

Agile Software Projects in Engineering Degree Projects

Edgar A. López G.¹ and Gustavo A. Rivera S.¹

¹Universidad Cooperativa de Colombia, Bogotá D.C., edgar.lopez@campusucc.edu.co, gustavorivera11@hotmail.com

Abstract- In areas of degree work in higher education, the proposed software projects demand the use of the so-called agile methodologies, which are based on the principles set forth in the Agile Manifesto and the disciplinary theoretical support of Engineering Engineering. Software. Methodologies characterized by iterative and incremental development, supported by heuristics, of great acceptability in the dynamics of current software development due to efficient time management; When adopted in degree works, where academic and administrative commitments of particular connotation are met, difficulties are evident in the methodological characterization, due to the business-oriented precepts of universally recognized agile methodologies. In this sense, the MaTraGra methodology is conceived as a solution to this problem.

Keywords- Software Engineering, MaTraGra, agile methodology, software project and degree work.

Proyectos Ágiles de Software en Trabajos de Grado de Ingeniería

Edgar A. López G.¹ y Gustavo A. Rivera S.¹

¹Universidad Cooperativa de Colombia, Bogotá D.C., edgar.lopez@campusucc.edu.co, gustavorivera11@hotmail.com

Resumen- *En espacios de los trabajos de grado en educación superior, los proyectos de software planteados demandan, de la utilización de las denominadas metodologías ágiles, las cuales se sustentan en los principios expuestos en el Manifiesto Ágil y el apoyo teórico disciplinar de la Ingeniería de Software. Metodologías caracterizadas por el desarrollo iterativo e incremental, apoyadas en la heurística, de gran aceptabilidad en la dinámica del desarrollo de software actual por el eficiente manejo del tiempo; al ser adoptadas en los trabajos de grado, en donde se cumplen compromisos académicos y administrativos de connotación particular, se evidencian dificultades en la caracterización metodológica, por los preceptos de orientación empresarial de las metodologías ágiles de reconocimiento universal. En este sentido, la metodología MaTraGra es concebida como solución a esta problemática.*

Palabras clave- *Ingeniería de Software, MaTraGra, metodología ágil, proyecto de software y trabajo de grado.*

I. INTRODUCCIÓN

Los entes estatales orientadores del sector educativo, establecen criterios de construcción curricular acogidos por las Instituciones de Educación Superior -IES- que, dentro de la autonomía universitaria, reglamentan el trabajo de grado como componente final del plan de estudios para evidenciar las competencias de formación profesional integral, en cumplimiento de requisitos académicos y administrativos para culminar estudios.

Una modalidad del trabajo de grado inmerso en el plan de estudios de Ingeniería, es apropiada con el fin de solucionar problemas a través de proyectos de software; momento en el cual se plantea la utilización de una metodología para el desarrollo de software, dentro del abanico de metodologías pregonadas por la Ingeniería de Software -IS- y de aceptación en el ámbito empresarial global.

Respecto a los métodos o metodologías para el desarrollo de software sustentadas en la IS, son regularmente clasificadas en tres categorías: tradicionales, ágiles e híbridas. Las primeras, identificadas por la rigurosidad metódica, acudiendo a lineamientos secuenciales de cumplimiento de etapas acorde a una rigurosa planeación y orientación procedimental a los productos [1].

Las metodologías ágiles, se caracterizan por promover la flexibilidad procedimental; el desarrollo incremental e iterativo de productos; participación continua del cliente; etapas de desarrollo cortas; en general, se orientan a la heurística afianzada en el recurso humano. Estas

metodologías son de gran acogida y reconocimiento mundial para desarrollo de software [2]. Finalmente, las metodologías híbridas, permiten trabajar las prácticas de las dos corrientes metodológicas tradicional e híbrida.

En los proyectos de desarrollo de software adscritos a los trabajos de grado ingenieriles, los estudiantes adoptan metodologías ágiles consecuentes con su caracterización y de acogimiento universal, presentando inconvenientes al acoplar, ha espacios académicos, una metodología orientada a ambientes empresariales; específicamente, en el dilema de la designación de roles, gestión de recursos e incumplimiento de compromisos académicos, problemática evidenciada en la sustentación ingenieril del proyecto [3].

Consecuente con lo anteriormente planteado, se propuso y se desarrolló la metodología ágil MaTraGra, resultado de una investigación en formación doctoral, orientada al desarrollo de proyectos de software dentro de los trabajos de grado ingenieriles. Metodología que pregona como ventajas: flexibilidad, desarrollo incremental y procesos iterativos; estadios colaborativos de evaluación ingenieril; documentación concurrente; y asignación de roles acordes a ambientes académicos.

Este documento presenta, la fundamentación teórica y conceptual de las metodologías ágiles que apoyan la concepción de la metodología MáTraGra, componentes metódicos y estructurales, dinámica procedimental, enfoque de aplicación y la evaluación ingenieril de la metodología, aunada, con la experimentación de la metodología en estudio de caso en entornos académicos en educación superior y los resultados obtenidos.

II. PROYECTOS DE DESARROLLO DE SOFTWARE EN TRABAJOS DE GRADO

Para la gestión y desarrollo de proyectos de software, la IS cataloga tres aspectos fundamentales: el contexto para la gestión y control del proyecto; en segundo lugar, los métodos alineados al ciclo de vida del software; y finalmente, las técnicas y herramientas soportadas en las Tecnologías de Información -TI-, que aplican a los dos anteriores.

Las metodologías para el desarrollo de software representan una forma de proceder dentro de un contexto de aplicación, para dar cumplimiento a los objetivos planteados en un proyecto [4]. Para Tinoco, et al. [5], su evolución se da, de acuerdo con las expectativas y exigencias de las dinámicas evolutivas de las TI. Suelen clasificarse desde el ámbito ingenieril y su perspectiva de aplicación: en tradicionales, ágiles e híbridas.

Respecto a la necesidad, de que el ciclo de vida del software se soporte en modelos o metodologías, Bauer [6], lo justifica en relación con la rentabilidad y fiabilidad; Boehm [7], lo califica como una forma óptima de utilizar el conocimiento científico. Lo que sustenta que, los proyectos para el desarrollo de sistemas de software participan del ciclo de vida del software.

Una metodología ágil para el desarrollo de software, según Sommerville [8], se encuentra planteada a partir de *modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos*, participando de un enfoque estructurado. Asimismo, debe responder a un contexto de aplicación desde la perspectiva de productividad y competitividad.

Al calificar las metodologías tradicionales, en ellas se identifican las siguientes características: el enfoque a los productos, control exhaustivo a los procesos, y composición en etapas o fases de cumplimiento en secuencia, como respuesta a una planeación compleja y estricta [9]; metodologías que han recibido críticas aduciendo la aplicabilidad utópica por usuarios hipotéticos [10].

Las metodologías ágiles se fundamentan en heurísticas expuesta por el recurso humano, agilidad asociada a los tiempos de respuesta, flexibilidad en los procesos y adaptabilidad a ambientes particulares [11]; propician la productividad del equipo de trabajo, la incorporación del cliente, sin hacer de la arquitectura del software un componente rígido; adheridas a los postulados del Manifiesto Ágil [12], de coautoría de Beck [13], constituido en el pilar histórico de mayor relevancia.

Según estudios [2], las razones para la mayor demanda de las metodologías ágiles son: la gestión a prioridades cambiantes, mejoramiento de tiempos de entrega de productos, productividad del equipo de trabajo, alineación con las TI y calidad de software. El abanico de metodologías ágiles abona el camino para su estudio y concepción, al respecto Hawrish [14], propone la selección de la metodología de acuerdo a la naturaleza del proyecto.

Las metodologías híbridas, acogen los mejores preceptos de las tradicionales y ágiles, incorporando cierto grado de flexibilidad en el proceso, atendiendo la disciplina de cumplimiento de planificación del trabajo en respuesta a limitantes externas de tiempo y recursos económicos [15]; adoptadas en consideración a sus bondades procedimentales o metódicas [16].

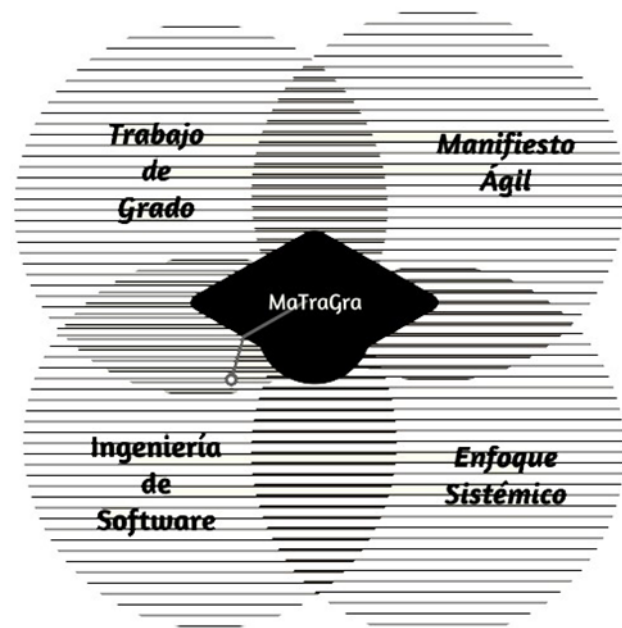
El trabajo de grado se plantea, como un componente terminal en los programas académicos de las IES, las cuales, haciendo uso de la autonomía universitaria proferida por el estado, establecen lineamientos reglamentarios abordados por los estudiantes en su aptitud discrecional. Trabajo de grado, entendido como una práctica académica integral para evidenciar y valorar las competencias profesionales disciplinares, referido por Torres [17], como el resultado de

un proceso creativo e innovador, objetivo, reflexivo controlado y crítico.

Definido por Sáez [18], como Software Mental, el Enfoque Sistémico es direccionado a la identificación estructural del objeto de estudio, la calidad del diseño y la concepción de metodologías de sistemas; para Petrella [19], es referente en la construcción de modelos administrativos graduales y evolutivos. Asimismo, propone la dimensión sistémica para abordar problemáticas en espacios de tiempo definidos y proyectar soluciones.

III. METODOLOGÍA ÁGIL MATRAGRA

Orientada al desarrollo de proyectos de software en el marco de trabajos de grado, justificada por la problemática advertida por los estudiantes, al asimilar metodologías ágiles de reconocimiento global perfiladas al mundo empresarial, que al ser aplicadas en espacios académicos, se participa de dificultades respecto a: espacios de trabajo y asesoramiento; adquisición y administración de recursos; y tiempos de cumplimiento; adheridos a la poca experticia de los estudiantes, en ejercicios profesionales de connotación evaluada.



Fuente: elaboración propia

Canva.com

Figura 1. Referentes de MaTraGra.

Expuesto por Pressman, et al. [20], una metodología se concibe sustentada en dos categorías: el aspecto filosófico, en relación a la afinidad a las metodologías ágiles y, en segundo lugar, los lineamientos sobre el desarrollo de proyectos de software, enmarcados disciplinalmente por la IS respecto al cumplimiento en proponer procesos de análisis, diseño, comunicación continua y retroalimentada. En consecuencia, en la metodología MaTraGra confluyen cuatro saberes referenciales de apoyo conceptual -Figura 1-:

- A. *Trabajo de grado*: determina el perfil de aplicación u orientación de la metodología.

- B. *Manifiesto Ágil*: provee los principios constitutivos de las metodologías ágiles.
- C. *Ingeniería de Software*: sustenta los componentes metódicos y metodológicos para abordar proyectos de software.
- D. *Enfoque Sistémico*: representa el espectro integral contextual.

Referido por Avison, et al. [21], una metodología se constituye de procedimientos, técnicas, herramientas y documentos; se estructura mediante fases y sub-fases, abarcando la planificación, gestión, control y evaluación. Además, se encuentra dispuesta y organizada en etapas apropiadamente definidas [22]. A continuación, se presentan los componentes de la metodología MaTraGra.

ROLES

El desarrollo ágil de software promueve el trabajo en equipo colaborativo [23], grupos de desarrollo pequeños y estables [8]; consecuente con lo citado, el recurso humano pregonado por la metodología proferida, se plantea acorde a las disposiciones de los trabajos de grado. En consecuencia, designa dos roles:

- A. *Estudiante*: responde por el desarrollo del proyecto de software en cumplimiento de los compromisos adquiridos en el anteproyecto.

- B. *Director*: cumple el rol de asesor y validador, certificando la calidad ingenieril del proceso y productos.

DOCUMENTACIÓN

Cobra relevancia para propósitos de revisión y corrección en el desarrollo del proyecto; se enfoca a facilitar la usabilidad de los productos [20]; y para el registro de actividades y administración de la agenda de compromisos [24]. En el desarrollo ágil de software por iteraciones o fases, los documentos formales contribuyen a procesos de comunicación en cumplimiento de requisitos [8]. Se disponen dos componentes de documentación concurrentes al desarrollo del trabajo de grado:

- A. *Registro de actividades*: se constituye en agenda y registro de actividades y compromisos planificados por el director o tutor y los estudiantes. El director o tutor es garante de su integridad.
- B. *Documento ingenieril*: fundamenta el soporte sustentable del ejercicio académico dentro del trabajo de grado, en observancia de los reglamentos o normas dispuestas por la institución educativa. Bajo la responsabilidad del estudiante y la validación del director o tutor.

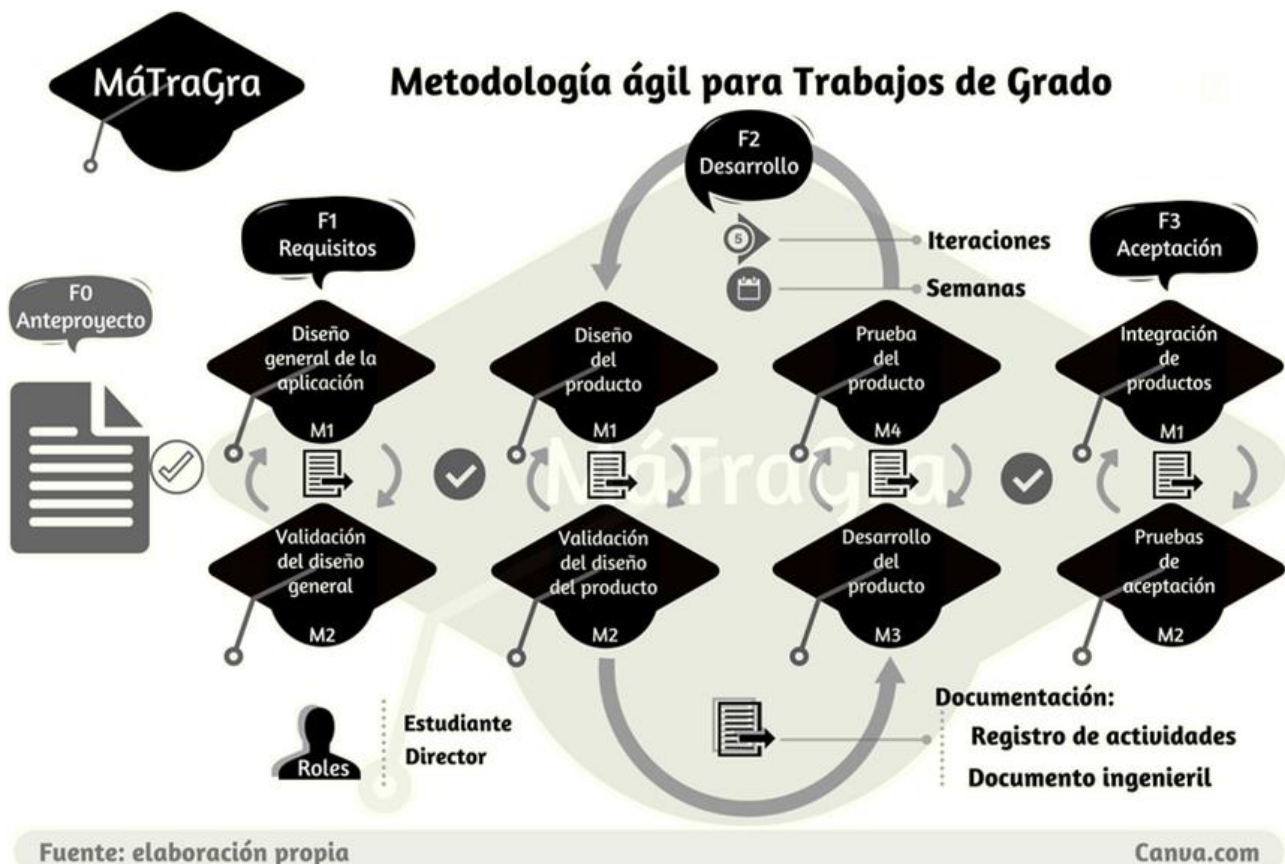


Figura 2. Metodología MaTraGra.

ESTRUCTURA Y PROCESOS DE MATRAGRA

Dispone de fases supeditadas a la iteración, desarrollo incrementa y flexibilidad, determinadas por los momentos de validaciones -Figura 2-:

- A. *Fase 0, Anteproyecto:* donde se concibe el proyecto de acuerdo a la reglamentación institucional. Se constituye en prerrequisito fundamental para abordar el desarrollo de la metodología.
- B. *Fase 1, Requisitos:* propone la identificación de requisitos para determinar el comportamiento, propiedades y restricciones del software, direccionado implícitamente a la calidad. Conformado por dos momentos iterativos: diseño general de la aplicación y validación del diseño general.
- C. *Fase 2, Desarrollo:* participa del desarrollo iterativo e incremental de productos, contempla la flexibilidad respecto a la redefinición de requerimientos. Compuesta de cuatro momentos: diseño del producto, validación del diseño del producto, desarrollo del producto y prueba del producto.
- D. *Fase 3, Aceptación:* refiere a la validación para la consolidación integral del sistema de software y los soportes. Compuesta de dos momentos: integración de productos y pruebas de aceptación.

Para cualificar un desarrollo de software es pertinente orientarlo a las personas [20], en consecuencia, se proponen como estrategias metódicas las siguientes: trabajo colaborativo, asignación de responsabilidades, retroalimentación frecuente, documentación concurrente, bitácora de procesos y actividades, y pruebas programáticas.

Adicionalmente, en apoyo al desarrollo sistemático, se presenta la aplicación TraGraapp, elaborada para plataformas de dispositivos móviles, con el fin de administrar los procesos y actividades inmersos en las fases y momentos.

VALIDACIÓN DE LA METODOLOGÍA

La metodología MaTraGra fue experimentada en estudio de casos, en el desarrollo de proyectos de software en el marco de trabajos de grado del programa académico de Ingeniería de Sistemas, al indagar sobre la percepción de su aplicación, los estudiantes y directores de los proyectos vinculados al estudio, calificaron positivamente la experiencia ingenieril de la metodología concebida, catalogándola en el orden de excelente, frente a la adopción de otras metodologías ágiles; valoración representada en la calificación expuesta en la -Figura 3-;

Fue sometida al análisis de afinidad con las metodologías ágiles Scrum y XP, mediante la Herramienta Dimensional Analítica 4-DAT, apoyados por lo referido en [25], obteniendo una calificación positiva; asimismo, expuesta al Modelo de Aceptación de Tecnología (TAM) [26], diagnosticado en el estudio de caso, advirtiendo una mayor adaptabilidad.

Se reseña la aplicación Web denominada Matragra, desarrollada por estudiantes en el marco de su trabajo de grado ingenieril, mediante la utilización de los preceptos ágiles pregonados por la metodología presentada; orientada a ser utilizada como herramienta para apoyar la administración de los procesos, actividades y artefactos proferidos por la metodología MaTraGra [28].

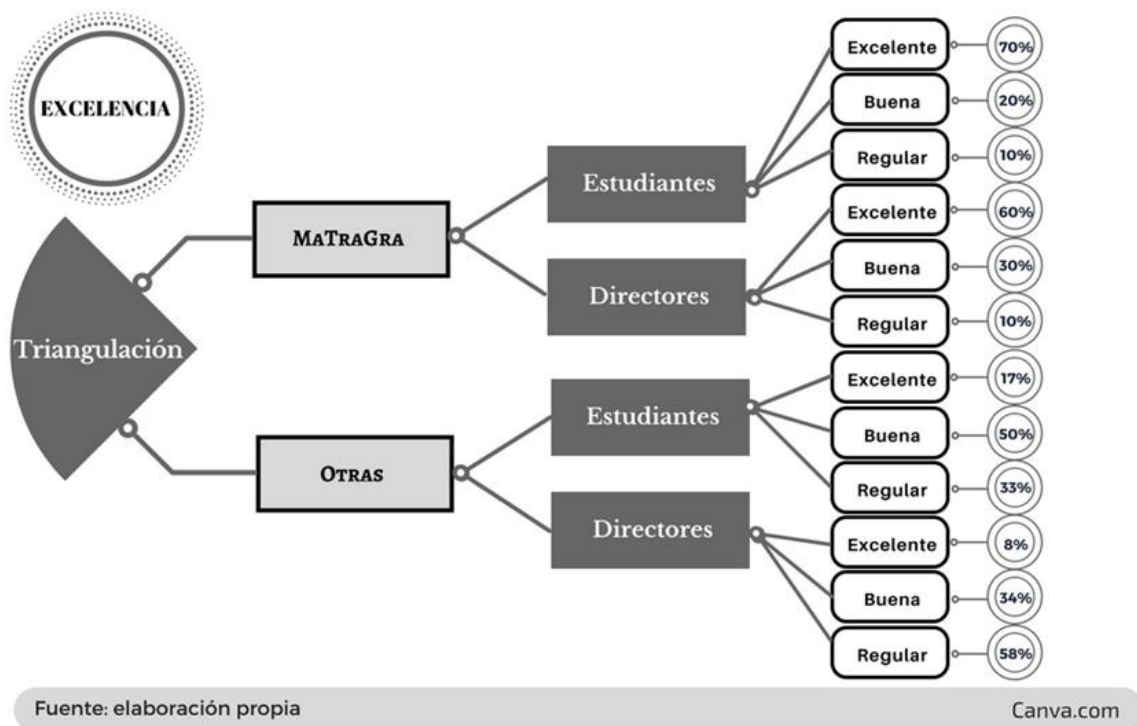


Figura 3. Evaluación de MaTraGra.

IV. CONCLUSIONES

La metodología ágil MaTraGra se propone, en su orientación, para ser seleccionada y abordada en el desarrollo de proyectos de sistemas de software propuestos en los trabajos de grado en ambientes académicos de IES, en donde se dificulta la aplicación de las metodologías ágiles de reconocimiento universal; participando del proceso de formación y sustentación de competencias profesional.

Son características relevantes de la metodología MaTraGra: flexibilidad metódica y desarrollo incremental de productos en procesos iterativos con ciclos cortos de trabajo, estadios de validación colaborativos que promueven la retroalimentación, equipos de trabajo pequeños y cohesionados, documentación concurrente, y participa del cumplimiento de las disposiciones normativas reglamentarias institucionales sobre trabajos de grado.

Consecuentes con el sometimiento de la metodología MaTraGra a herramientas de valoración de su agilidad, se puede conceptuar de manera concluyente, su afinidad con los principios pregonados en el *Manifiesto Ágil* y universalmente aceptados por la corriente del desarrollo ágil de sistemas de software.

Asimismo, la experimentación en espacios académicos a través de planteamientos de estudio de caso y apoyados en la percepción de los actores involucrados, se estableció la calificación de excelencia en la aplicabilidad metódica y alineación con la formación académica y la calidad de productos de software. Se relacionan como ventajas respecto a las metodologías ágiles de reconocimiento universal, los siguientes aspectos:

- A. La adaptabilidad a espacios académicos, participando de la formación del profesional pregonado por los objetivos curriculares de los programas de educación superior.
- B. Determinación y asignación de roles acordes a la disposición del recurso humano contemplado en los trabajos de grado.
- C. Componentes procedimentales de flexibilidad relevante, bajo la perspectiva de aprendizaje colaborativo y refactorización del diseño ingenieril.
- D. Espacios de valoración procedimental y de productos con fines de revisión crítica retroalimentada por el asesor o tutor.
- E. Disposición a la utilización de artefactos direccionados a la gestión del proyecto y al cumplimiento de las normas institucionales; refiérase al registro de actividades y la documentación concurrente.

La metodología ágil referida, fue desarrollada y experimentada en el marco de una investigación académica planteada en un doctorado en educación, sustentada, socializada y documentada [27].

Acorde a la dinámica de las metodologías ágiles, se advierte la disposición evolutiva de la metodología presentada, hacia

la gestión de proyectos, sin perder el perfil de adherencia metódica pregonada y apoyada por la IS, la orientación a espacios de formación académica, los preceptos de calidad adheridos al desarrollo de proyectos de sistemas de software, siempre en procura del conocimiento en el campo de la TI al servicio sostenible de la humanidad.

REFERENCIAS

- [1] E. Orejuela, T. Restrepo, D. Rojas y A. Abucha, “El traje nuevo del Emperador: la metodología de desarrollo SCRUM. ¿Por qué falla en proyectos de software?”, Avenir, 2019.
- [2] Digital.ai, “15th Annual State of Agile Report. Agile adoption accelerates across the Enterprise”, <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>, 2021.
- [3] S. Chalela, G. Valencia y M. Cadavid, “Psycho-social and familial factors influencing drop-out rates among university students in the context of developing countries”, *Revista Lasallista de Investigación*, vol. 17, 2020.
- [4] J. Candejas, “Implementación del Modelo Integral Colaborativo (MDSIC)”, México: Universidad Popular Autónoma del Estado de Puebla, 2014.
- [5] O. Tinoco, P. Rosales y J. Salas, “Criterio de selección de metodologías de desarrollo de software”, *España y Portugal: Red de revistas científicas de América Latina y el Caribe*, 2010.
- [6] L. Bauer, “Software Engineering, Information Processing”, 71, Amsterdam: North Holland Publishing Co, 1972.
- [7] B. Boehm, “Anchoring the Software Process”, *IEEE Software*, vol. 13, núm. 4, <https://ieeexplore.ieee.org/document/526834>, 1976.
- [8] I. Sommerville, “Ingeniería del software”, 9ª Edición, España: Pearson editores, 2011.
- [9] H. Janampa, K. Sotomayor y M. Prad, “Extreme programming software development model on scrum for agile software management”, *Revista Iberica de Sistemas e Tecnologías de Informacao – RISTI*, 2021.
- [10] D. Truex, R. Baskerville y J. Travis, “Methodical systems development: The deferred meaning of system development methods”, *Accounting, Management and Information Technology*, 2000.
- [11] S. Tridibesh, “A Guide to the Scrum Body of Knowledge”, Scrumstudy. Phoenix, Arizona, <http://doi.org/10.1017/CBO9781107415324.004>, 2016.
- [12] Manifiesto Ágil. “Manifiesto para el Desarrollo Ágil de Software”, <http://agilemanifesto.org/iso/es/manifesto.html>, 2001.

- [13] K. Beck, C. Andres, “Extreme Programming Explained: Embrace Change”, 2ª Ed., Boston: Addison Wesley Professional, 2004.
- [14] S. Hawrish, y J. Ruprecht, “Light Methodologies: It’s like Déjà Vu All over again”, Cutter IT Journal, 2002.
- [15] D. Alves, E. Costa., E. Días y M. Ferreira, “Application of a hybrid process software requirements management”, 11th Iberian Conference on Information Systems and Technologies (CISTI), Gran Canaria. (2016).
- [16] Mikarimin, “Metodología Híbrida de Desarrollo de Software Combinando XP y Scrum”, Revista Científica Multidisciplinaria, Universidad Regional Autónoma de los Andes UNIANDES - Sede Santo Domingo, 2019.
- [17] L. Torres, “Tesis de Maestría. Qué Hacer”, Bogotá: Universidad Autónoma de Colombia, 2013.
- [18] F. Sáez, “Complejidad y Tecnologías de Información”, Cuadernos de Tecnología y Sociedad n°3, Madrid: fundetel, http://dit.upm.es/~fsaez/intl/libro_complejidad.pdf, 2009.
- [19] C. Petrella, “Aportes del Enfoque Sistémico a la Comprensión de la Realidad. Avances del proyecto de investigación”, <https://docplayer.es/18981114-Avances-del-proyecto-de-investigacion-aportes-del-enfoque-sistemico-a-la-comprension-de-la-realidad-v02.html>, 2007.
- [20] S. Pressman y B. Maxim, “Software Engineering. A practitioner’s approach”, Eighth Edition, New York: McGraw Hill Education, 2015.
- [21] E. Avison, y G. Fitzgerald, “Information system development. Maidenhead”, Maidenhead: McGraw-Hill Education, 2006.
- [22] V. Chandra, “Comparison between Various Software Development Methodologies. International Journal of Computer Applications”, <https://pdfs.semanticscholar.org/e237/f9cb136f494c2bd0ce91525808c5c968b6b4.pdf>, 2015.
- [23] A. Cockbun, “Agile Software Development”, 2ª Edición, Madrid: Addison Wesley, 2007.
- [24] M. Piattini, y F. García, I. García y F. Pino, “Calidad de Sistemas de Información”, 3ª Edición ampliada y actualizada, Alfaomega, Ra-Ma, 2015.
- [25] A. Qumer y B. Henderson-Sellers, “An evaluation of the degree of agility in six agile methods and its applicability for method engineering”, In: Information and Software Technology, 2008.
- [26] S. Zaineldden, L. Hongbo, A. Koffi and B. Addallah, “Technology Acceptance Model' Concepts, Contribution, Limitation, and Adoption in Education”, School of Management, Jiangsu University, Jiangsu, China, School of Economics and Management, Beijing Forestry University, China, 2020.
- [27] G. Rivera, “Metodología ágil de desarrollo de software enfocada a trabajos de grado en Ingeniería”, URI: <https://hdl.handle.net/10901/23827>, 2021.
- [28] J. Mosquera y S. Prieto, “Aplicación Web para la administración del proceso de desarrollo en trabajos de grado elaborados bajo la metodología ágil Matragra”, Universidad Autónoma de Colombia, <https://github.com/SantiagoPri/Matragra>, 2021