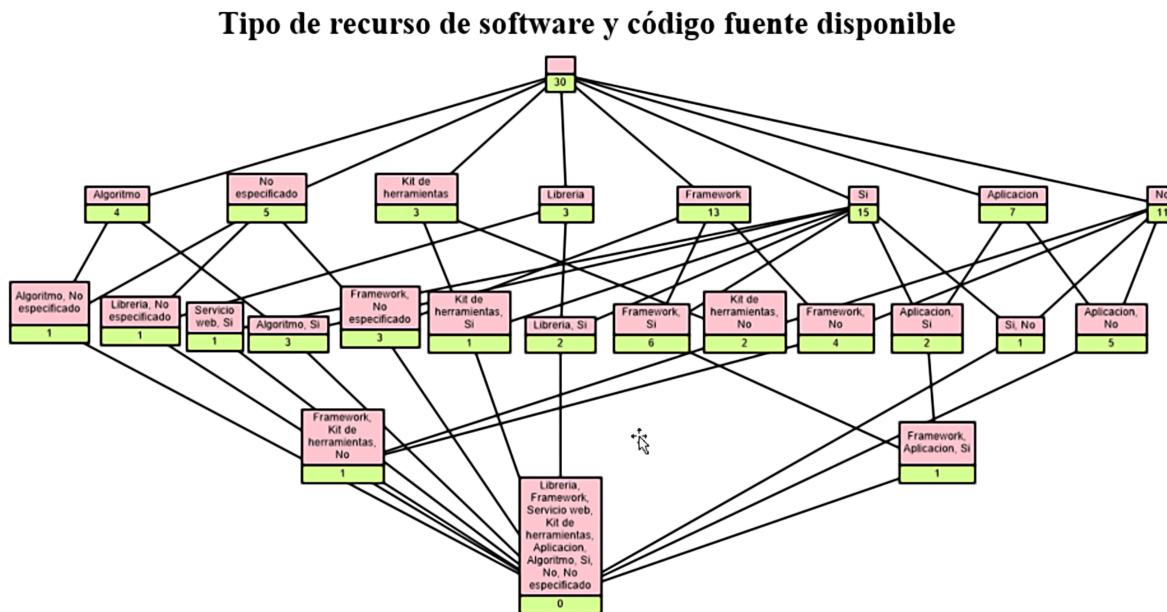


Fuente: Elaboración propia.

4.3.5. Lattice 5: Tipo de recurso de software y código fuente disponible (RQ-1 y RQ-6)

El lattice de la Figura 9 analiza la relación entre MDTSR y la disponibilidad de su código fuente. Muestra que 15 tienen el código fuente disponible, 10 no y el resto no lo especifica. En concreto, 6 frameworks, 2 aplicaciones y 3 algoritmos proporcionan código fuente. Además, 3 frameworks y 5 aplicaciones no lo hacen.

Figura 9: Tipo de recurso de software y código fuente disponible.



Fuente: Elaboración propia.

5. Discusión y Agenda de Investigación

Este trabajo identificó 29 investigaciones relacionadas con recursos de software para el desarrollo de aplicaciones multicriterio. Para obtenerlas, primero analizamos 190 trabajos publicados entre 2012 y 2024. Posteriormente, se clasificaron según las preguntas de investigación y sus atributos asociados. Finalmente, se seleccionaron solo aquellos directamente relacionados con el tema de interés.

De estos estudios se extrajeron los siguientes datos: tipos de recursos de software desarrollados, métodos multicriterio implementados, sistemas operativos compatibles, herramienta de desarrollo utilizada para construir el recurso de software, tipo de licencia ofrecida y si el código fuente del software desarrollado está disponible.

En cuanto a los tipos de recursos de software desarrollados, los resultados muestran que los frameworks, con 12, son los más desarrollados, seguidos de 7 aplicaciones y 4 algoritmos. Otros recursos, como bibliotecas, servicios web, paquetes, SDK, API y kits de herramientas, se han desarrollado escasamente, con entre 0 y 3 reportados. Los frameworks, las aplicaciones y los algoritmos predominan en este grupo. Por lo tanto, la mayoría de los SRPMDS existentes tienen un propósito específico y se centran en un tipo de problema específico. Esto demuestra que la mayoría de los SRPMDS desarrollados no son de propósito general y no se pueden adaptar fácilmente para crear nuevas aplicaciones de dominios distintos a aquel para el que fueron diseñados.

Los resultados relacionados con los métodos multicriterio implementados por recursos de software muestran que AHP y TOPSIS son los métodos más utilizados, con un total de 23 implementaciones. Además de estos, se implementaron 28 métodos, pero

su número fue mucho menor que el de los anteriores. Consideramos esto normal, ya que, según diversas revisiones bibliográficas realizadas en diferentes áreas, actualmente los métodos multicriterio más utilizados para la toma de decisiones son AHP y TOPSIS, implementados por numerosas herramientas gratuitas y comerciales con buenos resultados, lo que podría llevarlos a ser preferidos frente a opciones más nuevas o menos conocidas. En cuanto a los sistemas operativos compatibles con recursos de software, los resultados muestran que la gran mayoría funciona con Windows (23), Linux (15) y Mac (13) o es compatible con varios de estos sistemas operativos simultáneamente, dejando muy atrás la compatibilidad con web o móviles, con tan solo 5 y 3 apariciones cada uno. Por lo tanto, es evidente que predominan los sistemas operativos de escritorio, ya que la mayoría de este software se creó cuando estos sistemas eran tendencia. Sin embargo, esta situación actual puede representar un problema que reduce la posibilidad de desarrollar software más acorde con los tiempos modernos, donde las tecnologías web y móviles tienen un papel predominante.

Se utilizaron diversas herramientas para crear los recursos de software, entre las que destacan Java y Python, con 6 apariciones cada una. El resto se distribuye entre C#, JavaScript, Excel y otras. Ninguna de las herramientas utilizadas predomina claramente, abarcando desde lenguajes de programación y scripts hasta hojas de cálculo, lo que genera problemas de independencia de plataforma, velocidad de ejecución o escasas posibilidades de reutilización para el desarrollo de nuevas aplicaciones multicriterio.

Alrededor del 80 % del software revisado cuenta con información sobre licencias o disponibilidad del código fuente, mientras que el resto no. Es una cifra aceptable, a pesar de que gran parte de los recursos desarrollados no parecen estar orientados a un uso más allá del trabajo de investigación al que pertenecen.

6. Conclusiones y Perspectivas Futuras

El objetivo principal de este trabajo fue revisar el estado actual de la investigación sobre recursos de software para la programación de sistemas de soporte de decisiones multicriterio. Estos recursos tienen una importancia práctica significativa en diversas áreas del desarrollo de software, como el análisis, el diseño, la programación, las pruebas, etc. Los sistemas de escritorio se han utilizado durante décadas en el desarrollo de DSS. Sin embargo, su incapacidad para proporcionar un método práctico y flexible para respaldar problemas reales de toma de decisiones en situaciones que requieren trabajo colaborativo o individual en múltiples plataformas ha impulsado a desarrolladores, investigadores y profesionales a aprovechar los nuevos avances en el desarrollo de software web y móvil. Este documento describe el estado actual del uso de recursos de software para el desarrollo de MDSS. La revisión se centró en tres aspectos principales: (1) los tipos de recursos de software desarrollados para facilitar la creación de MDSS, (2) los métodos multicriterio utilizados para implementar recursos de software en los MDSS desarrollados, y (3) los sistemas operativos, las herramientas de desarrollo, el tipo de licencia y el código fuente utilizados para respaldar los recursos de software desarrollados para la creación de MDSS. La investigación futura en este campo puede extenderse en diferentes direcciones. Una de ellas es el desarrollo de recursos de software genéricos con licencia y código fuente libres, que puedan adaptarse o integrarse

fácilmente para crear nuevo software multicriterio en desarrollo. Esto incluye bibliotecas, servicios web, paquetes, API o kits de herramientas, que se han explorado poco, ya que la investigación actual se ha centrado más en la construcción de frameworks o aplicaciones para dominios problemáticos específicos que no fueron diseñados para ser reutilizados. La investigación futura en este tema es, sin duda, de gran importancia para diversificar las herramientas actuales para el desarrollo de MDSS, proporcionando así un apoyo significativo al responsable de la toma de decisiones en la resolución de problemas de decisión multicriterio. El tema mencionado está estrechamente relacionado con otra posible línea de investigación futura. Dado que es posible construir MDSS a partir de diferentes recursos de software, sería interesante explorar maneras de combinar algunos de ellos. Esta combinación de recursos de software podría tener un impacto beneficioso en la eficiencia y eficacia de las decisiones multicriterio tomadas mediante estos sistemas. Otra línea de investigación futura, que podría ser de interés, es la integración de más métodos multicriterio que han sido ampliamente reportados con buenos resultados en la literatura. La mayoría de los recursos desarrollados utilizan principalmente solo dos métodos multicriterio: AHP y TOPSIS, dando poca importancia o dejando de lado muchos otros como ELECTRE, PROMETHEE, MAUT, etc. También es importante investigar las similitudes y diferencias entre los métodos MCDA. Dado que existe un gran número de métodos MCDA, el analista debe poder recomendar el más conveniente al responsable de la toma de decisiones basándose en las características del problema en cuestión. Dicha recomendación requiere examinar las circunstancias bajo las cuales diferentes modelos MCDA arrojan resultados similares o diferentes. También se debe dar un manejo excepcional al desarrollo de recursos de software multiplataforma para brindar un mayor soporte al desarrollo de aplicaciones web y móviles. La información sobre los sistemas operativos compatibles indica que los recursos actuales están orientados, en gran medida, a aplicaciones de escritorio para Windows, Linux o Mac. Una investigación interesante y sugerida se centra en el uso de herramientas de desarrollo con buena compatibilidad y velocidad de ejecución aceptable. Varias de las herramientas más utilizadas no cumplen con estos elementos, lo que dificulta su integración en otros proyectos. En última instancia, es necesario enfatizar la importancia de construir sistemas de soporte de decisiones multicriterio que no se centren únicamente en el desarrollo de modelos, sino que también incorporen todas las futuras líneas de investigación mencionadas. Como resultado de esta investigación, existen muchas preguntas de investigación abiertas en torno a los recursos de software para la programación MDSS que parecen merecer la pena abordar desde una perspectiva académica o comercial.

Referencias

- Amazon. (2024a). *What is a Framework? - Framework in Programming and Engineering Explained* - AWS. Amazon Web Services, Inc. <https://aws.amazon.com/what-is/framework/>
- Amazon. (2024b). *What is SDK? - SDK Explained* - AWS. Amazon Web Services, Inc. <https://aws.amazon.com/what-is/sdk/>

- Baky, I. A. (2014). Interactive TOPSIS algorithms for solving multi-level non-linear multi-objective decision-making problems. *Applied Mathematical Modelling*, 38(4), 1417-1433. <https://doi.org/10.1016/j.apm.2013.08.016>
- Bigaret, S., Hodgett, R. E., Meyer, P., Mironova, T., & Olteanu, A.-L. (2017). Supporting the multi-criteria decision aiding process: R and the MCDA package. *EURO Journal on Decision Processes*, 5(1-4), 169-194. <https://doi.org/10.1007/s40070-017-0064-1>
- Bohanec, M. (2015). *DEXi: A Program for Multi-Attribute Decision Making*. <https://kt.ijs.si/MarkoBohanec/dexi.html>
- Boland, N., Charkhgard, H., & Savelsbergh, M. (2017). A new method for optimizing a linear function over the efficient set of a multiobjective integer program. *European Journal of Operational Research*, 260(3), 904-919. <https://doi.org/10.1016/j.ejor.2016.02.037>
- Bozorg-Haddad, O., Azarnivand, A., Hosseini-Moghari, S.-M., & Loáiciga, H. A. (2016). Development of a Comparative Multiple Criteria Framework for Ranking Pareto Optimal Solutions of a Multiobjective Reservoir Operation Problem. *Journal of Irrigation and Drainage Engineering*, 142(7). [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0001028](https://doi.org/10.1061/(ASCE)IR.1943-4774.0001028)
- Cabarcos, A., Almenarez Mendoza, P., Diaz Sanchez, F., & Lopez, D. M. (2018). FRiCS: A Framework for Risk-driven Cloud Selection. *2nd International Workshop on Multimedia Privacy and Security (MPS'18)*, 18-26. <https://doi.org/10.1145/3267357.3267362>
- Decision Deck Consortium. (2022). *Decision Deck*. Decision-Deck.org. <http://www.decision-deck.org/project/>
- Dybå, T., & Dingsøyr, T. (2008). Empirical Studies of Agile Software development: a Systematic Review. *Information and Software Technology*, 50(9-10), 833-859. <https://doi.org/10.1016/j.infsof.2008.01.006>
- Fabianek, P., Will, C., Wolff, S., & Madlener, R. (2020). Green and regional? A multi-criteria assessment framework for the provision of green electricity for electric vehicles in Germany. *Transportation Research Part D: Transport and Environment*, 87, 102504. <https://doi.org/10.1016/j.trd.2020.102504>
- Figueroa Pérez, J. F., Lopez, J. C., Noriega, A. D., & Contreras, E. O. (2022a). Software Resources for Developing Multicriteria Applications: A Review, Part 1: Introduction and types of software resources developed. *Proceedings of MOL2NET'22, Conference on Molecular, Biomedical & Computational Sciences and Engineering, 8th Ed. - MOL2NET: FROM MOLECULES to NETWORKS*, 13875. <https://doi.org/10.3390/mol2net-08-13875>
- Figueroa Pérez, J. F., Lopez, J. C., Noriega, A. D., & Contreras, E. O. (2022b). Software Resources for Developing Multicriteria Applications: A Review, Part 2: multicriteria methods implemented. *Proceedings of MOL2NET'22, Conference on Molecular, Biomedical & Computational Sciences and Engineering, 8th Ed. - MOL2NET: FROM MOLECULES to NETWORKS*, 13877. <https://doi.org/10.3390/mol2net-08-13877>
- Figueroa Pérez, J. F., Lopez, J. C., Noriega, A. D., & Contreras, E. O. (2022c). Software Resources for Developing Multicriteria Applications: A Review, Part 3: operating systems supported. *Proceedings of MOL2NET'22, Conference on Molecular, Biomedical*

- & Computational Sciences and Engineering, 8th Ed. - MOL2NET: FROM MOLECULES to NETWORKS, 13876. <https://doi.org/10.3390/mol2net-08-13876>
- Figueroa-Perez, J. F., Leyva-Lopez, J. C., Pérez-Contreras, E. O., Sánchez, P. J., & Ramirez-Noriega, A. D. (2020). An Agent-Based System for the Design of New Products Using a Fuzzy Multicriteria Approach. *International Journal of Fuzzy Systems*, 22(8), 2691-2707. <https://doi.org/10.1007/s40815-020-00934-6>
- Formal Concept Analysis Research Toolbox (FCART. (s.f.).
- Geeks for Geeks. (2024, 26 de febrero). *What is Software Package?* GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-software-package/>
- GeeksforGeeks. (2022, 25 de abril). *How to write Test Cases - Software Testing.* <https://www.geeksforgeeks.org/test-case/>
- Gomes, C., Parente, M., Azenha, M., & Lino, J. C. (2018). An integrated framework for multi-criteria optimization of thin concrete shells at early design stages. *Advanced Engineering Informatics*, 38, 330-342. <https://doi.org/10.1016/j.aei.2018.08.003>
- Goodwin, M. (2024, abril). *What is an Application Programming Interface (API).* <https://www.ibm.com/topics/api>
- Gunther, C., Tammer, C., Hillmann, M., & Winkler, B. (2015). *Project FLO.* <https://project-flo.de/>
- Halwai, S. (2021, 18 de junio). *What is Application Software: Examples & Types.* OpenXcell. <https://www.openxcell.com/blog/application-software/>
- Hämäläinen, R. P. (2013). DECISIONARIUM. <http://decisionarium.tkk.fi/>
- Hanine, M., Boutkhoum, O., Tikniouine, A., & Agouti, T. (2016). A new web-based framework development for fuzzy multi-criteria group decision-making. *SpringerPlus*, 5(1). <https://doi.org/10.1186/s40064-016-2198-1>
- IBM Corporation. (2024, 16 de diciembre). IBM Integration Bus for z/OS 10.1. <https://www.ibm.com/docs/es/integration-bus/10.1?topic=services-what-is-web-service>
- Ighravwe, D. E., Babatunde, M. O., Cornelius, M. T., Aikhuele, D., & Akinyele, D. (2021). A MCDM-based framework for the selection of renewable energy system simulation tool for teaching and learning at university level. *Environment, Development and Sustainability*, 24(11), 13035-13056. <https://doi.org/10.1007/s10668021019811>
- Jablonsky, J. (2014). MS Excel based Software Support Tools for Decision Problems with Multiple Criteria. *Procedia Economics and Finance*, 12, 251-258. [https://doi.org/10.1016/S2212-5671\(14\)00342-6](https://doi.org/10.1016/S2212-5671(14)00342-6)
- Karami, A., & Guo, Z. (2012). A Fuzzy Logic Multi-criteria Decision Framework for Selecting IT Service Providers. <https://doi.org/10.1109/hicss.2012.59>
- Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews.* Chapman & Hall/CRC.
- Kizielewicz, B., Shekhovtsov, A., & Sałabun, W. (2023). pymcdm – The universal library for solving multi-criteria decision-making problems. *SoftwareX*, 22, 101368. <https://doi.org/10.1016/j.softx.2023.101368>
- Labella, Á., & Martínez, L. (2019). FLINTSTONES 2.0 an Open and Comprehensive Fuzzy Tool for Multi-criteria Decision Analysis. En *Advances in Intelligent Systems and Computing* (pp. 762-769). https://doi.org/10.1007/978-3-030-23756-1_91

- Leyva López, J. C., Fernández González, E., & Trejos Alvarado, M. (2008). Special Issue on Multicriteria Decision Support Systems. *Computación y Sistemas*. https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462008000400001
- Löhne, A., & Weißing, B. (2017). The vector linear program solver Bensolve – notes on theoretical background. *European Journal of Operational Research*, 260(3), 807-813. <https://doi.org/10.1016/j.ejor.2016.02.039>
- Lutkevich, B. (2022). *What is Software Documentation? Definition, Types and Examples*. <https://www.techtarget.com/searchsoftwarequality/definition/documentation>
- MAD Preference. (2018). DecideIT | Preference AB. <https://www.preference.nu/decideit/>
- MCDALab. (2014). MEM: Multiplex Electionis Methodus. <https://github.com/MCDALab/MEM>
- MCDM Society. (2025, enero). *Software - Multiple Criteria Decision Making*. <https://www.mcdmsociety.org/2025/01/12/software/>
- Meyer, P., & Bigaret, S. (2012). Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA. *Intelligent Decision Technologies*, 6(4), 283-296. <https://doi.org/10.3233/IDT-2012-0144>
- Misitano, G., Saini, B. S., Afsar, B., Shavazipour, B., & Miettinen, K. (2021). DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization. *IEEE Access*, 9, 148277-148295. <https://doi.org/10.1109/access.2021.3123825>
- Muruganantham, A., & Gandhi, G. M. (2019). Framework for Social Media Analytics based on Multi-Criteria Decision Making (MCDM) model. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-019-7470-2>
- Nevagi, A. (2021). LinkedIn. <https://www.linkedin.com/pulse/what-toolkit-abhay-nevagi-1e>
- Omar, M., & Soltan, H. (2020). A framework for welding process selection. *SN Applied Sciences*, 2(3). <https://doi.org/10.1007/s42452-020-2144-2>
- Ozsahin, I., Ozsahin, D. U., & Uzun, B. (2021). *Applications of Multi-Criteria Decision-Making Theories in Healthcare and Biomedical Engineering*. Elsevier. <https://doi.org/10.1016/c2020-0-01611-7>
- Radulovic, F., García Castro, R., & Gómez-Pérez, A. (2013). A Recommendation Framework Based on the Analytic Network Process and its Application in the Semantic Technology Domain. *CAEPIA 2013: XV Conference of the Spanish Association for Artificial Intelligence*. <https://oa.upm.es/20660/1.haslightboxThumbnailVersion/Paper.pdf>
- Sametinger, J. (1997). *Software Engineering with Reusable Components*. Springer Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-03345-6>
- Sharma, H. K., Roy, J., Kar, S., & Prentkovskis, O. (2018). Multi Criteria Evaluation Framework for Prioritizing Indian Railway Stations Using Modified Rough AHP-Mabac Method. *Transport and Telecommunication Journal*, 19(2), 113-127. <https://doi.org/10.2478/ttj-2018-0010>
- Siraj, S., Mikhailov, L., & Keane, J. A. (2015). PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments. *International Transactions in Operational Research*, 22(2), 217-235. <https://doi.org/10.1111/itor.12054>

- Siskos, Y., & Spyridakos, A. (1999). Intelligent multicriteria decision support: Overview and perspectives. *European Journal of Operational Research*, 113(2), 236-246. [https://doi.org/10.1016/s0377-2217\(98\)00213-6](https://doi.org/10.1016/s0377-2217(98)00213-6)
- Source Making. (2019). *Design Patterns and Refactoring*. https://sourcemaking.com/design_patterns
- Sterling, T., Anderson, M., & Brodowicz, M. (2018). Libraries. En *Elsevier EBooks* (pp. 313-345). <https://doi.org/10.1016/b978-0-12-420158-3.00010-1>
- Szyperski, C. (2003). *Component Software: Beyond Object-Oriented Programming* (2.^a ed.). Pearson Education India.
- Toffano, F., Garraffa, M., Lin, Y., Prestwich, S., Simonis, H., & Wilson, N. (2022). A multi-objective supplier selection framework based on user-preferences. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-021-04251-5>
- Upadhyay, S. (2022, noviembre). *What Is An Algorithm? Characteristics, Types and How to write it*. <https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm>
- Więckowski, J., & Sałabun, W. (2024). Version [1.1] - pyFDM: A python library for uncertainty decision analysis methods. *SoftwareX*, 25, 101607. <https://doi.org/10.1016/j.softx.2023.101607>
- Wille, R. (2005). Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. En *Lecture Notes in Computer Science* (pp. 1-33). https://doi.org/10.1007/11528784_1
- Yang, D., Li, X., Jiao, R. J., & Wang, B. (2018). Decision support to product configuration considering component replenishment uncertainty: A stochastic programming approach. *Decision Support Systems*, 105, 108-118. <https://doi.org/10.1016/j.dss.2017.11.004>
- Yatsalo, B., Didenko, V., Gritsyuk, S., & Sullivan, T. (2015). Decerns: A Framework for Multi-Criteria Decision Analysis. *International Journal of Computational Intelligence Systems*, 8(3), 467. <https://doi.org/10.1080/18756891.2015.1023586>
- Zhao, J., Xie, B., Wang, Y., & Xu, Y. (2010). TSRR: A software resource repository for trustworthiness resource management and reuse.
- Zhou, Y. (2015). *Multi-criteria decision making in software development: A systematic literature review* [Thesis]. University of Oulu. <https://oulu.repo.oulu.fi/bitstream/handle/10024/41182/nbnfioulu-201508271929.pdf?sequence=1>