

## **Metodología ágil orientada a trabajos de grado - MaTraGra. Gustavo A. Rivera S.<sup>1</sup>**

### **Resumen.**

Las metodologías ágiles para el desarrollo de software, descansan su preferencia en las bondades traducidas en la rapidez en la obtención de productos de calidad, resaltando la aplicabilidad enfocada a espacios empresariales. Al ser acogidas en ambientes académicos, se presenta la dificultad de adaptarla, en razón de las diferencias respecto a la disposición de recursos humanos y económicos, y las responsabilidades en el orden académico. En consecuencia, se propone una metodología ágil orientada al desarrollo de proyectos de software en los trabajos de grado, denominada MaTraGra, acogiendo las particularidades demandadas en espacios de formación profesional ingenieril.

La metodología MaTraGra, adherida a los diseños expuestos para las metodologías ágiles, parte de la definición del anteproyecto, para proponer las fases de requisitos, desarrollo y aceptación, expuestas a la iteración de procesos, participando de las siguientes características: desarrollo incremental de productos, equipos de trabajo pequeños y cohesionados, trabajo colaborativo y asesorado, procesos retroalimentados consecuentes con puntos de validación y pruebas, y documentación concurrente en cumplimiento de normas institucionales.

### **PALABRAS CLAVES.**

Ingeniería de Software, MaTraGra, Manifiesto Ágil, metodología ágil, proyecto de software y trabajo de grado.

### **Abstract.**

Agile methodologies for software development, rest their preference in the benefits translated into the rapidity in obtaining quality products, highlighting the applicability focused on business spaces. When admitted to academic environments, the difficulty of adapting it is presented, due to differences in the disposition of human and economic resources, and responsibilities in the academic order. Consequently, an agile methodology is proposed aimed at the development of software projects in the degree works, called MaTraGra, accepting the particularities demanded in spaces of engineering professional training.

The MaTraGra methodology, adhering to the designs outlined for the agile methodologies, starts from the definition of the preliminary project, to propose the phases of requirements, development and acceptance, exposed to the iteration of processes, participating in the following characteristics: incremental development of products, small and cohesive work teams, collaborative and advisory work, feedback processes consistent with validation and testing points, and concurrent documentation in compliance with institutional norms.

### **KEYWORDS.**

Software Engineering, MaTraGra, Agile Manifesto, agile methodology, software project and degree work.

---

<sup>1</sup> Ingeniero de Sistemas, con estudios de: Maestría en Informática Aplicada a la Educación y en Telemática; Especialización en Redes de Telecomunicaciones y en Auditoría de Sistemas. Profesor investigador universitario.

## Introducción.

Se declara como propósito del documento, la presentación de una metodología inscrita en el dominio de las metodologías ágiles, enfocada a ser utilizada en el desarrollo de proyectos de software planteados en el marco de los procesos de trabajo de grado, en el cumplimiento del plan de estudios ingenieriles en Instituciones de Educación Superior -IES; metodología denominada MaTraGra, por sus siglas obtenidas de Metodología ágil para Trabajos de Grado.

Reconociendo los trabajos de grado, como un componente de los programas académicos de pregrado, en donde se plantea un ejercicio de profundización teórico y práctico, participe de la formación profesional integral, en un escenario de cumplimiento normativo expuesto a criterios reglamentarios formalmente establecidos por las instituciones educativas. Asimismo, provee del espacio con la flexibilidad de ser utilizado para el abordar y dar solución a problemas de perfil acorde a la orientación profesional que caracteriza el programa académico.

La concepción y desarrollo de proyectos ingenieriles, dentro del espectro constitutivo por los trabajos de grado, permite involucrar procesos de desarrollo de software, planteados bajo la tutela disciplinar de la Ingeniería de Software -IS-; proyectos de software, definidos por el IEEE (1993) como *la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de ingeniería al software*. En donde se acogen modelos y metodologías, estudiados y presentados por la IS en dos grandes categorías: metodologías ágiles y tradicionales.

Típicamente las metodologías tradicionales se caracterizan por orientarse en una planeación exhaustiva, perpetrada por la secuencialidad rigurosa y el control del proceso apoyado en la documentación, donde se puede generalizar su estructuración en cuatro etapas: identificación de requerimientos, diseño y planeación del trabajo, desarrollo de productos y etapa de pruebas con la intervención del usuario.

Son características diferenciadoras de las metodologías ágiles, al ser comparadas con las tradicionales: disposición al desarrollo del software en forma incremental e iterativo; equipos de trabajo pequeños y auto organizados; flexibilidad en los procesos; y, manejo de tiempo corto en la toma de decisiones y entrega de productos. Rasgos distintivos que promueven la mayor utilización (VersionOne, 2017), sustentada en el cumplimiento de entregas de productos de software de calidad en menor tiempo, tendencia acogida en los trabajos de grado que proponen el desarrollo de aplicaciones de software.

Un tercer grupo de metodologías se presentan como híbridas, en donde se advierten en su constitución, la presencia de las ventajas de las metodologías tradicionales y ágiles, sus mejores prácticas y herramientas, además, donde se determina la propiedad de robustez conjugada con la flexibilidad, planteamiento compartido por Alves, Costa, Dias y Ferreira (2016).

El ambiente académico donde transcurre los proyectos de grado, propone particularidades diferenciadoras para los procesos de desarrollo de software, promulgadas por la falta de experticia de los estudiantes en abordar proyectos de tal magnitud y responsabilidad, la complejidad para asimilar la metodología acorde a los tiempos de trabajo, la designación de roles, la adquisición y administración de recursos, entre otros. Todo esto propone un nivel de dificultad para adoptar la metodología adecuada, debido a la orientación de las mismas a espacios de trabajo empresariales.

Una de las consecuencias de acoger metodologías en las que se dificulta el acoplamiento a ambientes académicos, es la dilatación del proceso por el incumplimiento de los tiempos requeridos y conciliados con la institución universitaria, problemática que alimenta la deserción o la terminación tardía del ciclo de formación profesional. Planteamiento que sustenta la caracterización de la metodología MaTraGra, pensada acorde al desarrollo de software en los trabajos de grado, afín a los preceptos contemplados en el *Manifiesto Ágil*, sumergida en los planteamientos disciplinares de la IS, y la contextualización del enfoque sistémico.

La metodología MaTraGra, es un producto de un proceso investigación, proyectada para ser utilizada como un marco procedimental para perfilar el trabajo comprometido en los trabajos de grado en programas académicos inscritos a IES, desde la perspectiva de planificación, operación y control del proceso de desarrollo de software, bajo la premisa de participar en el ejercicio académico de formación profesional del ingeniero.

Desde la revisión de antecedentes de metodologías de connotación temática similar a la presentada, en el marco de la investigación, se pueden extraer singularmente las siguientes características: apoyadas en los designios universalmente aceptados para las metodologías ágiles, en relación al desarrollo incremental e iterativo, designación del cliente como integrante de los roles y la utilización de artefactos; y orientadas a espacios específicos de aplicación diferentes al desarrollo de software en los trabajos de grado.

En el documento, se presentan los sustentos teóricos de la metodología, en relación a los eslabones referenciales en los que se apoya, la caracterización conceptual que la identifica, la propuesta metodológica en sus facetas componentes y propuestas para su utilización y, las conclusiones sobre el trabajo de investigación fuente de la metodología.

### **Contexto teórico de MaTraGra.**

Considerado como un componente integral para el cumplimiento del plan de estudios de programas académicos en IES, interpretado por Torres (2013), como el resultado de un proceso creativo e innovador de carácter reflexivo y crítico, basado en conocimientos existentes y proyectados a producir un nuevo conocimiento, para dar solución a problemas atinentes a la sociedad para ampliar el panorama del conocimiento humano.

Las IES tienen la potestad de reglamentar y trazar los lineamientos del trabajo de grado, como requisito para la culminación del plan de estudios, entendiéndose como opción inmersa, la elaboración y desarrollo de proyectos de software, que abordan problemáticas que puedan ser resueltas mediante soluciones disciplinares afines a los planteamientos curricular del programa académico.

El trabajo de grado participa de una disyuntiva, el establecerse como un espacio académico de evidencia de competencias teóricas y prácticas para la calificación profesional del estudiante en su proceso de formación; y el asimilarse, como un proceso de búsqueda de la excelencia para desempeñarse profesionalmente en el ambiente externo a la academia. Disyuntiva latente en Ingeniería, reflejada al proyectar el ejercicio práctico en las aulas, a las realidades del ambiente real de desenvolvimiento profesional, que califique su idoneidad.

Con el propósito de sustentar la calificación de las metodologías ágiles, la agilidad referida a las metodologías para el desarrollo de software, es concebida por Qumer y Henderson-Sellers (2008), como una habilidad o comportamiento permanente para acoplarse a los cambios en forma expedita, para

obtener resultados en menor tiempo, con herramientas eficientes, fundamentada en la preparación y experticia para la comprensión del sistema en su totalidad.

Es considerado el denominado *Manifiesto Ágil* (Beck, et al, 2001), el principal hecho histórico que avala el surgimiento de las metodologías ágiles, documento del cual se extraen los siguientes planteamientos, tomados como base de la metodología expuesta: priorizar las personas sobre los productos; la visión de la calidad funcional del producto; el trabajo colaborativo; y, la flexibilidad como respuesta al cambio.

El abanico de metodologías para el desarrollo de software promulgadas por la IS, se catalogan en dos grandes grupos: las tradicionales que participan de un ciclo de vida del software, con procedimientos estrictos y complejos que responden a un plan (Bennett, McRobb, y Farmer, 2010); mientras que las metodologías ágiles, se caracterizan por la flexibilidad en el proceso, el desarrollo iterativo e incremental, y los resultados a corto plazo (Kendall y Kendall, 2011), complementada, con la capacidad de adaptarse a ambientes particulares de desarrollo de productos (Tridibesh Satpathy, 2016). Diferencias que proveen de razones preferenciales para la decisión de su elección.

Los postulados para la proliferación del mayor uso de las metodologías ágiles son presentados por (VersionOne, 2017), en el informe del estado de agilidad, de donde se resalta: la permisibilidad a los cambios de prioridades, aumento de la productividad asociada al equipo de trabajo, la calidad de los productos de software obtenidos y el mejoramiento en los tiempos de cumplimiento de compromisos. En el mismo informe se evidencia la tendencia al uso de las metodologías Scrum y XP, en su orden, por su funcionalidad y operatividad.

Una presentación en síntesis de las diferencias de metodologías ágiles en relación con las tradicionales, se expone en García y Concepción (2015), bajo las siguientes características:

- Basadas en heurística más que en estándares.
- Los cambios se dan más frecuentemente.
- Retroalimentación continua.
- Entrega de productos en menor tiempo.
- Diversidad de roles.
- Los equipos de trabajo incluyen al cliente.
- La arquitectura del software es revisable.
- Menor control normativo.
- Contratación flexible.

El camino de las metodologías ágiles, le proyecta además del desarrollo de software con técnicas de respuesta rápida, se complementa a nivel predictivo para minimizar la incertidumbre extendiendo su alcance perimetral a la gestión de proyectos, constituyendo ámbitos de perfiles de expertos asesores de reconocimiento empresarial, consecuente de una cultura empresarial de agilidad enfocada a la competitividad y centradas en el cliente.

En el campo ingenieril, Sáez (2009) asimila el enfoque sistémico con el término *software mental*, aduciendo la posibilidad de ser invocado en el momento de identificar la estructura y calidad de los diseños y desarrollos, adicionalmente, en la utilización de herramientas para el estudio de la complejidad que nos lleve a la construcción de metodologías. Entre tanto, Gay (1999) califica al enfoque sistémico, adecuado para organizar el conocimiento y promover la eficacia de la acción entorno a él.

Adicionalmente, en Petrella (2007) propone el enfoque sistémico para la construcción de modelos representativos de sistemas administrativos donde se tipifique el desarrollo gradual.

Al acoger el enfoque sistémico en la metodología MaTraGra, se plantea como un eslabón de contextualización, para aplicar y comprender integralmente el objeto de estudio desde tres perspectivas: un componente holístico, en la visión al abordar el problema; en segundo lugar, el componente heurístico que propicia la razón metódica al trabajo; finalmente, el componente ético y estético intencionado y direccionado a promocionar la proyección de la solución.

Una metodología para el desarrollo de software es ubicada como un proceso para el cumplimiento del adelanto de un proyecto de software, constituido por etapas bien definidas (Chandra, 2015). Las metodologías evolucionan en concordancia de las tecnologías de información, demandando un mayor nivel de eficiencia reflejada en mayor productividad, acogiendo procedimientos, técnicas herramientas y documentos orientados a facilitar la gestión y desarrollo de productos de software (Tinoco, Rosales y Salas, 2010).

La metodología MaTraGra fue expuesta al análisis propuesto por la herramienta Dimensional Analytical Tool 4-DAT obteniendo resultados altamente satisfactorios; herramienta presentada por Qumer y Henderson- Sellers (2008), como herramienta analítica encaminada a la medición de las metodologías ágiles, el nivel de agilidad y adaptabilidad en el proceso de desarrollo de software, basada en cuatro perspectivas de caracterización:

1. El alcance de la metodología, determinada por la orientación, tamaño del proyecto y equipo de trabajo, paradigma y plataforma de desarrollo de producto.
2. Grado de agilidad, medición fundamentada en la flexibilidad de la metodología, prontitud de resultados, capacidad de respuesta y facilidad de aprendizaje a través de ella.
3. Cumplimiento de los preceptos del *Manifiesto Ágil*, reflejado en la agilidad del proceso y productividad.
4. Ingeniería y gestión de procesos de desarrollo software, respecto a las prácticas utilizadas dentro del ciclo de vida.

Para la medición de la usabilidad, se expone como referencia utilizada, el Modelo de Aceptación de Tecnología -TAM, por sus siglas en inglés-, presentado por Davis (1989), para establecer el nivel de aceptación de la tecnología por parte de los usuarios, fundamentada en la actitud positiva y el uso adecuado, embebidos por aspectos atinentes a la convicción, actitud, intención y comportamiento de los usuarios de la metodología; modelo implícito en la concepción del estudio de caso.

### **Fundamentación metodológica de MaTraGra.**

Metodología desarrollada como un producto expuesto a un proceso de análisis experimental de carácter cualitativo, en un fenómeno educativo planteado como estudios de caso determinados por trabajos de grado inmersos en programas académicos de Ingeniería donde se plantearon proyectos de software. Para determinar su eficiencia comparativa en relación a la adopción de metodologías ágiles universalmente utilizadas.

La investigación cualitativa se aborda, justificada a partir de asumir muestras pequeñas alejadas de sustentación estadística, mediante el acercamiento disciplinar a el sujeto de estudio, apoyados por Montero y León (2002); complementado por aspectos como la experticia del investigador, utilización de

técnicas flexibles, perspectiva individual y holística, y la participación diagnóstica de hechos manifiestos e implícitos en el caso de estudio, de acuerdo a lo expuesto por Hernández, Fernández y Baptista (2014).

El estudio de caso, para su acogimiento en fenómenos educativos, propone el poderse aplicar en investigación de cobertura limitada, en la formación profesional, en donde se promueve el trabajo en equipo, en procesos de toma de decisiones de acuerdo a Latorre y otros (1996); todo ello se concibe como peculiaridades de los procesos de desarrollo de software inmersos en los trabajos de grado, en un ambiente limitado en tiempo y espacio, según Neiman y Quaranta (2006).

Para el diseño y desarrollo del estudio de caso, como base experimental para la identificación de razones de uso de la metodología MaTraGra, se llevaron a cabo cinco fases: selección y definición del estudio de caso, planteamiento de preguntas de soporte, identificación de las fuentes de datos, análisis e interpretación, e informes, apoyados en lo citado por Montero y León (2002). En un trabajo realizado desde el segundo semestre del 2016 hasta el primer semestre de 2018.

La entrevista acogida como técnica instrumental aplicada al contexto educativo en la formación profesional se plantea bajo la tutela de lo sugerido por Nahoum (1990), al proponerse dentro de un contexto determinado por el estudio de caso, no como experiencia de laboratorio Vallés (2009). Entrevista acogida de tipo conducida con formato semiestructurado, aceptada la confiabilidad como instrumento, por profesionales, mediante el Coeficiente de Validez de Contenido CVC de acuerdo a Hernández-Nieto (2002), y su calificación utilizando una escala de medición de Likert.

En la investigación, como técnicas complementarias, fueron utilizadas la observación consecuente con la participación en los estudios de caso, y la revisión documentada para el sometimiento a estudio, desde el soporte conceptual y teórico, hasta la verificación en orden de diagnóstico de los resultados del caso de uso, en aras de soportar la explicación del fenómeno experimentado adheridos a lo afirmado por Yuni y Urbano (2014).

Trabajo de experimentación suscrito al tipo de investigación explicativa, sustentado en la información obtenida de los participantes del trabajo de grado, mediante entrevista al equipo de estudiantes y tutores o directores participantes, y el monitoreo del proceso; todo ello para su entendimiento, de acuerdo a lo expuesto por Cazau (2006).

Para el desarrollo experimentación de la metodología, se cumplieron las siguientes etapas o fases, sustentadas en Latorre, Rincón y Arnal (1996):

1. Exploratoria: corresponde al planteamiento de la investigación.
2. Planeación: caracterización del escenario del estudio de caso y las técnicas y herramientas utilizadas.
3. Entrada al escenario: identificación de la población objetivo en los trabajos de grado.
4. Fase analítica: basada en el plan de trabajo y el proceso de acopio de información para su correspondiente análisis.
5. Desarrollo del estudio de caso: de acuerdo a la planeación concebida.
6. Fase informativa: de la cual hace parte la presentación del informe de investigación y la divulgación del artículos y ponencias.

Se exige de una metodología, apoyar la competitividad de las organizaciones permitiendo disponer de productos de software desarrollados con mayor prontitud sin riesgo alguno de la calidad, aspectos que sustentan el uso de las metodologías ágiles. En forma paralela, se hace necesario de personal experimentado que participe y lidere estos proyectos, momento en donde entra la academia a

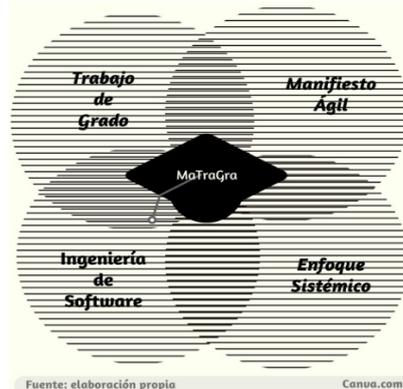
ejercer su labor de formación, apropiándose de la problemática como elemento sustentable de la concepción curricular.

La metodología MaTraGra, se justifica por la dificultad de acoger las metodologías tradicionalmente utilizadas, en un ambiente académico de connotación especial, sin perder la directriz de participar de las competencias de formación en el orden de la IS. Asimilando las características de los trabajos de grado, de conformar equipos pequeños, escasa diversidad de roles, recursos económicos no convencionales, manejo de tiempos con repercusión en responsabilidades académica, consecuencias centradas en el estudiante de acuerdo a la reglamentación institucional.

Para el entendimiento contextual de MaTraGra, se trazan cuatro saberes referenciales, expuestos en la figura 1:

1. Trabajo de grado: como fundamento de caracterización de la aplicación metodológica en el campo ingenieril y en cumplimiento de la normatividad institucional.
2. *Manifiesto Ágil*: en adherencia a las directrices o postulados, para ser catalogada como metodología ágil.
3. Ingeniería de Software: le provee de los planteamientos disciplinares metodológicos para el proceso de desarrollo de software.
4. Enfoque Sistémico: desde el ámbito de la complejidad, otorgando el espacio de visión sistémica sobre el problema tratado, la aplicación de la metodología con la perspectiva de uso de técnicas y herramientas pertinentes, y la participación en la formación integral.

**Figura 1. Referentes de MaTraGra.**



La metodología MaTraGra, exhibe un marco procedimental de conductas de acción orientadas al cumplimiento de unos compromisos académicos, presentados como un conjunto pasos o facetas que nos provee de directrices para hacer del trabajo una búsqueda permanente de la eficiencia a través de estrategias ingenieriles de cumplimiento. La concepción de la metodología se entendió en cumplimiento de los siguientes criterios fundamentales para su aplicación:

- Orientación a espacios académicos de formación en Ingeniería, para el desarrollo de software de calidad propuesto en los trabajos de grado.
- Trabajos de grado caracterizados por personal altamente cohesionado, comunicación fluida, ciclos de trabajo cortos, documentación permanente y procesos de validación frecuente con retroalimentación disciplinar.
- En el ámbito de las metodologías ágiles, respecto a la estructuración en procesos iterativos, desarrollo incremental, colaborativo y designación de roles en equipos de trabajo pequeños.
- Flexibilidad y, de fácil entendimiento, adaptación y puesta en marcha.

En cumplimiento de lo anteriormente expuesto, se abordaron los siguientes pasos para la construcción de la metodología:

- Identificación de requerimientos: reconocimiento documental y procedimental para afinar el marco teórico, conceptual y operativo.
- Análisis: estudio del espectro de las metodologías, haciendo énfasis en las ágiles, la especificidad disciplinar ateniende y extraída de la IS, el enfoque sistémico como pilar de contextualización, y la caracterización de los trabajos de grado en el desarrollo de proyectos de software.
- Construcción: acorde con el análisis, la determinación de lineamientos de la metodología MaTraGra, referentes, componentes y descriptores.
- Pruebas: mediante la aplicación de la metodología en estudios de caso experimentales, en trabajos de grado que incluyen procesos de desarrollo de software.
- Difusión: documentación pertinente y socialización en eventos formales.

### **Roles.**

El equipo humano determinado para el acogimiento de la metodología, se encuentra supeditado al ambiente expuesto por los trabajos de grado y se manifiesta como consecuencia de la importancia del trabajo en equipo en la formación profesional, acorde con lo afirmada por Humphrey (2005); asimismo, apoyado en talentos y habilidades, en donde el proceso de desarrollo ágil de software se adapta a ellos, (Cockburn, 2007); en cumplimiento de requerimientos de las organizaciones (Pressman y Maxim, 2015); adicionalmente, en equipos de desarrollo pequeños y estables (Somerville, 2011).

Siguiendo las premisas citadas y relacionadas con el recurso humano, la metodología MaTraGra advierte la conformación de dos roles:

1. Estudiante: por el compromiso académico, se le otorga la mayor responsabilidad en el cumplimiento de lo estipulado en el anteproyecto, en lo ateniende a los productos de software y su correspondiente documentación.
2. Director: coordina el proceso en forma integral, mediante validaciones frecuentes del software y la documentación, asumiendo informalmente la representación del cliente, con perfil especializado ingenieril. Emite, finalmente, la certificación de cumplimiento a cabalidad del proceso y la calidad de los productos.

### **Documentación.**

La documentación demuestra su importancia, para efectos de revisión con fines de corrección, apoyar la usabilidad del sistema, y soportar y evidenciar procesos de mantenimiento de productos, conforme a lo afirmado por Pressman y Maxim (2015), adicionalmente, se considera soporte documental para el registro de actividades y la administración de agenda asociada a los procesos y personas (Piattini y otros, 2015). En el desarrollo ágil, los documentos formales contribuyen a los procesos de comunicación, preferentemente cuando se direccionan de acuerdo a los requerimientos en iteraciones o fases (Somerville, 2011).

Para el acogimiento de la metodología MaTraGra, se establecen dos categorías concurrentes de documentación:

- Registro de actividades -figura 2-: documento en el que se consignan las actividades propuestas y realizadas, sumado, al visado del responsable, fecha propuestas y de cumplimiento. Constituida en agenda perpetrada mediante el consenso del equipo de trabajo

y el director, denotando la flexibilidad factible en las metodologías ágiles, y de seguimiento continuo y de elaboración concurrente con las fases e iteraciones.

- Documento ingenieril: en cumplimiento de los requerimientos o requisitos establecidos por la institución donde se suscribe el trabajo de grado, para el desarrollo ingenieril del ejercicio académico previsto para el proyecto de software, como soporte del proceso con miras a la sustentación formal. Conformado por evidencias de apropiación de competencias profesionales en el área disciplinas de la IS. Se determina como válido, mediante la certificación de idoneidad del director.

**Figura 2. Registro de actividades.**

MáTraGra

REGISTRO DE ACTIVIDADES

TRABAJO DE GRADO:

ESTUDIANTES: \_\_\_\_\_

DIRECTOR: \_\_\_\_\_

Nº.	FECHA HOY	ACTIVIDADES REALIZADAS	ACTIVIDADES A REALIZAR	FECHA PRÓXIMA	FIRMA ESTUDIANTE	FIRMA DIRECTOR	OBSERVACIONES
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

Fuente: elaboración propia.

### Componentes de MáTraGra.

Apoyados en lo dispuesto por Pressman y Maxim (2015), al presentar la metodología desde dos categorías: una filosofía, que responde a los preceptos estipulados para las metodologías ágiles; y, en segundo lugar, los lineamientos para el desarrollo, en donde se involucran procesos de análisis, diseño, comunicación frecuente y retroalimentación. Complementado por Anaya y Sepúlveda (2013), al plantear que el proceso desarrollo de software se constituye en fases cumplidas desde la definición de requisitos, pasando por el análisis, diseño e implementación, y concluyendo con las pruebas de cumplimiento.

Consecuente con lo anteriormente expuesto, la metodología MáTraGra propone las siguientes fases mostradas en la figura 3:

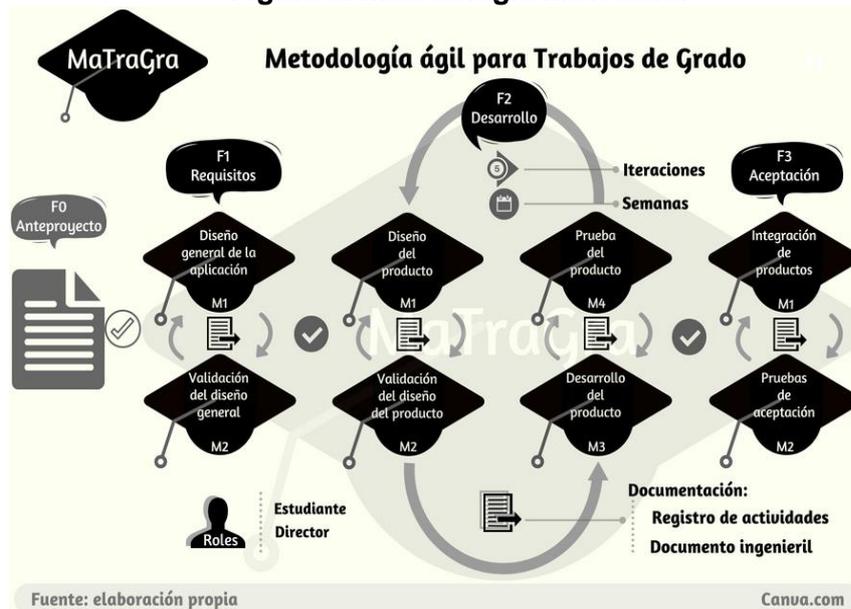
**Fase 0, Anteproyecto:** considerada como prerrequisito no constitutivo de la metodología, en donde se esbozan los compromisos que encierran el trabajo de grado. Anteproyecto precisado por INCONTEC (2017) como, el *documento en el que se identifica y precisa la idea que constituye el núcleo del problema del trabajo de grado*; asimismo, Ramos y otros (2017) lo consideran como la etapa para concebir el proyecto y entender el sistema a construir.

**Fase 1, Requisitos:** La identificación de requisitos corresponde a la determinación de las características del producto en procura de su calidad, de acuerdo a las conveniencias del negocio propuesta extraída de MADEJA (2013); se plantea como consecuencia del reconocimiento del problema; asimismo, permite entender el comportamiento del software a construir, sus propiedades y restricciones, (Wieggers y Beatty, 2013), además, proponen tres etapas iterables para llevarse a cabo: recolección, análisis, y especificación y validación. En procesos iterativos, en los requisitos se establece el alcance y

responsabilidades, además incluye, la asignación de tareas, identificación de riesgos y los criterios para realizar las pruebas (Ramos y otros, 2017).

La fase de requisitos se encuentra conformada por dos momentos iterativos o cíclicos: el diseño del producto fundamentado en la definición de requisitos y su correspondiente validación retroalimentada que alimenta la iteración.

Figura 3. Metodología MaTraGra.



**Momento 1, Diseño general de la aplicación:** entendiendo el diseño como la perspectiva arquitectónica del producto software, donde se evidencian los componentes, interfaces y demás características estructurales, expuesto por Sánchez y otros (2011). En proceso ágiles, en ésta etapa se propone la arquitectura general del sistema, afirmado por Somerville (2011).

En consecuencia, se establece como un momento en el que se determina, en diseño, la arquitectura global de la aplicación, la especificación de la plataforma y herramientas de desarrollo, el diseño general del tratamiento de datos, entre otros.; todo ello plasmado en el documento ingenieril, como sustento del trabajo. Adicionalmente, se establece la planificación funcional y operativa del trabajo, espacios físicos, momentos de entrega y responsabilidades, y especificaciones de las pruebas; compromisos consignados en el registro de actividades. Momento de responsabilidad del equipo de estudiantes.

**Momento 2, Validación del diseño general:** momento asignado a la evaluación ingenieril del diseño concebido, produciendo como consecuencia, la retroalimentación ingenieril pertinente utilizada para concebir la iteración dentro de la fase; en general, se establece como un espacio para determinar la viabilidad y consistencia tecnológica. Momento de responsabilidad del director, con la participación del equipo de trabajo de estudiantes.

**Fase 2, Desarrollo:** consecuente con la fase anterior, fase utilizada para el desarrollo incremental e iterativo del aplicativo de software, facilitando el acompañamiento y la retroalimentación, de acuerdo a lo expuesto por Ramos y otros (2017); etapa compuesta de ciclos de trabajo -cinco ciclos recomendados-

. Fase donde puede darse la redefinición de requerimientos a nivel incremental o por componentes (Somerville, 2011).

La construcción del software, conlleva implícitamente una serie de etapas que combina la codificación, verificación, pruebas unitarias y depuración (Sánchez y otros, 2011). La metodología MaTraGra, propone el desarrollo del software mediante la iteración de cuatro momentos, en donde paralelamente se procede a la documentación: diseño, validación, desarrollo y prueba.

**Momento 1, Diseño del producto:** se fundamenta en el análisis de requerimientos específicos del producto entregable, se procede al diseño en modelado ingenieril acudiendo a lo pregonado por la IS, complementado con la determinación del alcance, la especificación funcional y las pruebas contempladas para efectos de validación; procesos expuestos a la documentación ingenieril correspondiente. Momento bajo la responsabilidad del equipo de estudiantes.

**Momento 2, Validación del diseño del producto:** para el aseguramiento del cumplimiento de las especificaciones, mediante la validación del diseño acorde a las pruebas planificadas, validación que produce retroalimentación y la posible iteración de los momentos 1 y 2. Participa del momento el equipo de estudiantes y el director, este último con la responsabilidad de la emisión del concepto consecuente con la evaluación ingenieril. Consecuentemente, momento documentado en el registro de actividades, con la posibilidad de justificar modificaciones en el documento ingenieril.

**Momento 3, Desarrollo de producto:** de responsabilidad exclusiva del equipo de estudiantes con la asesoría del director. Se entiende como la construcción de unidades fieles a las especificaciones del diseño, de acuerdo a lo definido por Piattini y otros (2015). Hace parte de éste momento, las pruebas de componentes o pruebas unitarias, realizadas por el equipo de estudiantes desarrolladores. Trabajo documentado para su respectiva sustentación.

**Momento 4, Prueba del producto:** Las pruebas se orientan a verificar la calidad del software mediante la búsqueda de fallas (Sánchez y otros, 2011). Para la realización de las pruebas, se debe tener siempre la visión de los requisitos generales establecidos en la fase 1, recordando que los cambios en éstos momentos son menos costosos que los que tengan que hacerse sobre el producto final liberado, según lo citado por Kendall y Kendall (2011).

Se trata de un espacio donde se incorpora el director, para que, conjuntamente con el equipo de estudiantes, se efectúen las pruebas planificadas en el momento de diseño. De ésta actividad se derivan recomendaciones que promueven la iteración de los momentos 3 y 4, llegando a producir la iteración de todos los momentos de la fase de desarrollo, al preverse la necesidad del rediseño del producto.

**Fase 3, Aceptación:** propone la consolidación de los productos para el cumplimiento de los estipulado en el anteproyecto, considerando la verificación integral del sistema de software y sus soportes. Puede interpretarse como el reconocimiento ingenieril del trabajo realizado, reflejado en los productos entregables. Se compone de iteración de dos momentos: la integración para la liberación de productos y sus correspondientes pruebas de aceptación.

**Momento 1, Integración de productos:** la visión integral de los productos de software permite emitir conceptos de cumplimiento, respecto al comportamiento integral del mismo en un ambiente de producción. Momento de responsabilidad del equipo de estudiantes, apoyado en el documento ingenieril y estipulado consecuentemente en el registro de actividades.

**Momento 2, Pruebas de aceptación:** Se incorporan a estas pruebas, actividades como: revisión de cumplimiento de requisitos, análisis documental, pruebas de integración, con el compromiso del mejoramiento iterativo, algunas de ellas extraídas de lo citado por Müller (2013).

pruebas orientadas a la certificación final de cumplimiento de lo pactado y consignado en el anteproyecto. Participa el director del trabajo de grado para validar los productos entregables, acorde a los requisitos establecidos, para finalmente emitir la certificación ante la institución, sobre el cumplimiento a cabalidad de los compromisos, desde la perspectiva ingenieril y del usuario final. De los resultados de las pruebas puede ocasionarse la iteración de éstos dos momentos. Igualmente, el momento se documenta en el registro de actividades.

### **Estrategias de trabajo.**

Es conveniente demandar, como estrategias de trabajo para el acogimiento de la metodología MaTraGra, los siguientes aspectos:

- Trabajo colaborativo: entre el equipo de estudiantes y la participación del director desde su perfil de asesor ingenieril.
- Asignación de responsabilidades de orden sistémico y holístico: fundamentado en las competencias y experticia de los estudiantes.
- Cumplimiento de responsabilidades: para la sustentación de la asimilación de las competencias en formación profesional.
- Retroalimentación: consecuente con los procesos de verificaciones y pruebas, base para justificación de iteraciones.
- Documentación concurrente: soporte del trabajo ingenieril y operativo, y el cumplimiento de los requisitos formalmente establecidos por la institución.
- Bitácora del proceso: permitiendo la evidencia operacional e histórica de actividades realizadas.
- pruebas programáticas: planificadas y alineadas a estándares de aceptación en el orden nacional e internacional.

Para la administración y seguimiento de las fases y actividades de un trabajo de grado, se recomienda el uso de la aplicación TraGrapp -figura 4-, elaborada en el año 2017, para dispositivos móviles con plataforma Android, dirigida a dos perfiles de usuario: estudiantes y tutor o director. Provee el servicio Web relacionado con los módulos proveídos: registro y acceso seguro al sistema; registro de información sobre el trabajo de grado; agenda de actividades; módulo de comunicación entre participantes; y módulo de administración del documento maestro del proyecto.

**Figura 4. Pantalla inicial de TraGrapp.**



**Fuente:** foto de la pantalla inicial de TraGrapp.

## Validación de la metodología MaTraGra.

Para la verificación de la metodología, apoyados en Cabot (2013), se fundamentó en la validación de su afinidad con las metodologías ágiles, basados en las propiedades y valores expresados en el *Manifiesto Ágil*. En tal virtud, se acudió al protocolo de análisis valorativo prescrito para la aplicación de la Herramienta Dimensional Analítica 4-DAT, soportado por los planteamientos metodológicos de MaTraGra, confrontada con metodologías ágiles de reconocimiento universal Scrum y XP, para lo cual, se utilizaron datos obtenidos de Qumer y Henderson- Sellers (2006).

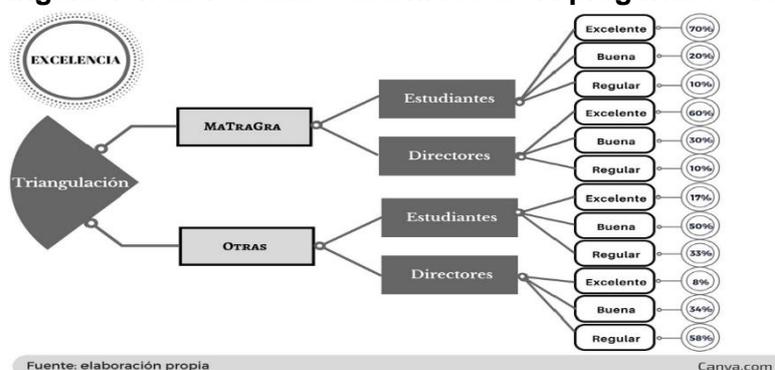
En consecuencia, al implementar el proceso de validación de la metodología MaTraGra, mediante las cuatro dimensiones determinadas por la herramienta de análisis 4-DAT, en síntesis, se obtuvieron los siguientes resultados:

1. Caracterización del alcance: se resalta la disposición a procesos iterativos, rapidez, desarrollo incremental, colaboración y equipos de trabajo pequeños.
2. Caracterización de agilidad: al ser utilizada para el desarrollo de software en los trabajos de grado, obtiene una alta calificación.
3. Caracterización de valores ágiles: la afinidad a los preceptos de metodologías ágiles, se determina por posibilitar los encuentros frecuentes y colaborativos, con retroalimentación basada en pruebas de productos, promocionando el desarrollo rápido.
4. Caracterización de los procesos de software: afinidad con las metodologías ágiles en aspectos procedimentales y de gestión de procesos asociados a los trabajos de grado, documentación concurrente y la asesoría ingenieril del director como experto.

La recopilación de datos a través de entrevistas a estudiantes y directores, Instrumento validado en su confiabilidad por el Coeficiente de Validez de Contenido -CVC-, presentado por Mousall-KavaT (2017), como adecuado para determinar la validez teórica en investigación educativa; sometiéndose a la calificación por parte de profesionales, haciendo uso de la escala de medición de Likert; obteniéndose un resultado satisfactorio por encima del umbral establecido.

La metodología fue sometida a procesos de experimentación en estudios de caso de trabajos de grado en el que se desarrollaron proyectos de software, mediante datos obtenidos por entrevista a los estudiantes y directos participantes, se evidencia el cambio positivo de la percepción de la metodología utilizada, al confrontarla con los trabajos donde se adoptó otras metodologías ágiles. En la calificación de la metodología MaTraGra, se obtuvieron resultados satisfactorios en todos los ítems sometidos a comparación como se muestra en la figura 5.

**Figura 5. Triangulación de los datos obtenidos de la pregunta 10 de la entrevista.**



## Conclusiones.

El trabajo presentado, se suscribe alrededor de la metodología MaTraGra, adherida a lo pregonado por las metodologías ágiles, concebida y orientada al proceso de desarrollo de software dentro de los trabajos de grado en programas de Educación Superior en Ingeniería, apoyada en la disyuntiva detectada en la dificultad para los estudiantes al utilizar metodologías ágiles que se proponen para espacios empresariales.

Las metodologías ágiles en general, resalta en ellas, el desarrollo incremental de los productos en procesos iterativos, proponiendo un buen manejo de los tiempos, con la visión implícita de perseguir la calidad del software; proponen procedimientos fundamentados en la experticia de los participantes, con la incorporación del cliente como parte del equipo de trabajo, y determinación de roles y artefactos de uso utilitarios.

Un trabajo de grado en Ingeniería que incluye proyectos de software, reglamentado al interior de las instituciones universitarias, se considera un ejercicio académico integral de connotación teórico y práctico para el cumplimiento del plan de formación profesional, en donde convergen conocimientos en el ámbito de la IS, para la solución de problemas ingenieriles en un espacio propiciado por la institución de educación.

Los estudiantes que emprenden los trabajos de grado en el que se incluyen proyectos de software, advierten un ambiente diferencial con lo pregonado por las metodologías de orientación empresarial, en lo atinente a la disposición de recursos humanos y económico, asimismo, en la asignación de responsabilidades asumidas preferentemente por el estudiante.

El trabajo de grado, es considerada como una primera experiencia integral de conocimientos alrededor de la IS, en donde el estudiante cumple un único rol dentro del proceso, complementado por la asesoría ingenieril del director del trabajo de grado, además de exigencias documentales particulares a la institución universitaria y con la presión centrada en el cumplimiento académico; elementos diferenciadores de los espacios empresariales a donde se direccionan las metodologías.

La problemática anteriormente expuesta, justifica la concepción de la metodología MaTraGra que, mediante su experimentación en formato de estudio de caso en trabajos de grado, se pudo evidenciar los resultados satisfactorios proferidos por los estudiantes y directores. Metodología en donde se resalta los siguientes criterios que participaron de su desarrollo: orientada a trabajos de grado en donde se planteen proyectos de software; adherida a los principios promulgados para las metodologías ágiles; flexibilidad al momento de aplicarse; y de fácil entendimiento, adaptación, planificación y puesta en práctica.

Confluyen en la concepción de la metodología presentada, la integración de cuatro componentes integradores: en primer lugar, la orientación a ambientes académicos típicos de los trabajos de grado; en segundo lugar, su afinidad con las metodologías ágiles en la caracterización adheridas a los designios del *Manifiesto Ágil*; en tercer lugar, el dominio conceptual fundamentado en la IS; y finalmente, el enfoque sistémico que le otorga el marco de contextualización en el ejercicio de su aplicabilidad.

En relación a la caracterización de MaTaGra, se destacan los siguientes aspectos de afinidad a las metodologías ágiles: enfocada a equipos pequeños y cohesionados, ciclos cortos de trabajo, procesos iterativos, desarrollo incremental de productos, estadios colaborativos expuestos a

retroalimentación, documentación concurrente, y responde a la reglamentación de la institución universitaria sobre trabajos de grado.

Se proponen dos tipos de roles para la utilización de la metodología, como consecuencia de la disposición del recurso humano en los trabajos de grado: en primer término, el estudiante, responsable del desarrollo del proyecto de software y su correspondiente documentación; y el rol de director, asumiendo el compromiso de validación en los puntos de control, y certificación del cumplimiento ingenieril y de calidad relacionado con el proceso y los productos obtenidos.

Respecto a la documentación, se encuentra expuesta al cumplimiento de la normatividad institucional, la metodología propone dos formas, diligenciadas concurrentemente con las fases o etapas: el registro de actividades, para la administración de la agenda de procesos; en segundo lugar, el documento ingenieril que responde a la necesidad de sustentar el proceso de desarrollo de software desde la perspectiva de la IS y la orientación reglamentaria en torno a los trabajos de grado.

La metodología MaTraGra, presenta una fase inicial de prerrequisito constituida por el anteproyecto, no participe de la metodología, el cuerpo sustantivo se conforma de tres fases que señalan en su interior iteraciones promocionadas por puntos de verificación y retroalimentación: fase de requisitos, donde se gesta globalmente el diseño ingenieril del software y la planificación del trabajo; fase de desarrollo, responde a la obtención de los productos de software; y fase final de aceptación, proceso que culmina con la liberación certificada de los productos. Fases que incluyen el desarrollo en paralelo de la documentación correspondiente.

Para la verificación de afinidad y adaptabilidad, la metodología fue expuesta a la categorización mediante la herramienta de análisis 4-DAT, en sus cuatro dimensiones, en donde se sintetiza de los resultados obtenidos: el desempeño estructural y del recurso humanos, acorde al alcance y aplicación esperada; la agilidad procedimental reflejada en los resultados; la alineación en el orden de agilidad en cumplimiento de lo pregonado en el *Manifiesto Ágil*; y, la caracterización de los procesos componentes y de gestión, en la calificación cualitativa positiva al ser concertada con otras metodologías ágiles.

El estudio de caso expuesto para la experimentación de la metodología en trabajos de grado, permitió evidenciar la percepción mediante entrevistas a los estudiantes y directores participantes, el mayor grado de aceptación y satisfacción por la aplicación de la metodología MaTraGra, respecto a las metodologías ágiles comúnmente seleccionadas; sustentando la conclusión de adaptabilidad de la metodología al desarrollo de proyectos de software en ambientes académicos.

De los resultados obtenidos por la experimentación de la metodología MaTraGra, se resaltan positivamente los siguientes: la aceptación de la metodología en orden de aplicabilidad, la retroalimentación frecuente consecuente con los procesos de validación, la participación del director con función de asesoramiento ingenieril y validación de los productos, la disposición de la metodología para el cumplimiento de responsabilidades académicas, y la documentación concurrente como apoyo y sustentación del trabajo.

La presentación de MaTraGra, como un producto metodológico de aplicabilidad en el espectro de la educación, consecuencia de la sed de conocimiento, hace que confluyan corrientes de satisfacción personal y profesional, ahondada por la convicción de participar de la solución ingenieril a una problemática latente, que desemboca en su preferente adopción en conformidad a la disposición curricular de formación integral, humectado por la aceptación a conformidad en su aplicabilidad, para participar de la sed de contribuir a la proyección del bienestar de la humanidad.

## Bibliografía.

- 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/>. (02/02/2018).
- Anaya D., R. y Sepúlveda, M. (2013). Reflexiones sobre ingeniería de requisitos y pruebas de software. Medellín: Fondo Editorial, Corporación Universitaria Remington, Recuperado de: [from https://ebookcentral.proquest.com](https://ebookcentral.proquest.com). (02/02/2018).
- 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> (25/10/2016).
- Bennett S., McRobb S., y Farmer R., (2010). Análisis y diseño en sistemas orientados a objetos con UML, 3ª edición. Madrid: McGraw-Hill.
- Cabot, S. J. (2013). Ingeniería del software. Barcelona: Editorial UOC. Recuperado de: <https://ebookcentral.proquest.com>. (02/02/2018).
- Cazau, P. (2006). Introducción a la investigación en Ciencias Sociales. Tercera Edición. Buenos Aires. Recuperado de: <http://alcazaba.unex.es/asg/400758/MATERIALES/INTRODUCCI%C3%93N%20A%20LA%20INVESTIGACI%C3%93N%20EN%20CC.SS..pdf> (10/03/2018).
- Chandra, V. (2015). Comparison between Various Software Development Methodologies. International Journal of Computer Applications, Recuperado de: <https://pdfs.semanticscholar.org/e237/f9cb136f494c2bd0ce91525808c5c968b6b4.pdf>. (05/02/2018).
- Cockbun, A. (2007). Agile Software Development. 2ª Edición. Madrid: Addison Wesley.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly. Recuperado de: <https://pdfs.semanticscholar.org/bf17/26dc842f91576c97037674c00a712bb5ba8a.pdf>. (02/02/2018).
- García R., M. J. y Concepción S., R. (2015). Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos de software. Universidad de Oviedo. Recuperada de: <http://digibuo.uniovi.es/dspace/bitstream/10651/32457/6/TFMMIJGarciaRodriguezRUO.pdf> (02/02/2018).
- Gay, A. (1999). Temas para Educación y Tecnología. Los Sistemas y el Enfoque Sistémico. Buenos Aires: Ediciones la obra S.A.
- Hernández, R. Fernández, C. y Baptista, L. (2014). Metodología de la Investigación. Sexta edición. México: Mc Graw Hill.
- Hernández-Nieto, R. A. (2012). Instrumentos de recolección de datos. Validez y confiabilidad. Normas y Formatos. Mérida, Venezuela: Universidad de Los Andes.
- Humphrey, W. S. (2005). Psp (sm): A self-improvement process for software engineers. Addison-Wesley Professional.

- IEEE- Instituto de Ingenieros Eléctricos y Electrónicos- (1993). Standards Collection: Software Engineering. IEEE Standard 610.12-1990. IEEE. Recuperado de: <https://ieeexplore.ieee.org/document/392549/> (02/02/2018).
- Kendall, K. E. y Kendall, J. E. (2011). Análisis y Diseño de Sistemas. Octava edición. Colombia: editorial Pearson, Prentice Hall.
- Latorre, A.; Rincón, D. y Arnal, J. (1996). Bases metodológicas de la investigación educativa. Barcelona: GR92.
- MADEJA - Marco de Desarrollo de la Junta de Andalucía (2013). Ingeniería de Requisitos. Version 1.5.0. Recuperado de: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos>. (05/02/2018).
- Montero, I. y León, O.G. (2002). Clasificación y descripción de las metodologías de investigación en Psicología. España: Universidad Autónoma de Madrid.
- Mousalli-Kayat, G. (2017), Los Instrumentos de Evaluación en la Investigación Educativa. Recuperado de: [https://www.researchgate.net/publication/321397866\\_Los\\_Instrumentos\\_de\\_Evaluacion\\_en\\_la\\_Investigacion\\_Educativa](https://www.researchgate.net/publication/321397866_Los_Instrumentos_de_Evaluacion_en_la_Investigacion_Educativa) (02/02/2018).
- Müller, T. (2013). Programa de estudio de nivel básico para probador certificado, ISTQB, recuperado de: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html> (02/02/2018).
- Nahoum, Ch. (1990). El proceso de la entrevista. México: Editorial Kapelusz.
- Neiman, G. y Quaranta G. (2006). Los estudios de caso en la investigación sociológica. En: Aldo y otros. Estrategias de investigación cualitativa. Barcelona: Gedisa.
- 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> (30/10/2017).
- 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.
- Poole, Damon B. (2009). Do It Yourself Agile, September 29th. Manifiesto por el Desarrollo Ágil de Software. Recuperado de: <http://www.agilealliance.org/the-alliance/the-agile-manifesto/> (25/10/2016).
- Pressman, R. S. and Maxim, B. R. (2015). Software Engineering. A practitioner's approach. Eighth Edition. New York: McGraw Hill Education.
- Qumer, A., Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. In: Information and Software Technology.
- Qumer A, Hendeson-Sellers B. (2006). Measuring agility and adptability of agile methods: A 4-Dimensional Analytical tool. IADIS international conference Applied Computing. Recuperado de: [https://www.researchgate.net/publication/268257179\\_Measuring\\_agility\\_and\\_adoptability\\_of\\_agile\\_methods\\_A\\_4-dimensional\\_analytical\\_tool](https://www.researchgate.net/publication/268257179_Measuring_agility_and_adoptability_of_agile_methods_A_4-dimensional_analytical_tool) (01/02/2018).

- Ramos, D. Noriega, R. Laínez, J. R. y Durango, A. (2017). Curso de Ingeniería de Software. 2ª. IT Campus Academy. Recuperado de: [https://books.google.com.co/books?id=G2Q4DgAAQBAJ&printsec=frontcover&dq=ingenieria+de+software&hl=es&sa=X&ved=0ahUKEwiRsp3f7\\_jaAhVozlkKHAVN3ClwQ6AEISDAG#v=onepage&q&f=false](https://books.google.com.co/books?id=G2Q4DgAAQBAJ&printsec=frontcover&dq=ingenieria+de+software&hl=es&sa=X&ved=0ahUKEwiRsp3f7_jaAhVozlkKHAVN3ClwQ6AEISDAG#v=onepage&q&f=false) (02/02/2018).
- Sáez V., F. (2009). Complejidad y Tecnologías de Información. Cuadernos de Tecnología y Sociedad n° 3. Madrid: fundetel. Recuperado en: [http://dit.upm.es/~fsaez/intl/libro\\_complejidad.pdf](http://dit.upm.es/~fsaez/intl/libro_complejidad.pdf) (30/10/2017).
- Sánchez, S.; Sicilia, M. Á. y Rodríguez, D. (2011). Ingeniería del Software. Un enfoque desde SWEBOK. Madrid: Alfaomega, Garceta grupo editorial.
- 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. (2013). Tesis de Maestría. Qué Hacer. Bogotá: Universidad Autónoma de Colombia.
- Tridibesh, S. (2016). *A Guide to the Scrum Body of Knowledge. Scrumstudy*. Phoenix, Arizona. <http://doi.org/10.1017/CBO9781107415324.004>. (02/02/2018).
- Somerville, I. (2011). Ingeniería del software. 9ª Edición. España: Pearson editores.
- Vallés, M. S. (2009). Entrevistas Cualitativas. Madrid: Centro de Investigaciones Sociológicas.
- VersionOne (2017). 11th Annual State of Agile™ Report. The 11th anual State of Agile Survey. Recuperado en: <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2> (25/11/2017).
- Wieggers, K. E. y Beatty, J. (2013). More About Software Requirements. Third Edition. ed. Redmon. Washington: Microsoft Press.
- Yuni, J., y Urbano, C. (2014). Técnicas para investigar. Recursos Metodológicos para la Preparación de Proyectos de Investigación. Córdoba, Argentina: Editorial Brujas.