



METODOLOGÍA ÁGIL ORIENTADA AL DESARROLLO DE PROYECTOS DE SOFTWARE EN TRABAJOS DE GRADO DE INGENIERÍA

Gustavo Armando Rivera Sánchez

**Universidad Cooperativa de Colombia
Colombia**

Resumen

La metodología referida, es el resultado de la investigación consecuente con problemáticas evidenciada en los trabajos de grado de los estudiantes de Ingeniería, quienes al abordar proyectos de software convienen en la selección de metodologías ágiles, inspirados por las bondades evidenciadas de la tendencia del mercado; encontrándose con la disyuntiva de trasegar espacios académicos con diferencias sustanciales, en relación con el ambiente empresarial a donde generalmente apuntan las metodologías.

El marco de concepción de la metodología MaTraGra proferida, se define metódicamente a partir de la Ingeniería de Software, afín a los preceptos de las metodologías ágiles y contextualizada en el Enfoque Sistémico; el perfil de aplicación se orienta al desarrollo proyectos de software inmersos en trabajos de grado. Se encuentra experimentada en estudios de caso en el desarrollo curricular de programas de Ingeniería, donde se advierten resultados exitosos.

La caracterización de la metodología, se constituye a partir del enfoque al espectro educativo del aprendizaje, promovido por la sencillez y direccionada a la apropiación y aplicación; consecuente con los componentes metódicos más relevantes: procesos flexibles e iterativos, desarrollo incremental de productos, estadios colaborativos, momentos de evaluación ingenieril retroalimentados, documentación concurrente, roles supeditados a los trabajos de grado y adaptabilidad a ambientes académicos.

Palabras clave: MaTraGra; proyecto de software; trabajo de grado

Abstract

The aforementioned methodology is the result of the consequent research with problems evidenced in the degree works of the engineering students, who when approaching software projects agree on the selection of agile methodologies, inspired by the evidences of the market trend; being faced with the dilemma of transferring academic spaces with substantial differences, in relation to the business environment to which the methodologies usually point.

The design framework of the MaTraGra methodology proffered, is methodically defined from the Software Engineering, related to the precepts of the agile methodologies and contextualized in the Systemic Approach; the application profile is oriented to the development of software projects immersed in degree projects. She is experienced in case studies in the curricular development of Engineering programs, where successful results are noticed.

The characterization of the methodology is based on the approach to the educational spectrum of learning, promoted by simplicity and directed to the appropriation and application; consistent with the most relevant methodical components: flexible and iterative processes, incremental development of products, collaborative stadiums, feedback moments of engineering evaluation, concurrent documentation, roles subordinated to degree works and adaptability to academic environments.

Keywords: *MaTraGra; software project; degree work*

1. Introducción

El trabajo de grado, es dispuesto por las Instituciones de Educación Superior -IES- para evidenciar y valorar las competencias profesionales disciplinares mediante una práctica académica integral, para el reconocimiento de saberes e idoneidad profesional. En los programas de Ingeniería, se plantean proyectos de software inscritos a los trabajos de grado, que demandan el trasegar por los preceptos sobre modelos y metodologías pertinentes, propiciando la decisión generalizada de acoger las metodologías ágiles, sobre las denominadas tradicionales e híbridas, sustentados en la caracterización y el dominio del mercado.

Los proyectos de software se emprenden en espacios académicos con particularidades relevantes consecuentes a la falta de experticia de los estudiantes en el abordaje del encargo ingenieril, aunado a la magnitud de la responsabilidad académica. Asimismo, se evidencian dificultades atinentes a la adopción de la metodología, relacionadas con la designación de roles, adquisición y administración de recursos, y manejo de tiempos en cumplimiento de los compromisos adquiridos; problemática no tipificada en ambientes empresariales, a donde se orientan las metodologías convencionales.

Como consecuencia, existe incertidumbre en el cumplimiento de compromisos, reflejándose en los indicadores de deserción universitaria o la dilatación del proceso; sustentando y justificando la constitución de la metodología MaTraGra, participante de lo visionado por los trabajos de grado, cimentada en el espectro promulgado y universalmente aceptado por las metodologías ágiles, con

la inspiración disciplinar patrocinada por la Ingeniería de Software -IS- y el ámbito contextual promovido por el Enfoque Sistémico.

Se suscribe la relevancia de la metodología MaTraGra, soportada en las características: desarrollo de productos de forma incremental y en procesos iterativos; determinación de roles consecuentes con el recurso humano participante en los trabajos de grado; patrocina el trabajo en equipo colaborativo; expone flexibilidad en los procesos dispuestos a replanteamientos; cataloga documentación en forma concurrente; y contempla momentos de evaluación y retroalimentación. Adicionalmente, se identifica metódicamente para ser direccionada acorde a cánones de aprendizaje atinentes a la facilidad de apropiación, comprensión y aplicación.

Se acentúa el propósito de este documento, en la presentación de la metodología MaTraGra desde diversos ámbitos: los aspectos que proveen la cimentación teórica y conceptual constitutiva; los componentes metódicos que participan de la estructura funcional y procedimental; los preceptos operativos que se involucran en las fases o momentos promulgados; los planteamientos resultantes de la experiencia investigativa alrededor de la metodología, a manera de conclusiones; finalmente, la disposición referencial de la conceptualización que soporta la fundamentación teórica y temática.

2. Fundamentos teóricos para la Metodología MaTraGra

Tomando lo expuesto por Pressman, et al. (2015): una metodología se respalda en dos categorías: aspectos filosóficos, que responden a la afinidad metódica; y los lineamientos para el desarrollo de proyectos de software. En MaTraGra confluyen cuatro saberes referenciales, que sustentan su concepción y orientación -Ver Figura 1-. El trabajo de grado, determina el perfil de aplicación; la Ingeniería de Software, aporta los insumos metodológicos disciplinares para el desarrollo de proyectos de software; el *Manifiesto Ágil*, provee los fundamentos de adherencia a las metodologías ágiles; y el Enfoque Sistémico, le otorga el espectro contextual e integral.

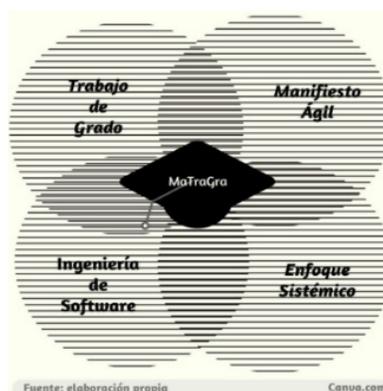


Figura 1. Referentes de MaTraGra.

El trabajo grado, entendido como un ejercicio académico demostrativo de apropiación, aplicación y prospección de saberes, para abordar y solucionar idealmente problemáticas afines al desempeño disciplinar, con el fin de valorar las competencias consecuentes con la formación

profesional integral, en un escenario de cumplimiento de la normatividad institucional. Referido por Torres (2012) como *un proceso creativo e innovador, objetivo, reflexivo controlado y crítico*, consecuente con una problemática social o enfocado a trascender las fronteras del conocimiento. Conforme a la aptitud discrecional del estudiante, opta por el abordaje de proyectos de software direccionados a la solución de problemas desde la perspectiva ingenieril; proyectos proferidos por el IEEE (1993) como *la aplicación de un enfoque sistemático, disciplinado y cuantificable* al proceso de desarrollo de software enmarcado por los postulados de la IS. Hecho causante del dilema, de estudiar y acoger los modelos y metodologías del espectro del mercado, presentados en tres categorías: metodologías ágiles, tradicionales e híbridas.

Aspectos diferenciadores de las metodologías ágiles sobre las tradicionales, son presentados por García, et al. (2015): sobrepone la heurística a los estándares sin responder al control riguroso, obtención de productos en menor tiempo, retroalimentación continua, expuesta a cambios recurrentes, diversidad de roles, participación activa del cliente, arquitectura del software no rígida y procesos de contratación flexibles. Sin desconocer lo expuesto por Alves, et al. (2016), respecto al surgimiento de una tercera alternativa híbrida, constituida por las mejores prácticas y artefactos de los dos paradigmas metódicos, donde se conjuga la flexibilidad con la propiedad de robustez. Las metodologías ágiles, responden, en su afinidad, a los principios paradigmáticos promovidos en el *Manifiesto Ágil* (Beck, et al., 2001): priorización a las personas sobre los productos, trabajo colaborativo y respuesta al cambio consecuente con la flexibilidad. Las razones para la mayor demanda, son referidas en el informe del estado de agilidad VersionOne (2018), consecuentes con los mejores resultados asociados a la productividad, calidad y cumplimiento; por la gestión al cambio de prioridades, visibilidad del proyecto, y alineadas a las Tecnologías de Información -TI- y el negocio. Han evolucionado, al recorrer caminos direccionados a la gestión de proyectos donde convienen las respuestas rápidas y el análisis predictivo para minimizar incertidumbre.

El Enfoque Sistemico, para Sáez (2009), *Software Mental acogido* en la identificación estructural del objeto de estudio, la calidad del diseño y la concepción de metodologías de sistemas; en Petrella (2007), es referente en la construcción de modelos administrativos graduales y evolutivos. Apoya a MaTraGra desde tres perspectivas: un componente holístico, al propiciar un marco metódico para abordar el objeto de estudio en la dimensión sistémica y alcanzar soluciones factibles; el componente heurístico, en la concepción evolutiva de la metodología en un espacio de tiempo, perfilando la aplicabilidad y funcionalidad; y, el componente ético y estético, responde a la razón, intencionalidad y proyección de los procedimientos y soluciones.

3. La metodología MaTraGra

Las metodologías para el desarrollo de software evolucionan en coherencia y exigencias de las TI, instando a mayores niveles de eficacia reflejada en la productividad, acogiendo técnicas, herramientas y demás artefactos para la gestión y desarrollo del software (Tinoco, et al. 2010).

3.1. Roles

El desarrollo ágil de software se centra en las personas (Cockburn, 2007); en grupos de desarrollo pequeños y estables (Somerville, 2011). El recurso humano pregonado por la metodología responde a la caracterización de los trabajos de grado, tipificados en dos roles:

- **Estudiante:** asume la responsabilidad del desarrollo del proyecto de software, en cumplimiento de los compromisos adquiridos en el anteproyecto.
- **Director:** participa del rol de asesor, validador del proceso y los productos, certificando la calidad de los mismos.

3.2. Documentación

En Pressman, et al. (2015), la documentación cobra importancia para propósitos de revisión y corrección, y en apoyo de la usabilidad del sistema; asimismo, para la administración de la agenda y registro de actividades, según Piattini, et al. (2015). En el desarrollo ágil de software por iteraciones o faces, para Somerville (2011), los documentos formales cobran relevancia en los procesos de comunicación en cumplimiento de requerimientos. La metodología propone dos categorías de documentos de elaboración concurrente al desarrollo del proyecto de software:

- **Registro de actividades:** se constituye en agenda para el registro de compromisos y el cumplimiento de los mismos. De responsabilidad del director.
- **Documento ingenieril:** soporte sustentable del ejercicio académico ingenieril en observancia de lo reglamentado por la institución. De responsabilidad en su elaboración del estudiante y validación del director.

3.3. Procesos estructurales de MaTraGra

Una metodología se encuentra estructurada en etapas apropiadamente definidas (Chandra, 2015). Se compone de fases que permiten la iteración y apropiada flexibilidad, determinada por los momentos de validaciones -Ver Figura 2-.



Figura 2. Metodología MaTraGra.

Fase 0, Anteproyecto: donde se concibe el proyecto de software, prerequisite fundamental para abordar la metodología. INCONTEC (2017) indica que, *el documento en el que se identifica y precisa la idea que constituye el núcleo del problema del trabajo de grado.*

Fase 1, Requisitos: donde promulga la identificación de requisitos que caracterizan la calidad de los productos (Madeja, 2013); para determinar el comportamiento, propiedades y restricciones del software a construir (Wieggers, 2013).

- **Momento 1, Diseño general de la aplicación:** propone el diseño ágil de la arquitectura general del sistema (Somerville, 2011), de los datos y las pruebas; especificaciones de la plataforma y herramientas de desarrollo. Asimismo, la planificación funcional y operativa del trabajo. Momento documentado concurrentemente; de responsabilidad del estudiante.
- **Momento 2, Validación del diseño general:** se refiere a la evaluación ingenieril de la viabilidad y consistencia del diseño planteado, produciendo iteración retroalimentada de los momentos. Momento bajo la responsabilidad del director con la participación del estudiante; pertinentemente documentado.

Fase 2, Desarrollo: promueve el desarrollo iterativo e incremental del software consecuente con la fase anterior. Estructurado en ciclos de trabajo -se recomienda cinco ciclos en promedio-, donde se contempla la redefinición de requerimientos a nivel incremental o por componentes (Somerville, 2011); estimando menos costos que al acometerlos al final (Kendall, et al. 2011).

- **Momento 1, Diseño del producto:** se nutre del análisis de los requerimientos específicos del producto, procediendo al diseño, determinación del alcance, especificación funcional y planificación de pruebas. Proceso documentado de responsabilidad del estudiante.
- **Momento 2, Validación del diseño del producto:** para el aseguramiento de cumplimiento de las especificaciones, desde la aplicación de las pruebas planificadas. La validación y retroalimentación puede producir la iteración con el momento anterior. De responsabilidad del director y el concurso del estudiante; debidamente documentado.
- **Momento 3, Desarrollo del producto:** construcción de componentes de software afín al diseño concebido, participan del momento las pruebas unitarias o de componentes. El estudiante es el responsable de las actividades del momento y documentación.
- **Momento 4, Prueba del producto:** acogiendo las especificaciones de diseño, en aras de verificar el cumplimiento de requisitos y aseguramiento de calidad definidos en la fase 1. Puede provocar iteraciones retroalimentadas sobre los momentos o la fase. Bajo la responsabilidad del director con la participación del estudiante. Momento documentado.

Fase 3, Aceptación: en cumplimiento de lo comprometido en el anteproyecto, se dispone para la consolidación integral del sistema de software y los soportes, en búsqueda del reconocimiento ingenieril, abarcando las pruebas de integración (ISTQB, 2018).

- **Momento 1, Integración de productos:** proceso de refinamiento para la integración de productos al sistema de software, direccionado desde la visión integral de la solución proferida. Asume la responsabilidad el estudiante, con apoyo de la documentación.
- **Momento 2, pruebas de aceptación:** participan las actividades de revisión integral de cumplimiento de compromisos y requisitos, pruebas integrales de la aplicación y análisis documental. De responsabilidad del director quien emite certificación de conformidad ingenieril; con el concurso del estudiante. Asiente el refinamiento documental.

3.4. Estrategias de trabajo

Consecuentes con lo pregonado por Pressman, et al. (2015), al calificar un buen proceso de desarrollo de software, aquel que se encuentra orientado a las personas; MaTraGra, propone como estrategias de trabajo: trabajo colaborativo, asignación y cumplimiento de responsabilidades, retroalimentación frecuente, documentación concurrente, bitácora de actividades y pruebas programáticas. Para apoyar el desarrollo sistemático del proyecto de software, se presenta la aplicación TraGrapp¹ -Ver Figura 3-, elaborada para plataformas de dispositivos móviles con servicios Web, con el propósito de administrar y hacer seguimiento a los procesos y actividades contemplados en las fases y momentos.

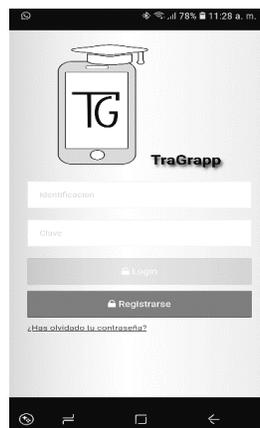


Figura 3. Aplicación TraGrapp.

4. Conclusiones

El perfil orientador de La metodología MaTraGra se direcciona preferentemente al desarrollo de proyectos de software inmersos en trabajos de grado en pregrado de Ingeniería, sin que se pierda la proyección de la misma a otros espacios de aplicabilidad. Las características relevantes de la metodología se adhieren a los principios universalmente aceptados para las metodologías ágiles y responde a espacios académicos dispuestos y reglamentados por las IES para el efecto.

La metodología MaTraGra, participa de la especificidad reveladora de los trabajos de grado donde se plantean proyectos de software, consecuente con la problemática instituida en la dificultad de asimilar las metodologías ágiles de aceptación en el mercado y promulgadas por la IS; contribuyendo a esta práctica ingenieril integral, para evidenciar las competencias de formación

¹ TraGrapp: aplicación elaborada por Castillo P, F. R. y D'Aleman S., C. S., en el marco de trabajo grado en Ingeniería de Sistemas.

académica disciplinar, mediante la aplicación de saberes en el ámbito académica, de connotación diferenciadora respecto al mundo empresarial u organizacional a donde apuntan el perfil de aplicación de las metodologías, sin aislarse de la visión del desempeño profesional.

Se estima la proyección de MaTraGra acogida a la dinámica evolutiva de la tecnología y productos ingenieriles, expuesta a la globalización y la consecuente conjunción con cánones formativos y humanísticos promovidos en el sistema educativo; asimismo, con el mercado competitivo globalizado, cimentada en las TI y la producción ágil de software de calidad; potenciando el recurso humano calificado participe de estos movimientos, acusando de la academia la responsabilidad de cubrir los espacios de formación profesional competente, mediante la concepción de currículos pertinentes.

5. Referencias

- Alves d. S., D.; Costa d. O., E.; Dias C., E. y Ferreira M., H. (2016). Application of a Hybrid Process Software Requirements Management. Digital Library: IEEE Xplore. Recuperada de: <https://ieeexplore.ieee.org/document/7521442/>. (20/05/2019).
- Beck, K.; Beedle, M.; Bennekum, A. V.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.M; Mellor, S.; Schwaber, K; Sutherland, J. y Thomas, D (2001). Manifiesto por el Desarrollo Ágil de Software. Recuperado de: <http://agilemanifesto.org/iso/es/manifesto.html> (20/05/2019).
- Chandra, V. (2015). Comparison between Various Software Development Methodologies. International Journal of Computer Applications, Recuperado de: <https://pdfs.semanticscholar.org/e237/f9cb136f494c2bd0ce91525808c5c968b6b4.pdf>. (20/05/2019).
- Cockburn, A. (2007). Agile Software Development. 2ª Edition. Madrid: Addison Wesley.
- IEEE -Instituto de Ingenieros Eléctricos y Electrónicos- (1993). Standards Collection: Software Engineering. Recuperado de: <https://ieeexplore.ieee.org/document/392549/> (20/05/2019).
- INCONTEC - Instituto Colombiano de Normas Técnicas y Certificación. (2017). Normas Colombianas para la presentación de tesis de grado. Bogota: INCONTEC, NTC 1486.
- ISTQB - International Software Testing Qualification Board. (2018). Pruebas de aceptación del software. Recuperado de: <https://www.istqb.org/>. (02/05/2019).
- Kendall, K. E. y Kendall, J. E. (2011). Análisis y Diseño de Sistemas. Octava edición. Colombia: editorial Pearson, Prentice Hall.
- Madeja - Marco de Desarrollo de la Junta de Andalucía (2013). Ingeniería de Requisitos. Versión 1.5.0. Recuperado de: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos>. (20/05/2019).
- Piattini V., M. G.; García R., F. O; García R., I. y Pino, F. (2015). Calidad de Sistemas de Información. 3ª Edición ampliada y actualizada. Alfaomega, Ra-Ma.
- Petrella, C. (2007). Aportes del Enfoque Sistémico a la Comprensión de la Realidad. Avances del proyecto de investigación. Recuperado de:

<https://www.fing.edu.uy/catedras/disi/DISI/pdf/Teoriadesistemasaplicadoaorganizaciones.pdf>. (20/05/2019).

- Pressman, R. S. and Maxim, B. R. (2015). Software Engineering. A practitioner's approach. Eighth Edition. New York: McGraw Hill Education.
- Sáez V., F. (2009). Complejidad y Tecnologías de Información. Cuadernos de Tecnología y Sociedad n°3. Madrid: fundetel. Recuperado de: http://dit.upm.es/~fsaez/intl/libro_complejidad.pdf. (20/05/2019).
- Somerville, I. (2011). Ingeniería del software. 9ª Edición. España: Pearson editores.
- Tinoco, O.; Rosales L., P. P. y Salas B., J. (2010). Criterio de selección de metodologías de desarrollo de software. España y Portugal: Red de revistas científicas de América Latina y el Caribe.
- Torres S., L. C. (2012). Tesis de Maestría. Qué Hacer. Bogotá: Universidad Autónoma de Colombia.
- VersionOne (2018). 13th Annual State of Agile™ Report. Recuperado en: <https://explore.versionone.com/state-of-agile/13th-annual-state-of-agile-report>. (20/05/2019).
- Wiegers, K. E. y Beatty, J. (2013). More About Software Requirements. Third Edition. ed. Redmon. Washington: Microsoft Press.

Sobre el autor

- **Gustavo A. Rivera S.**, Ing. de Sistemas; Esp. en Auditoría de Sistemas y Redes de Telecomunicaciones; MSc. en Telemática e Informática Aplicada en la Educación; y doctorando en Educación. Profesor Investigador Universitario. Correo: gustavorivera11@hotmail.com.

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2019 Asociación Colombiana de Facultades de Ingeniería (ACOFI)