

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PROCESO Y PROGRESO DE LA FORMALIZACIÓN DE REQUISITOS

Carmen Eduvina Bermúdez

FACULTAD DE INGENIERÍAS

Ingeniería de Sistemas

Director
Prof. Edgar Serna M.

INSTITUTO TECNOLÓGICO METROPOLITANO

Febrero 2016

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

Los métodos formales son un área de las Ciencias Computacionales que ha evolucionado ampliamente en las últimas dos décadas. Actualmente tienen mayor uso en el desarrollo de software crítico, pero varios investigadores vienen trabajando en su incorporación en todo tipo de producto software. En este proyecto se averiguará por el estado actual de este proceso, desde el punto de vista de la formalización de los requisitos. El objetivo es encontrar cuál ha sido el proceso y el progreso de esta formalización, para tener un mapa de la temática y pensarla como contenido del plan de estudios en la Ingeniería de Sistemas.

Palabras clave: Ingeniería del Software, Ingeniería de Requisitos, métodos formales, formalización.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	4
2. MARCO TEÓRICO	9
3. METODOLOGÍA	12
4. RESULTADOS Y DISCUSIÓN	15
5. CONCLUSIONES	28
REFERENCIAS	34
APÉNDICE (Artículo científico)	39

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

Las empresas se enfrentan a un entorno cambiante en el que las demandas y soluciones para los nuevos productos tienen que ser analizadas y evaluadas cuidadosamente. Con el fin de mantener la competitividad, deben elegir una combinación adecuada de funciones y tecnologías, también deben ser capaces de poner en el mercado dichas soluciones de la forma más económica posible. Con ciclos de vida más cortos, se ven obligadas a llevar los productos aún más rápido a la madurez del mercado (Chalupnik, Wynn y Clarkson, 2009). Además, se debe tener especial cuidado en la planificación de tiempos para el desarrollo de los nuevos productos, dichos tiempos deben estar sincronizados con el fin de hacer posible la alineación de las soluciones, con las necesidades de los usuarios.

Cuando se diseña un sistema de desarrollo de un software, se recomienda que los objetivos, las funcionalidades y las restricciones se identifiquen con la mayor precisión posible, esto es lo que se constituye como la especificación de requisitos. Según la investigación realizada se encontró que muchos autores como Azeem, M. et al. (2014), de Sousa, T. et al. (2010), Ambrosio, M. y Andrade, G. (2011), Pandey, S. y Batra, M. (2013), Hassan, R. et al. (2010), Parnas, D. (2010), coinciden en que algo que se repite cotidianamente en el proceso de diseño de un desarrollo de software es que los requisitos suelen ser escritos en lenguaje natural, ya sea porque los ingenieros no conocen los lenguajes formales o porque consideran que es demasiado pronto en el ciclo de vida como para utilizar una especificación formal (es decir, aducen no tener suficientes datos para escribir con éxito un requisito formal). Yan (2011) afirma que los métodos formales son técnicas basadas en las matemáticas, a menudo con el apoyo de herramientas de razonamiento, que pueden ofrecer una manera rigurosa y eficaz para modelar, diseñar y analizar sistemas informáticos.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El principal problema con la escritura de requisitos en lenguaje natural, es que no pueden servir como insumos para las técnicas de verificación y validación automatizadas. De hecho, los lenguajes formales son obligatorios si en el plan de pruebas se pretenden utilizar técnicas automatizadas de verificación, porque son los únicos lenguajes que pueden entender los computadores. Por otra parte, no son ambiguos, es decir, que una sentencia no puede ser entendida de diferentes maneras. Esto implica que los seres humanos también se pueden beneficiar del uso de estos lenguajes, porque entenderán plenamente los requisitos sin una interpretación previa del usuario. El principal problema es que ambas actividades, la lectura y la escritura, no son fáciles de lograr cuando se trata de lenguajes formales.

De acuerdo con la práctica industrial tradicional, los requisitos se especifican en lenguaje natural y la comprobación de errores de forma manual. Las deficiencias de esta cadena de herramientas son bien conocidos: ambigüedad en la descripción de los requisitos, incrementos en costos y tiempos de entrega, y procesos de reingeniería en el diseño y la arquitectura (Dahlstedt y Persson, 2003). Además, las revisiones pueden detectar errores, pero no garantizar su ausencia. Una manera de evitar estos inconvenientes es a través de la formalización de requisitos que según Diallo, R. (2013) es el proceso de describir un sistema y sus propiedades deseadas utilizando un lenguaje con una sintaxis y la semántica definida matemáticamente. Diversas investigaciones se han llevado a cabo con este objetivo, especialmente en soporte desde los lenguajes formales y las herramientas de automatización (Heimdahl y Leveson, 1995; Heitmeyer, Jeffords, y Labaw, 1996; Post, Hoenicke, y Podelski, 2011, 2011a, 2011b; Yu et al., 2008). El inconveniente es que poder determinar si estos resultados son factibles en la práctica, no se puede lograr sin un decidido argumento de principios que se apliquen uniformemente en proyectos de la vida real.

El proceso de formalización consiste en escribir requisitos utilizando una notación formal matemática. Lo que puede ser una tarea difícil si se hace directamente desde un requisito informal y no puede haber una verdadera guía para la formalización: solamente

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

los conocimientos y las experiencias, y tal vez algunas heurísticas (Jureta, Faulkner y Schobbens, 2008), pueden ayudar a saber cómo conseguir una escritura formal.

Los estudios acerca de la formalización de requisitos realizados por Boehm (1984, 1987) y otros han demostrado que el impacto potencial de una formulación deficiente de requisitos es sustancial. Boehm sugiere que los errores en la especificación y el análisis de requisitos son los más numerosos en un sistema, y que la mayoría de ellos no se encuentran en la etapa de desarrollo, sino en las pruebas y la entrega. El costo resultante para corregir estos errores tiene una relación directa con el tiempo invertido en su búsqueda, se puede evidenciar este argumento en la investigación realizada por Kaur, A., Gulati, S. y Singh, S. (2012) quien afirma que el costo de la renovación de fallos es mucho mayor en fase posterior que en la fase anterior. De ahí la importancia de trabajar en la formalización de los requisitos, porque de esta manera se podrían entregar requisitos no-ambiguos y posiblemente con pocos errores. En este artículo se presenta un análisis al progreso y proceso de la formalización de requisitos. El objetivo es trazar un mapa de qué se ha hecho, qué se ha logrado y cuál es el futuro de la formalización como herramienta en la Ingeniería de Requisitos.

La seguridad y calidad en los sistemas software son motivos para afirmar que en nuestra época es pertinente aplicar la formalización de requisitos para dar solución a los problemas de la ingeniería de requisitos, Como mala estructuración del desarrollo, sistemas no fiables, sobrecostos por corrección de problemas en otras etapas del desarrollo o al final, entre otros. La ingeniería de requisitos según Roberto Cavada y sus colegas (2010) es una de las fases más importantes del proceso de desarrollo de software, por esta razón es muy importante establecer modelos de trabajo que garanticen calidad de los sistemas de software.

Justificación

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El lenguaje natural ha demostrado desde hace mucho tiempo que no es el más indicado para gestionar y administrar la Ingeniería de Requisitos. Los computadores son máquinas que utilizan y procesan información desde un lenguaje matemático, por eso se les dificulta y los hace lentos el trabajo de compilación y de estructuración de respuestas en el desarrollo de software. Ellos necesitan traducir el lenguaje natural humano al lenguaje máquina con el que operan, y esto hace que no respondan a las velocidades que los usuarios quisieran. Si la investigación en la academia se acercara un poco más a la comprensión de los métodos formales, como base de la matematización de la Ingeniería de Software, sería posible que los estudiantes se capacitaran de mejor forma en los procesos de formalización del desarrollo de sistemas. De esta forma estarían aprovechando de mejor manera el hardware y aportarían al logro de la calidad del software. Este proyecto busca encontrar el panorama de la formalización de los requisitos, como una primera fase para que la academia se acerque a la tecnología de los métodos formales.

Planteamiento del problema

La calidad, fiabilidad y seguridad de los productos software son características que la Ingeniería de Software no satisface completamente. Desde que en los años 60 se promulgó la llamada crisis del software, los investigadores y la industria han intentado encontrar una solución para la misma, pero hasta el momento no se ha logrado. Los métodos formales surgieron como una iniciativa para alcanzarla, pero debido a sus exigencias en conocimiento matemático y a que los planes de estudios todavía no los incorporan ampliamente, por lo momento se limitan al desarrollo de software crítico. Se necesita con premura que la formalización empiece a permear los procesos formativos en la academia y que los estudiantes conozcan sus ventajas y potencialidades, lo mismo que sus desventajas y problemas. En este proyecto se pretende aportar a esta problemática y se espera involucrar los resultados en los planes de estudios relacionados con el ciclo de vida del software.

Objetivos

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

General: Determinar el proceso y el progreso que ha alcanzado la formalización de los requisitos.

Específicos:

Encontrar y analizar en la literatura acerca los trabajos y propuestas publicados en relación con la formalización de los requisitos.

Recoger y valorar los aportes, el proceso y el progreso de los trabajos en esta temática.

Presentar los resultados en un documento de análisis.

Organización del trabajo

Este trabajo de investigación está compuesto por los siguientes capítulos: introducción, marco teórico, metodología, resultados y discusión

En el capítulo 1 introducción se da a conocer los diversos temas que componen el proyecto de investigación como son: situación actual en las empresas dedicadas al desarrollo de software, especificación de requisitos, métodos formales y también incluye las razones por las que se debe hacer uso de la formalización de requisitos.

El capítulo 2 marco teórico que habla de los antecedentes de la formalización y la descripción de términos planteados por varios autores como: formalización, ingeniería de requisitos, requisitos, especificación formal y métodos formales, todos estos términos permitieron abordar el problema de investigación fueron cruciales en la realización del actual proyecto de investigación.

El capítulo 3 es la metodología del proyecto de investigación donde se describe cada una de las etapas de la investigación y contiene el cronograma de actividades. Finalmente, en el capítulo 4 resultados y discusión y conclusiones, se da respuesta a las preguntas de investigación y se muestra el análisis de los resultados.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEORICO

Para Lars Mathiassen y Andreas Munk (1985), las formalizaciones están relacionada tanto con los tipos de expresión como con los tipos de comportamiento del software. Este autor discute los límites a la aplicación de las formalizaciones en estos dos sentidos y los ilustra con ejemplos en el desarrollo de sistemas prácticos. También establece que las formalizaciones son valiosas en algunas situaciones, pero insuficiente en otras, y plantea que la alternativa a la utilización acrítica de las formalizaciones es que los desarrolladores analicen las situaciones en las que se encuentran, y a partir de ahí elijan una combinación de lo formal y lo informal hasta que puedan comprenderlas y dominarlas.

Vilkomir, Bowen y Ghose (2006) afirman que los requisitos reglamentarios, a diferencia de los requisitos para un sistema en particular, tienen un carácter genérico, son aplicables a una amplia gama de sistemas y son la base para la certificación o el proceso de concesión de licencias. La formalización de los requisitos está resolviendo las inconsistencias entre éstos y la interpretación de los desarrolladores de sistemas críticos. Estos autores proponen un enfoque para hacerlo, incluyendo el uso de requisitos reglamentarios formales como base para el desarrollo de métodos de evaluación del software. Argumenta que este paquete debe incluir no solamente los requisitos normativos, sino también los métodos para su evaluación. Ilustran su enfoque con ejemplos de requisitos formalizados para la protección de sistemas de control contra el acceso no autorizado y los fallos de software en modo común, usando la notación Z para la formalización.

Por su parte, Christian Webel y Reinhard Gotzhein (2007) establecen que la necesidad de la formalización de los requisitos se debe al hecho de que se necesita una descripción precisa de ellos y una amplia comprensión entre el usuario y el proveedor de los servicios de calidad. Pero los mecanismos que realizan estos procesos requieren descripciones más precisas y un alcance real de la formalización. Estos mecanismos suelen estar integrados en capas, por lo que se necesita más de un punto de vista sobre los requisitos.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Pero resulta que actualmente, esta relación rigurosa de los puntos de vista no se está logrando, y los métodos formales necesarios todavía están por definir. Se espera que en el futuro penetren mejor y de mayor forma en los procesos de la Ingeniería de Requisitos, y que los desarrolladores tengan una mejor comprensión de la formalización.

La especificación formal es el proceso de describir un sistema y sus propiedades deseadas utilizando un lenguaje con una sintaxis y la semántica definida matemáticamente Schraps, M. y Peters, M. (2014). Por eso, el objetivo del trabajo de Ram Chatterjee y Kalpana Johari (2010) tiene que ver con la formalización automatizada de la especificación de requisitos, de forma que se puedan generar fácilmente los casos de prueba formales.

Roberto Cavada y sus colegas (2010) llegan a la conclusión que la Ingeniería de Requisitos es una de las fases más importantes del proceso de desarrollo de software, y que en aplicaciones críticas es importante apoyar la validación de los requisitos con técnicas formales para identificar y eliminar los defectos. Sin embargo, los requisitos se escriben a menudo en documentos textuales y su formalización y validación no es la adecuada debido a la falta de conocimiento de los métodos formales.

En el trabajo de Post y Hoenicke (2012) se evalúa una cadena de herramientas para analizar algorítmicamente requisitos de tiempo real. De acuerdo con esta cadena de herramientas, los autores formalizan los requisitos de un sistema en lenguaje natural. Ellos experimentan la compilación automática desde fórmulas en una lógica en tiempo real. Las fórmulas se pueden comprobar de forma automática para las propiedades cuya violación indica un error en la especificación de requisitos. Presentan un estudio de viabilidad en el contexto de varios proyectos en la herramienta Bosch. Los resultados del estudio indican que el esfuerzo para formalizar los requisitos de tiempo real es aceptable; los algoritmos de análisis son computacionalmente factibles; y el beneficio (la detección de errores de especificación) parece significativo.

Li et al. (2015) concluyen que los requisitos son informales, vagos, ambiguos y muchas veces inalcanzables, por lo que el problema de la Ingeniería de Requisitos es formalizarlos y luego transformarlos a través de un proceso sistemático

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

en una especificación formal que pueda ser entregada a los diseñadores para el desarrollo. En su trabajo proponen un marco para la transformación de los requisitos informales a formales y luego a una especificación. El marco consiste en una ontología de requisitos, un lenguaje formal de modelado de requisitos para la representación de los funcionales y los no-funcionales, así como un amplio conjunto de operadores de refinamiento por el cual los requisitos incrementalmente se transforman en una especificación formal y medible. Esta propuesta incluye una metodología sistemática y herramientas de apoyo para realizar la transformación. Los resultados sugieren que la ontología y el lenguaje de modelado son adecuados para la captura de requisitos, y que la metodología es eficaz en el manejo de los requisitos en la práctica.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGIA

Este proyecto consiste en consultar en la literatura las diferentes publicaciones donde los autores proponen o experimentan la formalización de requisitos y sus resultados contienen alto rigor de credibilidad, para identificar y seleccionar dichas publicaciones se hizo uso del protocolo diseñado por Serna y Serna y está compuesto por las siguientes etapas:

1. Identificar y definir el problema de investigación: Se recopila información referente a la formalización de requisitos para identificar y formular el problema de investigación.
2. Estructurar el protocolo de búsqueda
 - Definir área temática: muestra el área temática.
 - Definir preguntas de investigación: en este ítem se plantean las preguntas de investigación que busca dar respuesta al problema planteado.
 - Definir proceso de búsqueda: consiste en definir los términos de búsqueda y las bases de datos donde se consultará la información.
3. Definir criterios para evaluar el rigor de los resultados publicados
 - Definir criterios de inclusión y exclusión: consiste en identificar criterios que serán tenidos en cuenta para aprobar o descartar un documento que contenga los términos de búsqueda definidos.
 - Definir valoración de la calidad: se validará antes de incluir un documento que contenga una fuente (factor de impacto), resultados verificables, aceptación, trayectoria del autor, aplicación, relevancia de la organización.
 - Definir la recopilación de los datos: define que datos de cada documento encontrado serán almacenados.
 - Definir el análisis de datos: proporciona una secuencia de actividades con el fin de hacer el análisis de los datos obtenidos en la investigación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. Consultar y analizar las fuentes directas: consiste en buscar en las bases de datos: IEEEExplore, ACM Digital library, ScienceDirect, SpringerLink, DOAJ y CiteSeerX los siguientes terminos: formal methods, formalizing requirements, requirements engineering, software engineering, software quality. Estas bibliotecas y terminos fueron definidos en el protocolo de búsqueda. Luego se descargarán los documentos que cumplan la valoración de calidad definida y con los siguientes criterios de inclusión y exclusión:
- Línea de tiempo 2010-2015
 - Trabajos que relacionen claramente resultados de investigación en el área de interés
 - Estudios en los que los proyectos verificaron los resultados
 - Investigaciones cuyo enfoque se expresa claramente
 - Estudios cuyo método de investigación es experimental
5. Estructurar y presentar los resultados: para este ítem se realiza recopilación de los siguientes datos de cada documento: Tipo de publicación, Editorial, País, Fecha, Clasificación temática, Enfoque de investigación (cualitativa, cuantitativa, mixta), Método de investigación (experimental, exploratoria, descriptiva, confirmatoria). Luego se relacionan las publicaciones encontradas por el área temática, con base en los resultados responder cada una de las preguntas de investigación y finalmente se realiza reporte para el proyecto de investigación.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

CRONOGRAMA DE ACTIVIDADES

Actividad	Meses				
	1	2	3	4	5
	1. Identificar y definir el problema de investigación	X			
2. Estructurar el protocolo de búsqueda	X				
3. Definir criterios para evaluar el rigor de los resultados publicados	X				
4. Consultar y analizar las fuentes directas	X	X	X		
5. Estructurar y presentar los resultados			X	X	X

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSION

Este trabajo es el resultado de una revisión de literatura, con la que se pretende encontrar respuesta a las siguientes preguntas de investigación:

- 1) ¿Qué se entiende por métodos formales?
- 2) ¿Qué se publica en la literatura en relación con la formalización de requisitos?
- 3) ¿Qué tanto se ha progresado en la formalización de requisitos?

Luego de aplicar la metodología descrita por estos autores, se conformó una muestra final de 40 trabajos relacionados, cuyos resultados sirvieron de base para encontrar las respuestas que se detallan a continuación.

Los métodos formales

Yan (2011) afirma que los métodos formales son técnicas basadas en las matemáticas, a menudo con el apoyo de herramientas de razonamiento, que pueden ofrecer una manera rigurosa y eficaz para modelar, diseñar y analizar sistemas informáticos. Adicionalmente, Seceleanu (2011), Ambrosio, y Andrade (2011), Kaur, Gulati y Singh (2012) y Barbosa y Lumpe (2014) coinciden en que los métodos formales proporcionan una estructura matemática dentro de la cual, de manera sistemática, se utiliza para especificar, desarrollar y verificar un sistema. Para Kaur, Gulati y Singh (2012) permiten desarrollar un sistema informático que es más completo, coherente e inequívoco, y, por otro lado, para González y Cabot (2014) evitan la introducción de imprecisiones o ambigüedades en el proceso.

Hay que tener en cuenta que la noción de *formal* en este contexto debe entenderse en términos generales como *lógica/matemática* (Gruner, 2011). Adicionalmente, Barlas, Koletsos y Stefaneas (2012) manifiestan que estos principios matemáticos no necesitan conocimientos de lenguajes y que pueden ser fácilmente entendidas en un contexto internacional. Al exponer algunos puntos de vista sobre los métodos formales, You, Li y Xia (2012) reflexionan que desarrollar software correcto y confiable ha sido un problema

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

continuo que debe resolverse con urgencia, y que una manera eficaz de lograrlo es la teoría formal. Porque ofrece una posibilidad para crear software con pocos o ningún defecto (Jeffery et al., 2015).

El uso de métodos formales permite plantear de manera clara la especificación de un sistema, generando modelos que definen el comportamiento en términos de *qué debe hacer* y no de *cómo lo hace* (Ambrosio y Andrade, 2011). Gracias al correcto proceso de especificación, se puede verificar propiedades derivadas de cada módulo mediante técnicas de razonamiento asociadas a los modelos formales, tales como probadores de teoremas y verificadores de modelos. Al recopilar opiniones de diferentes autores sobre métodos formales, se encontró que: generalmente son costosos de aplicar (Shirali y Shirali, (2010), especialmente para la formación de los ingenieros; son demasiado lentos en su aplicación (Gruner, 2011); incrementan la calidad y la fiabilidad de un diseño y muestran su relación con los problemas prácticos y su potencial para el futuro, por otro lado, su aplicación exige que sea fácil de usar y que se tenga herramientas de verificación eficientes (Seceleanu, 2011). Por otro lado, los métodos formales pueden ayudar a resolver algunos de los problemas que enfrenta la industria del software actualmente, tales como la insatisfacción de los clientes por incapacidad para cumplir con los requisitos y altos costos de apoyo (Shirali y Shirali, 2010). El objetivo principal de la Ingeniería del Software es permitirles a los desarrolladores crear sistemas que operen de forma fiable (Kaur, Gulati y Singh, 2012), y los métodos formales están destinados a proporcionar los medios para lograr una mayor precisión en el pensamiento y la documentación de esta ingeniería.

Las principales ventajas de los métodos formales son: las especificaciones formales son correctas, coherentes, pueden ser probadas y verificadas; la ambigüedad en la especificación se evita automáticamente (Kaur, Gulati y Singh, 2012); pueden garantizar la seguridad y la confiabilidad de los sistemas software (You, Li y Xia, 2012); se reduce en gran medida el error en el trabajo (You, Li y Xia, 2012); y se eliminan los errores de diseño o inconsistencias antes de su implementación. Esto es esencial para el desarrollo de sistemas

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

fiables, especialmente en aplicaciones de software crítico (Lockhart, Purdy y Wilsey, 2014), los cuales presentan menos errores durante la verificación y a menudo requieren tiempos más cortos de desarrollo y consumen menos recursos económicos (Mandrioli y Milano, 2015). Estas ventajas se logran por el uso de las matemáticas, porque son un medio para desarrollar un modelo exacto que puede describir el objeto y el funcionamiento de forma breve y precisa. Además, porque pueden representar unidades abstractas y la especificación del sistema es coherente con las necesidades de clientes y usuarios. Por otro lado, las matemáticas ofrecen una manera de mostrar la contradicción y la imperfección en la especificación, a la vez que la coherencia entre el diseño y especificación (You, Li y Xia, 2012).

En el otro extremo, también se encuentran limitaciones a los métodos formales. De acuerdo con Kaur, Gulati y Singh (2012), son difíciles de aprender y usar y, debido a la falta de modelos de desarrollo y herramientas, los problemas complejos son difíciles de manejar. Para Mandrioli y Milano (2015) nos encontramos con una actitud general asocial contra el razonamiento riguroso de los métodos formales. Lockhart, Purdy y Wilsey (2014) concluyen que, desafortunadamente, actualmente los métodos formales son complejos y costosos para ser utilizados ampliamente en los sistemas software, porque el refinamiento de una especificación formal es un proceso complejo, lento y no está bien apoyado por herramientas. Otra cuestión importante, pero negativa en el caso de los métodos formales, es que la mayoría de programadores los perciben como una teoría inútil que no tiene ninguna relación con lo que hacen (Mandrioli y Milano, 2015). De acuerdo con estos investigadores, no hay manera más rápida de perder la atención en una audiencia programadores que a mostrarles una fórmula matemática. Por desgracia, uno de los problemas centrales es que no ha habido avances comparables en métodos formales y, aunque surgen nuevos lenguajes y nuevas lógicas, los errores en el diseño de los programas, que provienen desde 1967, todavía se pueden encontrar en el software de hoy (Parnas, 2010). Serna (2010) coincide en afirmar que, a pesar de sus ventajas significativas, los

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

métodos formales no son ampliamente utilizados en el desarrollo de software industrial, y la academia todavía no demuestra amplio interés para incluirlos en los planes de estudios.

La formalización de requisitos

Los requisitos son considerados como la piedra fundamental sobre la que se pueden desarrollar las soluciones software, y actualmente existe una creciente demanda de enfoques más rigurosos y sistemáticos para formalizarlos (Pandey y Batra, 2013). Serna y Serna (2014) argumentan que los métodos formales se utilizan hoy para modelar complejos sistemas críticos de seguridad, pero poco se trabaja en la formalización de los requisitos desde las primeras etapas de la Ingeniería de Requisitos. La representación más utilizada para formalizar los requisitos son los lenguajes de especificación formal (Pandey y Batra, 2013). Con este tipo de especificación, los desarrolladores comprenden mejor el sistema y eluden las ambigüedades, los flujos, las omisiones y las inconsistencias de los requisitos. Además, la especificación es un importante mecanismo de comunicación entre los clientes y los diseñadores, entre los diseñadores y ejecutores, y entre los ejecutores y los probadores (Krad, 2011). Pero, las especificaciones formales no son documentos que se escriben una vez y generalmente no se logran al principio del proceso de desarrollo de software. Se necesita tiempo para crear una primera versión de utilidad que, luego de mucho esfuerzo y revisiones, permita desarrollar una especificación cercana a las necesidades que los clientes tienen en mente (Yan, 2011), y Bollin y Rauner (2014) sostienen que un buen nivel de comprensión es una cuestión clave como atributo de calidad. Para Wolff (2012), con el uso de un lenguaje de especificación formal, el sistema puede ser descrito con precisión en cuanto a funcionalidad, concurrencia, integridad y exactitud. Esto significa que las propiedades de un sistema se pueden analizar sin tener que ejecutarlo realmente.

Al utilizar la especificación formal, las propiedades del sistema se describen utilizando un lenguaje con una sintaxis y una semántica definidas matemáticamente (Vilkomir, Bowen y Ghose, 2006). Algunos lenguajes formales, tales como Z, B y VDM, se centran en especificar el comportamiento de los sistemas secuenciales, donde los estados se describen en

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

estructuras matemáticas, tales como conjuntos, relaciones y funciones (You, Li y Xia, 2012); mientras que métodos como CSP, CCS, Statecharts, Lógica Temporal, y Autómatas, se centran en la especificación de los comportamientos del sistema en términos de secuencias, árboles u órdenes parciales de eventos (Schrapf y Peters, 2014). El uso de lenguajes formales como Z y B puede mejorar la confianza del usuario en el sistema y sus impactos en el uso del mismo (Azeem, Ahsan y Noreen, 2014). Tamrakar y Sharma (2015) afirman que Z, B y VDM son lenguajes de especificación formal utilizados para especificar las necesidades del usuario en un lenguaje matemático, cuyo producto puede ser probado y verificado automáticamente. Z trabaja en alta abstracción a nivel del sistema y proporciona una sólida base para el diseño del mismo. Mediante el uso de este lenguaje, no solamente se descubren más errores en la especificación sino también en las fases de pruebas y mantenimiento. Esto es conveniente porque con la ingeniería tradicional, el costo de corregir errores en las fases posteriores es mucho mayor que en las primeras (Azeem, Ahsan y Noreen, 2014); además, Z es una manera de descomponer una especificación en pequeñas partes llamadas esquemas. Por su parte, B es uno de los métodos formales más conocidos. Se basa en la lógica de primer orden, la teoría de conjuntos, la aritmética de enteros y las sustituciones generalizadas; se utiliza para el diseño de software desde los requisitos funcionales y permite producir casos de prueba que demuestran la exactitud y la consistencia del modelo software aplicado. El objetivo de B es obtener un producto probado y fiable (de Sousa et al., 2010). Mientras que VDM se utiliza para demostrar la equivalencia de los conceptos del lenguaje (Tamrakar y Sharma, 2015).

De acuerdo con Pandey y Batra (2013), en la actual era digital, las empresas se enfrentan al desafío de liberar proyectos software de calidad, a tiempo y dentro del presupuesto, pero la realidad es que se entrega con errores, la falta de funcionalidad y a veces con sobrecostos. El costo extra se genera debido a errores en la especificación de requisitos, que pueden costar mucho tiempo y dinero corregirlos cuando se detectan en las fases tardías del ciclo de vida. Por lo tanto, en la especificación de requisitos se debe seleccionar

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

una metodología formal para abordar la fiabilidad durante la Ingeniería de Requisitos y el diseño, porque los métodos formales permiten desarrollar los errores antes de la liberación del producto (Lockhart, Purdy y Wilsey, 2014).

Los errores de especificación pueden ser reducidos drásticamente mediante el uso de métodos formales y, en consecuencia, el ingeniero de software puede crear una especificación más completa, coherente e inequívoca que con los métodos convencionales (Kaur, Gulati y Singh, 2012; Azeem, Ahsan y Noreen, 2014). Por su parte, Bollin y Rauner (2014) manifiestan que una buena especificación formal es sintáctica y semánticamente correcta, que permite un mapeo sin pérdidas entre todos los conceptos de la especificación y el modelo mental del sistema especificado; también agrega que debe ser completa, coherente y adecuada, y tener en cuenta que la facilidad de comprensión es un requisito esencial para decidir sobre su corrección semántica. No existe una herramienta informática que pueda garantizar la corrección completa de un modelo de computador (Tamrakar y Sharma, 2015). Por lo tanto, aunque la especificación esté escrita utilizando cualquiera de los lenguajes formales, siempre contiene un alto potencial de errores. Significa que el arte de escribir una especificación formal no asegura que el sistema desarrollado sea coherente, correcto y completo. Por otro lado, si la especificación se comprueba y se analiza con el apoyo de herramientas automatizadas, aumenta la confianza sobre el sistema con la posible identificación de los errores potenciales, si existen, en la sintaxis y la semántica de la descripción formal. Una de las ideas más difundidas sobre el uso de métodos formales es su casi nula aplicación en la industria software. Si bien su adopción ha sido lenta, existen casos de importantes empresas que han tenido éxito al aplicarlos en proyectos reales (Ambrosio y Andrade, 2011).

En resumen, para conocer qué se publica en la literatura en relación con la formalización de requisitos, en la Tabla 1 se presentan los resultados de esta investigación. La línea de tiempo de observación es a partir de 2010. Aunque es conocido que la mayor parte del trabajo en esta área se presentó en la segunda mitad del siglo pasado, el objetivo de esta investigación era verificar su progreso actual. Los trabajos

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

de la muestra final se clasificaron de acuerdo con el enfoque primario presentado. Se aclara que muchos autores presentan resultados en más de una tipología, es decir, su investigación puede ser teórica y al mismo tiempo experimental.

- ☐ *Investigación teórica*: artículos que contienen definiciones o descripciones acerca de la formalización de requisitos.
- ☐ *Investigación experimental*: demuestra resultados a partir de experimentos en laboratorio.
- ☐ *Aplicación práctica*: describen y aplican métodos, técnicas o procedimientos de formalización de requisitos en casos de estudio.
- ☐ *Aplicación industrial*: son los artículos donde la formalización de requisitos se aplica en casos reales.

Tabla 1. Qué se publica en la literatura acerca de la formalización de requisitos

Medio		Teórica	Experimental	Práctica	Industrial
Revista	66%	63%	34%	20%	23%
Conferencia	34%				

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Progreso del trabajo en la formalización de requisitos

El progreso de la formalización de requisitos será definido como acelerado o lento dependiendo del porcentaje de aplicaciones en la industria.

En la tabla 2 se clasifican los resultados de la investigación y se muestran su participación porcentual.

Tabla 2. Progreso de la formalización de requisitos

Teórica	Experimental	Práctica	Industrial
63%	34%	20%	23%

En la Tabla 3 se presenta un resumen general de los artículos consultados, donde se puede visualizar el desarrollo de las publicaciones de la formalización de requisitos en el último quinquenio.

Tabla 3. Desarrollo de la formalización de requisitos

Autor	Desarrollo
Bhavsar (2010)	Presenta un uso de los métodos formales en la formalización de requisitos centrados en el usuario. De acuerdo con el autor, los resultados son prometedores en cuanto al mejoramiento de la calidad del producto software.
De Sousa et al. (2010)	Analiza el método B para la especificación de requisitos y describe los casos de uso y las propiedades de seguridad. Aunque su aplicación es sencilla, es un aporte hacia la masificación de los métodos formales en la especificación de requisitos.
Honghao (2010)	Propone una metodología para formalizar la Ingeniería de Requisitos y el diseño, orientada a analizar y gestionar requisitos. Esta investigación brinda luces para demostrar la utilidad de los métodos formales en la Ingeniería de Requisitos.
Hassan et al. (2010)	Plantea el análisis y el diseño formal de la seguridad de los requisitos utilizando especificaciones en B. Los resultados demuestran efectividad y capacidad en la producción de software seguro.
van der Poll (2010)	Describe los métodos formales más comunes y la importancia de aplicar la especificación formal en los primeros pasos del ciclo de vida. Enfrenta las posiciones de los críticos y defensores de los métodos formales y presenta ejemplos de historias de éxito en la industria.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Singh (2010)	Utiliza la notación Z para formalizar los requisitos, buscando protección y seguridad contra amenazas internas, en un caso de Estudio de éxito.
Cimatti et al. (2010)	Desarrolla una metodología y técnicas para formalizar y validar requisitos en aplicaciones críticas de seguridad. Aplican el resultado en un caso de estudio con resultados mejorados.
Serna (2010)	Describe la aplicación de la especificación formal de requisitos y el mejoramiento que se logra en la calidad del producto final. Propone la incorporación de los métodos formales en los contenidos curriculares de los programas en Ciencias
Barringer et al. (2011)	Presenta un método práctico para usar especificaciones formales en pruebas de registro en Mars Science Laboratory. Demuestra las ventajas de utilizar un lenguaje de especificación formal.
Bishop, Hay y Nance (2011)	Integra los conceptos de los métodos formales desde el inicio de la programación y, mediante una didáctica simple, muestra cómo hacer una especificación formal y qué modelos se pueden utilizar.
Fernández et al. (2011)	Es una revisión a la aplicación de los métodos formales en proyectos reales en la industria. Presenta los diversos casos de éxito y describe la situación actual en México en cuanto a la aplicación de métodos
Serna (2011)	Hace un recorrido histórico por los métodos formales y describe sus aplicaciones y beneficios en la formalización de requisitos. Propone que la investigación en esta área se debe orientar a aplicaciones prácticas y a buscar reducir los costos de su utilización.
Ibrahim, Alagar y Mohammad (2011)	Aplica una base formal para la especificación y verificación de requisitos, presenta una definición formal y una teoría de composición formal para la Ingeniería de Requisitos. Aplica la verificación formal y describe los resultados.
Yan (2011)	Describe la aplicación de la especificación formal en Communications-Based Train Control (CBTC), obteniendo una descripción profunda y exacta del sistema, y lenguaje Z en el desarrollo. Concluye que al usar métodos formales se gana confianza para asegurar el sistema.
Ammar y Abdallah (2011)	Presenta un nuevo enfoque para la especificación y verificación formal basado en la reescritura la lógica. Luego lo aplica en caso de estudio con resultados prometedores en mejoramiento de la calidad.
Kaur, Gulati y Singh (2012)	Estudio comparativo entre los métodos de especificación formal y descripción de las razones de por qué usarlos en los procesos industriales.
You, Li y Xia (2012)	Muestra los logros y problemas de los métodos formales y valido su aplicación para garantizar la seguridad y estabilidad del sistema.
Barlas, Koletsos y Stefaneas (2012)	Presenta una comparación entre la especificación de requisitos tradicional y la formal. Con la segunda logra mayor comprensión, elimina ambigüedades y obliga a una mejor precisión de la especificación. Muestra aplicaciones y beneficios de su uso en la
Wolff, S. (2012)	Utiliza la experiencia en la industria para combinar el desarrollo ágil Scrum con los métodos formales, pero desde una conceptualización teórica. Presenta una evaluación y discusión acerca de los beneficios de utilizarlos para formalizar los requisitos.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Bollin (2013)	Desde un marco teórico concluye que el proceso de la formalización de los requisitos es un problema de adaptación transcultural, analiza los pros y los contras del mismo y da a conocer un modelo refinado para un proceso de desarrollo de software formal.
Serna y Serna (2013)	Investigación teórica en la que se concluye que el método formal ha alcanzado significativos avances, lo que vislumbra aplicaciones ambiciosas en el futuro para apoyar rigurosos y coherentes planes de estudios en Ciencias Computacionales.
Pandey y Batra (2013)	Discute los diversos lenguajes de especificación formal y analiza sus fortalezas y debilidades. Realiza recomendaciones para el uso de métodos formales, especialmente en la Ingeniería de Requisitos.
Atlee et al. (2013)	Mediante una serie de recomendaciones para formalizar los requisitos, aporta al logro práctico de la selección de una técnica específica. Su objetivo es mejorar la aceptación de los métodos formales en el modelado de requisitos.
Serna y Serna (2013a)	La especificación formal todavía tiene usos limitados, pero la comunidad tiene una comprensión diferente acerca de su utilidad y necesidad. Hasta el momento, su desarrollo se focaliza principalmente en evaluar las herramientas relacionadas.
Chen (2013)	Analiza el dominio de los métodos formales para identificar defectos y disminuir fracasos. Aplica la formalización en la especificación, el modelado y la verificación.
Chan, Hexel y Wen (2013)	Propone una metodología de modelado de requisitos para convertir los informales en formales. Con ellos busca eliminar ambigüedades y defectos, refinar y perfeccionar el sistema y servir como un medio de comunicación entre las partes interesadas.
Diallo (2013)	Hace un estudio de los lenguajes de especificación formal, describe los modelos que se pueden aplicar y realiza un análisis general. Realiza una experimentación con estadística de depuración en un estudio de caso.
Noaman, Alsmadi y Jaradat (2013)	Aplica la especificación formal por medio de Z para potencializar la seguridad del sistema y reducir las amenazas. Los resultados son prometedores para ampliaciones y elaboraciones futuras.
Lockhart, Purdy y Wilsey (2013)	Demuestra la utilidad de los métodos formales para la especificación de requisitos en sistemas de seguridad críticos. La especificación formal reduce la ambigüedad del diseño, ayuda a probar la consistencia, e incrementa la confianza y el rendimiento.
Wu, Dong y Hu (2013)	Propone reglas de transformación desde el modelo B y describe los requisitos con máquinas abstractas. Para el autor, la especificación formal es un proceso intermedio entre la especificación de requisitos y la
Serna y Serna (2014a)	Realiza una revisión de la literatura, hace un recorrido por la esencia, la función, el uso y los inconvenientes de las técnicas de especificación formal y analiza criterios de valoración y evaluación a sus debilidades. Busca que la especificación formal se afiance como actividad básica de investigación y formación en la academia.
Bollin y Rauner (2014)	Muestra una serie de recomendaciones para realizar especificaciones formales, e identifica que la facilidad de lectura de una especificación es un factor clave para mejorar la calidad del software.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Schraps y Peters (2014)	Aplica gramática formal con anotaciones semánticas para formalizar requisitos, de tal forma que puedan ser analizados y procesados por el computador. Los requisitos formulados pueden ser analizados y procesados en las primeras etapas de desarrollo.
Serna y Serna (2014b)	Afirma que muchos de los problemas desafiantes en la construcción de sistemas requieren el apoyo formal para su modelado y análisis. En los procesos de investigación y aplicación de las Ciencias Computacionales existe un número creciente de aplicaciones. Pero todavía no constituyen una parte integral de los procesos formativos en pregrado y posgrado.
Azeem et al. (2014)	Utiliza especificación formal Z para mejorar la calidad y confianza del sistema. Muestra trabajos relacionados y aplica una propuesta en un estudio de caso en salud.
Li et al. (2014)	Aplica objeto-Z para describir sistemas complejos. Demuestra su utilidad en un caso de estudio de un sistema de suministro de gasolina. Los resultados obtenidos alientan el uso de la especificación formal.
Serna y Serna (2014c)	Describe aspectos relevantes de los métodos formales y realiza un marco del futuro de la investigación en el área. Los resultados invitan a la reflexión de este componente de las Ciencias Computacionales y de la necesidad de tener una comunidad más amplia y sólida dedicada a fomentar y aplicar la formalización de los requisitos.
Jeffery (2015)	Realiza un estudio empírico para encontrar la productividad de los proyectos que utilizan métodos formales, específicamente al formalizar los requisitos. Identifica una serie de preguntas sobre los resultados en <u>productividad de estos proyectos</u> .
Tamrakar y Sharma (2015)	Compara tres métodos para especificar formalmente: Z, B y VDM. Usar especificaciones formales no asegura un sistema completamente correcto, pero se aumenta la confianza en el mismo.
Singh y Yadav (2015)	Utiliza el método Event-B para escribir especificaciones formales, con el fin de garantizar la comprensión de los límites en los que un algoritmo puede ser usado. Para el autor, la especificación, la validación y la verificación formales con la clave para obtener un mejor
Walter (2015)	Realiza una representación de formalización de requisitos por medio de diagramas SysML usando la semántica RSL. Formaliza únicamente los requisitos no-funcionales y ese diseño lo aplica a un caso de estudio en el área automotriz.

ANÁLISIS DE RESULTADOS

De acuerdo con la revisión de la literatura en esta investigación, el trabajo en formalización de requisitos ha tenido un progreso lento, en la tabla 2 se puede evidenciar que la mayoría de los trabajos tenidos en cuenta para la presente investigación se enfocan más en definiciones teóricas que son representadas por el 63% y solo el 23% de estos en aplicación en la industria. Algunos autores han analizado este hecho y presentan sus conclusiones. Entre las causas se destacan: falta de formación en

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

la academia (Serna, 2011), falta de aceptación en la industria (Serna, 2012), no hay demostraciones realmente importantes de sus ventajas, los costos, falta de personal capacitado, los ingenieros les tienen fobia a las matemáticas (Atlee, J. et al. 2013), la industria no quiere experimentar con principios que no tienen comprobación suficiente y en la última década la investigación en métodos formales ha perdido interés y patrocinio (Serna y Serna, 2014).

Los investigadores de mediados del siglo pasado estaban convencidos que la única manera de mejorar la calidad de los productos software era a través de la matematización de la Ingeniería del Software (van der Poll, 2010), pero a medida que sus resultados comenzaron a publicarse, la industria comenzó a encontrar y colocar barreras al progreso de este trabajo. Solamente el software crítico los arropó como tabla de salvación para solucionar sus problemas de fiabilidad y seguridad, y son diversos los casos de éxito en estos desarrollos. El problema surge cuando las empresas de desarrollo de software comercial se enfrentan a situaciones de incumplimiento, que incrementan los costos del proceso y que las obligan a dedicar menos tiempo a la estructura misma de las actividades. La decisión es disminuir la mayor cantidad de tiempo posible y, como la forma tradicional de realizar la verificación y validación es como una barrera de contención al final del ciclo de vida, esta fase es la que se recorta para encontrar algo de tiempo (You, Li y Xia, 2012). El estudio sobre la formalización de requisitos se ha relegado a unos pocos investigadores que convencen a sus estudiantes de posgrado para que trabajen con ellos. Con el inconveniente analizado antes de que los profesionales le temen a todo lo que tenga que ver con matemáticas. La recomendación es a que las universidades incluyan cursos relacionados en los planes de estudios de los pregrados en Ciencias Computacionales, de tal forma que los estudiantes se interesen y profundicen en su asimilación y comprensión.

De acuerdo con los autores analizados en esta investigación, la formalización de requisitos se investiga de forma mayoritariamente teórica, y de esta manera no se logrará convencer a la industria de su aplicación. Puede ser que la etapa de la teorización ya haya sido superada en los muchos trabajos que se presentaron el siglo pasado. La necesidad

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

actual es masificar los métodos formales y encontrar la forma de aplicarlos en la matematización de la Ingeniería de Software, pero con costos que estén al alcance de las empresas desarrolladoras de software. De esta manera será posible reavivar el interés en esta área de trabajo y de encontrar el patrocinio necesario para ejecutar la investigación necesaria. Si el objetivo es lograr por fin superar la crisis del software, los métodos formales deben ser el centro de desarrollo. Porque sus bondades ya han sido suficientemente demostradas y porque el lenguaje con el que funciona el computador es el matemático.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES

Los resultados demuestran que el progreso y desarrollo de la formalización de requisitos son lentos, y que en los últimos años el interés por darle continuidad ha disminuido. Aunque se acepta que los métodos formales son una herramienta útil y necesaria para superar la crisis de la calidad del software, muy pocos investigadores quieren abordar la formalización de requisitos.

Dentro de las investigaciones obtenidas se puede evidenciar la importancia de la aplicación de los métodos formales durante el desarrollo de un producto de software, ya que evita ambigüedades y permite obtener productos de software más precisos y seguros, pero cabe resaltar que en los últimos 5 años la aplicación de la formalización de requisitos en la industria ha sido muy poca por factores como: desconocimiento de la formalización de requisitos por parte de los desarrolladores, la búsqueda de desarrollos de software ágiles o hechos en el menor tiempo posible y la falta de publicaciones por parte de las empresas que aplican la formalización de requisitos en sus desarrollos.

La industria necesita demostraciones de los beneficios que representa la formalización, pero hasta ahora no tiene inclinación directa por apoyar las iniciativas que en este sentido proponen la comunidad. Solamente en el desarrollo de sistemas críticos se aprecia una amplia participación de las empresas de desarrollo y la industria, porque el objetivo es que funcionen sin poner en riesgo la vida humana o las inversiones económicas. Se necesita mayor trabajo para reducir los costos actuales de la formalización, pero, sobre todo, es urgente contar con el personal capacitado para aplicarla y darle continuidad.

Esta es una Sociedad software-dependiente, porque muy pocas de sus actividades están por fuera del ámbito de este desarrollo tecnológico. Pero los productos que se liberan o entregan a los usuarios todavía no satisfacen aspectos como fiabilidad y seguridad. Las matemáticas ofrecen la posibilidad de superar este problema y su representación en los métodos formales es una alternativa prometedora.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Ambrosio, M. y Andrade, G. (2011). Métodos formales aplicados en la industria del software. *Temas de Ciencia y Tecnología* 15(43), pp. 3-12.
- Ammar, B, y Abdallah, K. (2011). Towards the formal specification and verification of multi-agent based systems. *International Journal of Computer Science Issues* 8(4), pp. 200–210.
- Atlee, J. et al. (2013). Recommendations for improving the usability of formal methods for product lines. *Proceedings 1st FME Workshop on Formal Methods in Software Engineering* (pp. 43–49). San Francisco, USA.
- Azeem, M. et al. (2014). Specification of e-Health system using Z: A motivation to formal methods. *Proceedings International Conference for Convergence of Technology* (pp. 1–6). Pune, India.
- Azeem, M., Ahsan, M. y Noreen, K. (2014). Specification of e-Health System using Z: A motivation to formal methods. *Proceedings International Conference for Convergence of Technology* (pp. 1–6). Pune, India.
- Barbosa, L y Lumpe, M. (2014). Formal aspects of component software. *Lecture Notes in Computer Science* 7684, pp. 253–254.
- Barlas, K., Koletsos, G. y Stefanias, P. (2012). Extending standards with formal methods: Open document architecture. *Proceedings International Symposium on Innovations in Intelligent Systems and Applications* (pp. 1-5). Trabzon, Turkish.
- Barringer, H. et al. (2011). An entry point for formal methods : Specification and analysis of event logs. *Proceedings Workshop on Formal Methods for Aerospace* (pp. 16–21). New Mexico, USA.
- Bhavsar, M. (2010). Analysis of Multiagent Based Interactive Grid Using Formal Methods - A Reliable Approach. *Proceedings 3rd International Conference on Emerging Trends in Engineering and Technology* (pp. 462-466). Goa, India.
- Bishop, M., Hay, B. y Nance, K. (2011). Applying formal methods informally. *Proceedings of the Annual Hawaii International Conference on System Sciences. Proceedings Annual Hawaii International Conference on System Sciences* (pp. 1-8). Kauai, USA.
- Boehm, B. (1984). Model and metrics for software management and engineering. IEEE, USA.
- Boehm, B. (1987). Industrial software metrics top 10 list. *IEEE Software* 4(5), pp. 84–85.
- Bollin, A. (2013). Do you speak Z? Formal methods under the perspective of a cross-cultural adaptation problem. *Proceedings 1st FME Workshop on Formal Methods in Software Engineering* (pp. 8-14). San Francisco, USA.
- Bollin, A. y Rauner, D. (2014). Formal specification comprehension the art of reading and writing Z. *Proceedings 2nd FME Workshop on Formal Methods in Software Engineering* (pp. 3–9). Hyderabad, India.
- Chalupnik, M., Wynn, D. y Clarkson, J. (2009). Approaches to mitigate the impact of uncertainty in development processes. *Proceedings 17th International Conference on Engineering Design* (pp. 1-12). Stanford, USA.
- Chan, L., Hexel, R. y Wen, L. (2013). Rule-based behaviour engineering: Integrated, intuitive formal rule modelling. *Proceedings 22nd Australian Software Engineering Conference* (pp. 20-29). Melbourne, Australia.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Chatterjee, R. & Johari, K. (2010). A simplified and corroborative approach towards formalization of requirements. In Ranka, S. et al. (Eds.), IC3 2010, pp. 486-496. Berlin: Springer.
- Chatterjee, R. y Johari, K. (2010). A simplified and corroborative approach towards formalization of requirements. *Communications in Computer and Information Science* 94, pp. 486-496.
- Chen, X. (2013). Research on the implementation of internal control in enterprise information system by domain analysis and formal methods - A case study of sales activities internal control under chinese enterprise environment. *Proceedings Fourth World Congress on Software Engineering* (pp. 171-175). Hong Kong, China.
- Cimatti, A. et al. (2010). Formalization and Validation of Safety-Critical Requirements. *Proceedings Workshop on Formal Methods for Aerospace* (pp. 68-75). Eindhoven, Netherlands.
- Dahlstedt, A. y Persson, A. (2003). Requirements interdependencies - Moulding the state of research into a research agenda. *Proceedings Ninth International Workshop on Requirements Engineering* (pp. 71-80). Klagenfurt/Velden, Austria.
- de Sousa, T. et al. (2010). Automatic analysis of requirements consistency with the B method. *Software Engineering Notes* 35(2), pp. 1-4.
- Diallo, R. (2013). The need for usable formal methods in verification and validation. *Proceedings Winter Simulation Conference* (pp. 1257-1268). Washington, USA.
- Fantechi, A. (1994). Assisting requirement formalization by means of natural language translation. *Formal Methods in System Design* 4(3), pp. 243-263.
- Fernández, C. et al. (2011). Métodos formales aplicados en la industria del software. *Temas de Ciencia y Tecnología* 15(43), pp. 3-12.
- González, C. y Cabot, J. (2014). Formal verification of static software models in MDE: A systematic review. *Information and Software Technology* 56(8), pp. 821-838.
- Gruner, S. (2011). Editorial: Special section on formal plus agile methods. *ACM Software Engineering Notes* 36(4), pp. 26.
- Hassan, R. et al. (2010). Formal analysis and design for engineering security automated derivation of formal software security specifications from goal-oriented security requirements. *IET Software* 4(2), pp. 149-160.
- Heimdahl, M. y Leveson, N. (1995). Completeness and consistency analysis of state-based requirements. *IEEE Transactions on Software Engineering* 22(6), pp. 3-14.
- Heitmeyer, C., Jeffords, R. y Labaw, B. (1996). Automated consistency checking of requirements specifications. *ACM Transactions Software Engineering and Methodology* 5(3), pp. 231-261.
- Honghao, G. (2010). Based on formal methods in trustable software requirements engineering. *Proceedings International Conference on Internet Technology and Applications* (pp. 1-4). Wuhan, China.
- Ibrahim, N., Alagar, V. y Mohammad, M. (2011). Specification and Verification of Context-dependent. *Proceedings of the 20th international conference on World Wide Web* (pp. 17-33). Hyderabad, India.
- Jeffery, R. et al. (2015). An empirical research agenda for understanding formal methods productivity. *Information and Software Technology* 60, pp. 102-112.
- Jureta, I., Faulkner, S. y Schobbens, P. (2008). Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering* 13, pp. 87-115.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Kaur, A., Gulati, S. y Singh, S. (2012). A comparative study of two formal specification languages: Z-notation & B-method. Proceedings Second International Conference on Computational Science, Engineering and Information Technology (pp. 524-531). Coimbatore, India.
- Krad, H. (2011). Formal methods and automation for system verification. Proceedings 4th International Conference on Modeling, Simulation and Applied Optimization (pp. 1–12). Kuala Lumpur, Malaysia.
- Li, F. et al. (2015). From stakeholder requirements to formal specifications through refinement. In Fricker, S. & Schneider, K. (Eds.), Requirements Engineering: Foundation for Software Quality, pp. 164-180. Berlin: Springer.
- Li, Y et al. (2014). Specifying complex systems in object-Z: A case study of petrol supply systems. Journal of Software 9(7), pp. 1707–1717.
- Lockhart, J., Purdy, C. y Wilsey, P. (2013). Formal methods for safety critical system specification. Proceedings 57th International Midwest Symposium on Circuits and Systems (pp. 201-204). College Station, USA.
- Mandrioli, D. y Milano, P. (2015). On the heroism of really pursuing formal methods - Title inspired by Dijkstra's "On the Cruelty of really Teaching Computing Science". Proceedings of the Third FME Workshop on Formal Methods in Software Engineering (pp. 1–5). Austin, USA.
- Mathiassen, L. & Munk, A. (1985). Formalization in systems development. Proceedings International Joint Conference on Theory and Practice of Software Development on Formal Methods and Software, pp. 101-116. Berlin, Germany.
- Mathiassen, L. y Munk, A. (1985). Formalization in systems development. Lecture Notes in Computer Science 186, pp. 101-116.
- Morimoto, et al. (2008). Classification, formalization and verification of security functional requirements. Lecture Notes in Computer Science 4910, pp. 622-633.
- Noaman, M., Alsmadi, I. y Jaradat, A. (2013). The specifications of E-Commerce secure system using Z language. The Research Bulletin of Jordan ACM II(III), pp. 127-131.
- Pandey, S. y Batra, M. (2013). Formal methods in requirements phase of SDLC. International Journal of Computer Applications 70(13), pp. 7–14.
- Parnas, D. (2010). Really rethinking "formal methods". Computer 43 (1), pp. 28–34.
- Peres, F, Yang, J. y Ghazel, M. (2012). A formal framework for the formalization of informal requirements. The International Journal of Soft Computing and Software Engineering 2 (8), pp. 14-27.
- Post, A. & Hoenicke, J. (2012). Formalization and analysis of real-time requirements: A feasibility study at BOSCH. Lecture Notes in Computer Science 7152, pp. 225-240.
- Post, A., Hoenicke, J. y Podelski, A. (2011). Vacuous of real-time requirements. Proceedings 19th IEEE International Requirements Engineering Conference (pp. 153–162). Trento, Italy.
- Post, A., Hoenicke, J. y Podelski, A. (2011a). Rt-inconsistency: A new property for real-time requirements. Lecture Notes in Computer Science 6603, pp. 34–49.
- Post, A., Menzel, I. y Podelski, A. (2011b). Applying restricted English grammar on automotive requirements - Does it work? A case study. Lecture Notes in Computer Science 6606, pp. 166–180.
- Roberto Cavada, R. (2010). OthelloPlay – A Plug-in based tool for requirement formalization and validation. Proceedings 1st Workshop on Developing Tools as Plug-ins, pp. 59. Waikiki, USA.
- Schraps, M. y Peters, M. (2014). Semantic annotation of a formal grammar by Semantic Patterns. Proceedings 4th International Workshop on Requirements Patterns (pp. 9–16). Karlskrona, Sweden.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Seceleanu, C. (2011). Formal methods applied in industry : Success stories, limitations, perspectives – panel introduction. Proceedings IEEE 35th Annual Computer Software and Applications Conference (pp. 448-449). Munich, Germany.
- Serna, A.A. (2012). Formal Methods in industry. Los Métodos Formales en la Industria. Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS) 2(2), pp. 44-51.
- Serna, M.E. & Serna, A.A. (2014). Formal specification in context: Current and future. Ingeniare - Revista Chilena de Ingeniería 22(2), pp. 243-256.
- Serna, M.E. (2010). Formal methods and Software Engineering. Revista Virtual Universidad Católica del Norte 30, pp. 158-184.
- Serna, M.E. (2011). Formal Methods: Perspective and future application. Memorias III Jornadas de Investigación de la Facultad de Ingenierías (pp. 64-68). Medellín, Colombia.
- Serna, M.E. y Serna, A.A. (2013). Challenges and opportunities of research in formal methods. Memorias XII Conferencia Iberoamericana en Sistemas, Cibernética e Informática (pp. 1-6). Orlando, USA.
- Serna, M.E. y Serna, A.A. (2013a). Formal Specification - Present and Future. Proceedings Informática 2013 (pp. 1-13). La Habana, Cuba.
- Serna, M.E. y Serna, A.A. (2014). Methodology for perform reliable literature reviews. Revista Información, cultura y sociedad. In press.
- Serna, M.E. y Serna, A.A. (2014a). Formal specification in context: Current and future. Ingeniare - Revista Chilena de Ingeniería 22(2), pp. 243-256.
- Serna, M.E. y Serna, A.A. (2014b). Formal methods in context. Memorias XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática (pp. 61-66). Orlando, USA.
- Serna, M.E. y Serna, A.A. (2014c). Perspective and application of the formal methods. Memorias XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática (pp. 67-72). Orlando, USA.
- Shirali, S. y Shirali, M. (2010). Using formal methods in component based software development. In Sobh, T. (Ed.), Innovations and advances in computer sciences and engineering (pp. 429-432). London, Springer.
- Singh, A. y Yadav, D. (2015). Formal specification and verification of total order broadcast through destination agreement using e vent-B. International Journal of Computer Science & Information Technology 7(5), pp. 85–95.
- Singh, M. (2010). Formal Specification of Common Criteria Based Access Control Policy Model International Journal of Network Security 11(3), pp. 139-148.
- Tamrakar, S. y Sharma, A. (2015). Comparative study and performance evaluation of formal specification language based on Z, B and VDM tools. International Journal of Scientific & Engineering Research 6(9), pp. 1540–1543.
- van der Poll, J. (2010). Formal methods in software development : A road less travelled. South African Computer Journal 45, pp. 40-52.
- Vilkomir, S., Bowen, J. y Ghose, A. (2006). Formalization and assessment of regulatory requirements for safety-critical software. Innovations in Systems and Software Engineering 2(3), pp. 165–178.
- Walter, S. (2015). Towards formalized model-based requirements for a seamless design approach in safety-critical systems development. Proceedings 18th International Symposium on Real-Time Distributed Computing Workshops (pp. 111–115). Auckland, USA.
- Webel, C. & Gotzhein, R. (2007). Formalization of network quality-of-service requirements. Lectures Notes on Computer Science 4574, pp. 309-324.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Wolff, S. (2012). Scrum goes formal : Agile methods for safety-critical systems. Proceedings First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches (pp. 23–29). Zurich, Switzerland.

Wu, T., Dong, Y. y Hu, N. (2013). Formal specification and transformation method of system requirements from B Method to AADL Model. Proceedings 17th International Conference on Computational Science and Engineering (pp. 1621-1628). Chengdu, China.

Yan, F. (2011). Studying formal methods applications in CBTC. Proceedings International Conference on Management and Service Science (pp. 1-3). Wuhan, China.

You, J., Li, J. y Xia, S. (2012). A survey on formal methods using in software development. Proceedings International Conference on Information Science and Control Engineering (pp. 1–4). Shenzhen, China.

Yu, L. et al. (2008). Completeness and consistency analysis on requirements of distributed event-driven systems. Proceedings 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering (pp. 241–244). Washington, USA.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APENDICE (Artículo científico)

Proceso y progreso de la formalización de requisitos

Process and progress of formalizing requirements Edgar

Serna M., Carmen Bermúdez B.

Resumen

Desde mediados del siglo pasado se inició la investigación en métodos formales y se presentaron propuestas y metodologías para aplicarlos en el desarrollo de software. La idea era superar la diagnosticada crisis del software mediante la matematización del ciclo de vida del desarrollo de este producto. En este artículo se presentan los resultados de una revisión de la literatura en los últimos cinco años, acerca del progreso y desarrollo de la formalización de requisitos. La conclusión es que ambos aspectos son lentos: no hay interés suficiente en la industria, la academia no capacita en métodos formales, no hay patrocinio para este campo de investigación, los profesionales le temen a las matemáticas y las metodologías tradicionales de la Ingeniería de Software siguen siendo las más utilizadas en los equipos de desarrollo. Debido a la deficiencia en la calidad, la seguridad y la fiabilidad del software, es necesario reactivar la investigación y experimentación con los métodos formales, porque la esperanza es que las matemáticas serán la herramienta con la que se supere la crisis promulgada en los años 60.

Palabras clave: Métodos formales, Ingeniería de Requisitos, Calidad del software, lenguaje matemático.

Abstract

Since the middle of last century was initiated the investigation in formal methods and were presented proposal and methodologies to apply it in software development. The idea was to overcome the diagnosed crisis of software through mathematization of software lifecycle development of this product. In this article is presented the results of a literature review in the last five years, about the progress and development of formalizing requirements. The conclusions is that both aspects are slow: there is not enough interest in the industry, the academy does not trained in formal methods, there is not sponsorship for this research field, professionals are afraid of mathematics and traditional methodologies of software engineering they continue to be the most used in the development teams. Due to the deficiency in the quality, safety and reliability of the software, it is necessary to reactivate the investigation and experimentation with formal methods, because hope is that mathematics will be the tool with which the crisis passed in the 60s is exceeded.

Keywords: Formal methods, Requirements Engineering, software quality, mathematical language.

INTRODUCCIÓN

Las empresas se enfrentan a un entorno cambiante en el que las demandas y soluciones para los nuevos productos tienen que ser analizadas y evaluadas cuidadosamente. Con el fin de mantener la competitividad, deben elegir una combinación adecuada de funciones y tecnologías, también deben ser capaces de poner en el mercado dichas soluciones de la forma más económica posible. Con ciclos de vida más cortos, se ven obligadas a llevar los productos aún más rápido a la madurez del mercado (Chalupnik, Wynn y Clarkson, 2009). Además, se debe tener especial cuidado en la

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

planificación de tiempos para el desarrollo de los nuevos productos, dichos tiempos deben estar sincronizados con el fin de hacer posible la alineación de las soluciones, con las necesidades de los usuarios.

Cuando se diseña un sistema de desarrollo de un software, se recomienda que los objetivos, las funcionalidades y las restricciones se identifiquen con la mayor precisión posible, esto es lo que se constituye como la especificación de requisitos. Según la investigación realizada se encontró que muchos autores como Azeem, M. et al. (2014), de Sousa, T. et al. (2010), Ambrosio, M. y Andrade, G. (2011), Pandey, S. y Batra, M. (2013), Hassan, R. et al. (2010), Parnas, D. (2010), coinciden en que algo que se repite cotidianamente en el proceso de diseño de un desarrollo de software es que los requisitos suelen ser escritos en lenguaje natural, ya sea porque los ingenieros no conocen los lenguajes formales o porque consideran que es demasiado pronto en el ciclo de vida como para utilizar una especificación formal (es decir, aducen no tener suficientes datos para escribir con éxito un requisito formal). Yan (2011) afirma que los métodos formales son técnicas basadas en las matemáticas, a menudo con el apoyo de herramientas de razonamiento, que pueden ofrecer una manera rigurosa y eficaz para modelar, diseñar y analizar sistemas informáticos.

El principal problema con la escritura de requisitos en lenguaje natural, es que no pueden servir como insumos para las técnicas de verificación y validación automatizadas. De hecho, los lenguajes formales son obligatorios si en el plan de pruebas se pretenden utilizar técnicas automatizadas de verificación, porque son los únicos lenguajes que pueden entender los computadores. Por otra parte, no son ambiguos, es decir, que una sentencia no puede ser entendida de diferentes maneras. Esto implica que los seres humanos también se pueden beneficiar del uso de estos lenguajes, porque entenderán plenamente los requisitos sin una interpretación previa del usuario. El principal problema es que ambas actividades, la lectura y la escritura, no son fáciles de lograr cuando se trata de lenguajes formales.

De acuerdo con la práctica industrial tradicional, los requisitos se especifican en lenguaje natural y la comprobación de errores de forma manual. Las deficiencias de esta cadena de herramientas son bien conocidos: ambigüedad en la descripción de los requisitos, incrementos en costos y tiempos de entrega, y procesos de reingeniería en el diseño y la arquitectura (Dahlstedt y Persson, 2003). Además, las revisiones pueden detectar errores, pero no garantizar su ausencia. Una manera de evitar estos inconvenientes es a través de la formalización de requisitos que según Diallo, R. (2013) es el proceso de describir un sistema y sus propiedades deseadas utilizando un lenguaje con una sintaxis y la semántica definida matemáticamente. Diversas investigaciones se han llevado a cabo con este objetivo, especialmente en soporte desde los lenguajes formales y las herramientas de automatización (Heimdahl y Leveson, 1995; Heitmeyer, Jeffords, y Labaw, 1996; Post, Hoenicke, y Podelski, 2011, 2011a, 2011b; Yu et al., 2008). El inconveniente es que poder determinar si estos resultados son factibles en la práctica, no se puede lograr sin un decidido argumento de principios que se apliquen uniformemente en proyectos de la vida real.

El proceso de formalización consiste en escribir requisitos utilizando una notación formal matemática. Lo que puede ser una tarea difícil si se hace directamente desde un requisito informal y no puede haber una verdadera guía para la formalización: solamente los conocimientos y las experiencias, y tal vez algunas heurísticas (Jureta, Faulkner y Schobbens, 2008), pueden ayudar a saber cómo conseguir una escritura formal.

Los estudios acerca de la formalización de requisitos realizados por Boehm (1984, 1987) y otros han demostrado que el impacto potencial de una formulación deficiente de requisitos es sustancial. Boehm sugiere que los errores en la especificación y el análisis de requisitos son los

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

más numerosos en un sistema, y que la mayoría de ellos no se encuentran en la etapa de desarrollo, sino en las pruebas y la entrega. El costo resultante para corregir estos errores tiene una relación directa con el tiempo invertido en su búsqueda, se puede evidenciar este argumento en la investigación realizada por Kaur, A., Gulati, S. y Singh, S. (2012) quien afirma que el costo de la renovación de fallos es mucho mayor en fase posterior que en la fase anterior. De ahí la importancia de trabajar en la formalización de los requisitos, porque de esta manera se podrían entregar requisitos no-ambiguos y posiblemente con pocos errores. En este artículo se presenta un análisis al progreso y proceso de la formalización de requisitos. El objetivo es trazar un mapa de qué se ha hecho, qué se ha logrado y cuál es el futuro de la formalización como herramienta en la Ingeniería de Requisitos.

La seguridad y calidad en los sistemas software son motivos para afirmar que en nuestra época es pertinente aplicar la formalización de requisitos para dar solución a los problemas de la ingeniería de requisitos, Como mala estructuración del desarrollo, sistemas no fiables, sobrecostos por corrección de problemas en otras etapas del desarrollo o al final, entre otros. La ingeniería de requisitos según Roberto Cavada y sus colegas (2010) es una de las fases más importantes del proceso de desarrollo de software, por esta razón es muy importante establecer modelos de trabajo que garanticen calidad de los sistemas de software.

Justificación

El lenguaje natural ha demostrado desde hace mucho tiempo que no es el más indicado para gestionar y administrar la Ingeniería de Requisitos. Los computadores son máquinas que utilizan y procesan información desde un lenguaje matemático, por eso se les dificulta y los hace lentos el trabajo de compilación y de estructuración de respuestas en el desarrollo de software. Ellos necesitan traducir el lenguaje natural humano al lenguaje máquina con el que operan, y esto hace que no respondan a las velocidades que los usuarios quisieran. Si la investigación en la academia se acercara un poco más a la comprensión de los métodos formales, como base de la matematización de la Ingeniería de Software, sería posible que los estudiantes se capacitaran de mejor forma en los procesos de formalización del desarrollo de sistemas. De esta forma estarían aprovechando de mejor manera el hardware y aportarían al logro de la calidad del software. Este proyecto busca encontrar el panorama de la formalización de los requisitos, como una primera fase para que la academia se acerque a la tecnología de los métodos formales.

Planteamiento del problema

La calidad, fiabilidad y seguridad de los productos software son características que la Ingeniería de Software no satisface completamente. Desde que en los años 60 se promulgó la llamada crisis del software, los investigadores y la industria han intentado encontrar una solución para la misma, pero hasta el momento no se ha logrado. Los métodos formales surgieron como una iniciativa para alcanzarla, pero debido a sus exigencias en conocimiento matemático y a que los planes de estudios todavía no los incorporan ampliamente, por lo momento se limitan al desarrollo de software crítico. Se necesita con premura que la formalización empiece a permear los procesos formativos en la academia y que los estudiantes conozcan sus ventajas y potencialidades, lo mismo que sus desventajas y problemas. En este proyecto se pretende aportar a esta problemática y se espera involucrar los resultados en los planes de estudios relacionados con el ciclo de vida del software.

Objetivos

General: Determinar el proceso y el progreso que ha alcanzado la formalización de los requisitos.

Específicos:

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Encontrar y analizar en la literatura acerca los trabajos y propuestas publicados en relación con la formalización de los requisitos.

Recoger y valorar los aportes, el proceso y el progreso de los trabajos en esta temática.

Presentar los resultados en un documento de análisis.

Organización del trabajo

Este trabajo de investigación está compuesto por los siguientes capítulos: introducción, marco teórico, metodología, resultados y discusión

En el capítulo 1 introducción se da a conocer los diversos temas que componen el proyecto de investigación como son: situación actual en las empresas dedicadas al desarrollo de software, especificación de requisitos, métodos formales y también incluye las razones por las que se debe hacer uso de la formalización de requisitos.

El capítulo 2 marco teórico que habla de los antecedentes de la formalización y la descripción de términos planteados por varios autores como: formalización, ingeniería de requisitos, requisitos, especificación formal y métodos formales, todos estos términos permitieron abordar el problema de investigación fueron cruciales en la realización del actual proyecto de investigación.

El capítulo 3 es la metodología del proyecto de investigación donde se describe cada una de las etapas de la investigación y contiene el cronograma de actividades. Finalmente, en el capítulo 4 resultados y discusión y conclusiones, se da respuesta a las preguntas de investigación y se muestra el análisis de los resultados.

TRABAJOS RELACIONADOS

Lars Mathiasen y Andreas Munk (1985), afirman que las formalizaciones se relacionan tanto con los tipos de expresiones como con los de comportamiento. En su trabajo discuten los límites para aplicar la formalización en ambos sentidos y los ilustran con ejemplos prácticos de desarrollo de sistemas. Además, establecen que las formalizaciones son valiosas en algunas situaciones, pero insuficientes en otras. La alternativa es usarlas de manera acrítica y analizar cada situación particular, y de ahí elegir una combinación entre un enfoque formal y uno informal. No hacen un análisis al desarrollo de la formalización hasta ese momento, ni tampoco proyectan su uso futuro.

Fantechi y sus colegas (1994) presentan un asistente prototipo para la formalización de requisitos en el diseño de sistemas reactivos. Es una herramienta para traducir automáticamente frases en lenguaje natural a fórmulas de lógica temporal. De acuerdo con ellos, las frases en lenguaje natural se utilizan para expresar requisitos de manera informal, y que la lógica temporal es adecuada para expresar las propiedades de los sistemas, especificadas en términos de álgebra de procesos. La cuestión con este trabajo es que, si su objetivo era acercar la formalización a un público más amplio, logra exactamente lo contrario debido al alto contenido de lógica aplicada. No presentan un análisis previo desde el cual poder apalancar su propuesta.

Para Vilkomir, Bowen y Ghose (2006), las tareas importantes en la Ingeniería de Requisitos se orientan a resolver inconsistencias entre los reguladores y los desarrolladores de sistemas informáticos críticos. En su trabajo, proponen un nuevo enfoque para el proceso de regulación, incluyendo el uso de los requisitos reglamentarios formales como base para el desarrollo de métodos de evaluación de software. Se concentran en las diferencias entre las regulaciones y sugieren un enfoque intermedio. Argumentan que el paquete normativo debe incluir no solamente los requisitos reglamentarios, sino también los métodos para su evaluación. Este enfoque lo ilustran con ejemplos de requisitos para la protección de los sistemas informáticos de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

control a través de accesos no autorizados y contra fallos comunes del software, usando la notación Z para la formalización. En su propuesta no tienen en cuenta el trabajo que otros han adelantado en este sentido y tampoco definen qué hacer en los sistemas no-críticos. Por eso es difícil poder determinar el desarrollo de la formalización de requisitos desde los aportes de su artículo.

Morimoto et al. (2008) proponen un método híbrido para verificar formalmente si la especificación de requisitos de un sistema satisface los criterios de evaluación de la seguridad definidos en la norma ISO/IEC 15408. Clasifican los requisitos funcionales en estáticos y dinámicos, luego formalizan los primeros con la notación Z y los segundos con lógica temporal. Como resultado, los desarrolladores pueden utilizar fácilmente el método para verificar si la especificación de un sistema satisface los requisitos funcionales de seguridad. De acuerdo con los autores, este método es una evolución y mejora los propuestos anteriormente, donde solamente se aplica la notación Z. Aunque es una propuesta válida para formalizar requisitos funcionales, no se encuentran referencias para un proceso similar con los no-funcionales. Además, no proponen ni analizan el estado de desarrollo de la formalización de requisitos, desde el cual pudieron extraer y estructurar su modelo.

Por su parte, Ram Chatterjee y Kalpana Johari (2010), describen un enfoque simplificado y corroborativo para la formalización de requisitos. Su propuesta implica casos de uso, escenarios y diagramas de transición de estados como base para la automatización del proceso de formalización, que logran a través de una herramienta de desarrollo propio, la cual ejemplifica el concepto subyacente e ilustra la facilidad de automatización. Previo a su propuesta, hacen un análisis de las teorías, principios y aportes alrededor del desarrollo de la formalización de requisitos. El problema es que su estudio y propuesta se orientan al paradigma de la programación orientada por objetos, la cual, aunque de uso muy generalizado, todavía es un modelo estático. Esto contradice la necesidad del dinamismo de las soluciones a los complejos problemas actuales.

Para Serna (2012), la aplicación de los métodos formales en la industria ha progresado ampliamente en la última década y los resultados son prometedores. Describe ocho experiencias de casos reales, de las cuales cinco aplican la especificación formal de requisitos. A pesar de estos logros y de que se han documentado en numerosos estudios, para este autor todavía es común el escepticismo acerca de su utilidad y aplicabilidad. Con los resultados obtenidos en estas experiencias encuentra un perfil de las necesidades de la industria, y propone que debe ser la comunidad de los métodos formales la encargada de darles solución. Concluye que los métodos formales son utilizados de forma generalizada y rodean todas las Ciencias Computacionales, pero todavía no tienen la acogida necesaria en los planes de estudios.

Peres, Yang y Ghazel (2012) abordan una cuestión importante relacionada con la Ingeniería de Requisitos: cómo guiar y ayudar la formalización de requisitos. Con el fin de apoyar el proceso de formalización, proponen una metodología basada en una estructura formal, como la piedra angular del proceso de refinamiento. Para estos autores, los requisitos del sistema suelen ser escritos en un lenguaje natural, debido a que, por lo general, es del que tienen un mayor entendimiento los diferentes grupos de interés. Sin embargo, el uso de este lenguaje potencialmente da lugar a problemas de interpretación, que deben ser resueltos antes de usar técnicas de verificación automatizadas. Presentan la revisión a una serie de trabajos en los que

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

propone formalizar directamente los requisitos, pero que se dirigen esencialmente a la Ingeniería del Software, pero no describen cómo hacerlo en la Ingeniería de Requisitos.

RESULTADOS

Este artículo es el resultado de una revisión de literatura con la que se pretende encontrar respuesta a: 1) ¿Qué se entiende por métodos formales? 2) ¿Qué se publica en la literatura en relación con la formalización de requisitos? 3) ¿Qué tanto se ha progresado en la formalización de requisitos? Luego de aplicar la metodología descrita por estos autores, se conformó una muestra final de 40 trabajos relacionados, cuyos resultados sirvieron de base para encontrar las respuestas que se detallan a continuación.

Los métodos formales

Yan (2011) afirma que los métodos formales son técnicas basadas en las matemáticas, a menudo con el apoyo de herramientas de razonamiento, que pueden ofrecer una manera rigurosa y eficaz para modelar, diseñar y analizar sistemas informáticos. Adicionalmente, Seceleanu (2011), Ambrosio, y Andrade (2011), Kaur, Gulati y Singh (2012) y Barbosa y Lumpe (2014) coinciden en que los métodos formales proporcionan una estructura matemática dentro de la cual, de manera sistemática, se utiliza para especificar, desarrollar y verificar un sistema. Para Kaur, Gulati y Singh (2012) permiten desarrollar un sistema informático que es más completo, coherente e inequívoco, y, por otro lado, para González y Cabot (2014) evitan la introducción de imprecisiones o ambigüedades en el proceso.

Hay que tener en cuenta que la noción de *formal* en este contexto debe entenderse en términos generales como *lógica/matemática* (Gruner, 2011). Adicionalmente, Barlas, Koletsos y Stefaneas (2012) manifiestan que estos principios matemáticos no necesitan conocimientos de lenguajes y que pueden ser fácilmente entendidas en un contexto internacional. Al exponer algunos puntos de vista sobre los métodos formales, You, Li y Xia (2012) reflexionan que desarrollar software correcto y confiable ha sido un problema continuo que debe resolverse con urgencia, y que una manera eficaz de lograrlo es la teoría formal. Porque ofrece una posibilidad para crear software con pocos o ningún defecto (Jeffery et al., 2015). Esto hace que un proceso así, sea cualitativamente diferente a la Ingeniería del Software convencional, porque son fiables, robustos e ideales para utilizar en el diseño de sistemas de alta integridad que involucren cuestiones de seguridad, donde el costo de los errores puede ser muy altos (Barlas, Koletsos y Stefaneas, 2012).

El uso de métodos formales permite plantear de manera clara la especificación de un sistema, generando modelos que definen el comportamiento en términos de *qué debe hacer* y no de *cómo lo hace* (Ambrosio y Andrade, 2011). Gracias al correcto proceso de especificación, se puede verificar propiedades derivadas de cada módulo mediante técnicas de razonamiento asociadas a los modelos formales, tales como probadores de teoremas y verificadores de modelos. Al recopilar opiniones de diferentes autores sobre métodos formales, se encontró que: generalmente son costosos de aplicar (Shirali y Shirali, (2010), especialmente para la formación de los ingenieros; son demasiado lentos en su aplicación (Gruner, 2011); incrementan la calidad y la fiabilidad de un diseño y muestran su relación con los problemas prácticos y su potencial para el futuro, por otro lado, su aplicación exige que sea fácil de usar y que se tenga herramientas de verificación eficientes (Seceleanu, 2011). Por otro lado, los métodos formales pueden ayudar a resolver algunos de los problemas que enfrenta la industria del software actualmente, tales como

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

la insatisfacción de los clientes por incapacidad para cumplir con los requisitos y altos costos de apoyo (Shirali y Shirali, 2010). El objetivo principal de la Ingeniería del Software es permitirles a los desarrolladores crear sistemas que operen de forma fiable (Kaur, Gulati y Singh, 2012), y los métodos formales están destinados a proporcionar los medios para lograr una mayor precisión en el pensamiento y la documentación de esta ingeniería.

Las principales ventajas de los métodos formales son: las especificaciones formales son correctas y coherentes y pueden ser probadas y verificadas; la ambigüedad en la especificación se evita automáticamente (Kaur, Gulati y Singh, 2012); pueden garantizar la seguridad y la confiabilidad de los sistemas software (You, Li y Xia, 2012); se reduce en gran medida el error en el trabajo (You, Li y Xia, 2012); y se eliminan los errores de diseño o inconsistencias antes de su implementación. Esto es esencial para el desarrollo de sistemas fiables, especialmente en aplicaciones de software crítico (Lockhart, Purdy y Wilsey, 2014), los cuales presentan menos errores durante la verificación y a menudo requieren tiempos más cortos de desarrollo y consumen menos recursos económicos (Mandrioli y Milano, 2015). Estas ventajas se logran por el uso de las matemáticas, porque son un medio para desarrollar un modelo exacto que puede describir el objeto y el funcionamiento de forma breve y precisa. Además, porque pueden representar unidades abstractas y la especificación del sistema es coherente con las necesidades de clientes y usuarios. Por otro lado, las matemáticas ofrecen una manera de mostrar la contradicción y la imperfección en la especificación, a la vez que la coherencia entre el diseño y especificación (You, Li y Xia, 2012).

En el otro extremo, también se encuentran limitaciones a los métodos formales. De acuerdo con Kaur, Gulati y Singh (2012), son difíciles de aprender y usar y, debido a la falta de modelos de desarrollo y herramientas, los problemas complejos son difíciles de manejar. Para Mandrioli y Milano (2015) nos encontramos con una actitud general *asocial* contra el razonamiento riguroso de los métodos formales. Lockhart, Purdy y Wilsey (2014) concluyen que, desafortunadamente, actualmente los métodos formales son complejos y costosos para ser utilizados ampliamente en los sistemas software, porque el refinamiento de una especificación formal es un proceso complejo, lento y no está bien apoyado por herramientas. Otra cuestión importante, pero negativa en el caso de los métodos formales, es que la mayoría de programadores los perciben como una teoría inútil que no tiene ninguna relación con lo que hacen (Mandrioli y Milano, 2015). De acuerdo con estos investigadores, no hay manera más rápida de perder la atención en una audiencia programadores que a mostrarles una fórmula matemática. Por desgracia, uno de los problemas centrales es que no ha habido avances comparables en métodos formales y, aunque surgen nuevos lenguajes y nuevas lógicas, los errores en el diseño del programa, que provienen desde 1967, todavía se pueden encontrar en el software de hoy (Parnas, 2010). Serna (2010) coincide en afirmar que, a pesar de sus ventajas significativas, los métodos formales no son ampliamente utilizados en el desarrollo de software industrial, y la academia todavía no demuestra amplio interés para incluirlos en los planes de estudios.

La formalización de requisitos

Los requisitos son considerados como la piedra fundamental sobre la que se pueden desarrollar las soluciones software, y actualmente existe una creciente demanda de enfoques más rigurosos y sistemáticos para formalizarlos (Pandey y Batra, 2013). Serna y Serna (2014) argumentan que los métodos formales se utilizan hoy para modelar complejos sistemas críticos de seguridad, pero poco

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

se trabaja en la formalización de los requisitos desde las primeras etapas de la Ingeniería de Requisitos.

La representación más utilizada para formalizar los requisitos son los lenguajes de especificación formal (Pandey y Batra, 2013). Con este tipo de especificación, los desarrolladores comprenden mejor el sistema y eluden las ambigüedades, los flujos, las omisiones y las inconsistencias de los requisitos. Además, la especificación es un importante mecanismo de comunicación entre los clientes y los diseñadores, entre los diseñadores y ejecutores, y entre los ejecutores y los probadores (Krad, 2011). Pero, las especificaciones formales no son documentos que se escriben una vez y generalmente no se logran al principio del proceso de desarrollo de software. Se necesita tiempo para crear una primera versión de utilidad que, luego de mucho esfuerzo y revisiones, permita desarrollar una especificación cercana a las necesidades que los clientes tienen en mente (Yan, 2011), y Bollin y Rauner (2014) sostienen que un buen nivel de comprensión es una cuestión clave como atributo de calidad. Para Wolff (2012), con el uso de un lenguaje de especificación formal, el sistema puede ser descrito con precisión en cuanto a funcionalidad, concurrencia, integridad y exactitud, Esto significa que las propiedades de un sistema se pueden analizar sin tener que ejecutarlo realmente. Al utilizar la especificación formal, las propiedades del sistema se describen utilizando un lenguaje con una sintaxis y una semántica definidas matemáticamente (Vilkomir, Bowen y Ghose, 2006). Algunos lenguajes formales, tales como Z, B y VDM, se centran en especificar el comportamiento de los sistemas secuenciales, donde los estados se describen en estructuras matemáticas, tales como conjuntos, relaciones y funciones (You, Li y Xia, 2012); mientras que métodos como CSP, CCS, Statecharts, Lógica Temporal, y Autómatas, se centran en la especificación de los comportamientos del sistema en términos de secuencias, árboles u órdenes parciales de eventos (Schrap y Peters, 2014). El uso de lenguajes formales como Z y B puede mejorar la confianza del usuario en el sistema y sus impactos en el uso del mismo (Azeem, Ahsan y Noreen, 2014). Tamrakar y Sharma (2015) afirman que Z, B y VDM son lenguajes de especificación formal utilizados para especificar las necesidades del usuario en un lenguaje matemático, cuyo producto puede ser probado y verificado automáticamente.

Z trabaja en alta abstracción a nivel del sistema y proporciona una sólida base para el diseño del mismo. Mediante el uso de este lenguaje, no solamente se descubren más errores en la especificación sino también en las fases de pruebas y mantenimiento. Esto es conveniente porque con la ingeniería tradicional, el costo de corregir errores en las fases posteriores es mucho mayor que en las primeras (Azeem, Ahsan y Noreen, 2014); además, Z es una manera de descomponer una especificación en pequeñas partes llamadas esquemas. Por su parte, B es uno de los métodos formales más conocidos. Se basa en la lógica de primer orden, la teoría de conjuntos, la aritmética de enteros y las sustituciones generalizadas; se utiliza para el diseño de software desde los requisitos funcionales y permite producir casos de prueba que demuestran la exactitud y la consistencia del modelo software aplicado. El objetivo de B es obtener un producto probado y fiable (de Sousa et al., 2010). Mientras que VDM se utiliza para demostrar la equivalencia de los conceptos del lenguaje de programación (Tamrakar y Sharma, 2015).

De acuerdo con Pandey y Batra (2013), en la actual era digital, las empresas se enfrentan al desafío de liberar proyectos software de calidad, a tiempo y dentro del presupuesto, pero la realidad es que se entrega con errores, la falta de funcionalidad y a veces con sobrecostos. El costo extra se genera debido a errores en la especificación de requisitos, que pueden costar mucho tiempo y

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

dinero corregirlos cuando se detectan en las fases tardías del ciclo de vida. Por lo tanto, en la especificación de requisitos se debe seleccionar una metodología formal para abordar la fiabilidad durante la Ingeniería de Requisitos y el diseño, porque los métodos formales permiten desarrollar los errores antes de la liberación del producto (Lockhart, Purdy y Wilsey, 2014).

Los errores de especificación pueden ser reducidos drásticamente mediante el uso de métodos formales y, en consecuencia, el ingeniero de software puede crear una especificación más completa, coherente e inequívoca que con los métodos convencionales (Kaur, Gulati y Singh, 2012; Azeem, Ahsan y Noreen, 2014). Por su parte, Bollin y Rauner (2014) manifiestan que una buena especificación formal es sintáctica y semánticamente correcta, que permite un mapeo sin pérdidas entre todos los conceptos de la especificación y el modelo mental del sistema especificado; también agrega que debe ser completa, coherente y adecuada, y tener en cuenta que la facilidad de comprensión es un requisito esencial para decidir sobre su corrección semántica.

No existe una herramienta informática que pueda garantizar la corrección completa de un modelo de computador (Tamrakar y Sharma, 2015). Por lo tanto, aunque la especificación esté escrita utilizando cualquiera de los lenguajes formales, siempre contiene un alto potencial de errores. Significa que el arte de escribir una especificación formal no asegura que el sistema desarrollado sea coherente, correcto y completo. Por otro lado, si la especificación se comprueba y se analiza con el apoyo de herramientas automatizadas, aumenta la confianza sobre el sistema con la posible identificación de los errores potenciales, si existen, en la sintaxis y la semántica de la descripción formal. Una de las ideas más difundidas sobre el uso de métodos formales es su casi nula aplicación en la industria software. Si bien su adopción ha sido lenta, existen casos de importantes empresas que han tenido éxito al aplicarlos en proyectos reales (Ambrosio y Andrade, 2011).

En resumen, para conocer qué se publica en la literatura en relación con la formalización de requisitos, en la Tabla 1 se presenta el resumen de los resultados de esta investigación. La línea de tiempo de observación es a partir de 2010. Aunque es conocido que la mayor parte del trabajo en esta área se presentó en la segunda mitad del siglo pasado, el objetivo de esta investigación era verificar su progreso actual. Los trabajos de la muestra final se clasificaron de acuerdo con el enfoque primario presentado. Se aclara que muchos autores presentan resultados en más de una tipología, es decir, su investigación puede ser teórica y al mismo tiempo experimental.

- ❑ *Investigación teórica*: artículos que contienen definiciones o descripciones acerca de la formalización de requisitos.
- ❑ *Investigación experimental*: demuestra resultados a partir de experimentos en laboratorio.
- ❑ *Aplicación práctica*: describen y aplican métodos, técnicas o procedimientos de formalización de requisitos en casos de estudio.
- ❑ *Aplicación industrial*: en los que la formalización de requisitos se aplica en casos reales.

Tabla 1. Qué se publica en la literatura acerca de la formalización de requisitos

Medio		Teórica	Experimental	Práctica	Industrial
Revista	66%	63%	34%	20%	23%
Conferencia	34%				

Progreso del trabajo en la formalización de requisitos

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El progreso de la formalización de requisitos será definido como acelerado o lento dependiendo del porcentaje de aplicaciones en la industria.

En la tabla 2 se clasifican los resultados de la investigación y se muestran su participación porcentual.

Tabla 2. Progreso de la formalización de requisitos

Teórica	Experimental	Práctica	Industrial
63%	34%	20%	23%

En la Tabla 3 se presenta un resumen general de los artículos consultados, donde se puede visualizar el desarrollo de las publicaciones de la formalización de requisitos en el último quinquenio.

Tabla 3. Desarrollo de la formalización de requisitos

Autor	Desarrollo
Bhavsar (2010)	Presenta un uso de los métodos formales en la formalización de requisitos centrados en el usuario. De acuerdo con el autor, los resultados son prometedores en cuanto al mejoramiento de la calidad del producto software.
de Sousa et al. (2010)	Analiza el método B para la especificación de requisitos y describe los casos de uso y las propiedades de seguridad. Aunque su aplicación es sencilla, es un aporte hacia la masificación de los métodos formales en la especificación de requisitos.
Honghao (2010)	Propone una metodología para formalizar la Ingeniería de Requisitos y el Diseño, orientada a analizar y gestionar requisitos. Esta investigación brinda luces para demostrar la utilidad de los métodos formales en la Ingeniería de Requisitos.
Hassan et al. (2010)	Plantea el análisis y el diseño formal de la seguridad de los requisitos utilizando especificaciones en B. Los resultados demuestran efectividad y capacidad en la producción de software seguro.
van der Poll (2010)	Describe los métodos formales más comunes y la importancia de aplicar la especificación formal en los primeros pasos del ciclo de vida. Enfrenta las posiciones de los críticos y defensores de los métodos formales y presenta ejemplos de historias de éxito en la industria.
Singh (2010)	Utiliza la notación Z para formalizar los requisitos, buscando protección y seguridad contra amenazas internas, en un caso de estudio de éxito.
Cimatti et al. (2010)	Desarrolla una metodología y técnicas para formalizar y validar requisitos en aplicaciones críticas de seguridad. Aplican el resultado en un caso de estudio con resultados mejorados.
Serna (2010)	Describe la aplicación de la especificación formal de requisitos y el mejoramiento que se logra en la calidad del producto final. Propone la incorporación de los métodos formales en los contenidos curriculares de los programas en Ciencias Computacionales.
Barringer et al. (2011)	Presenta un método práctico para usar especificaciones formales en pruebas de registro en Mars Science Laboratory. Demuestra las ventajas de utilizar un lenguaje de especificación formal.
Bishop, Hay y Nance (2011)	Integra los conceptos de los métodos formales desde el inicio de la programación y, mediante una didáctica simple, muestra cómo hacer una especificación formal y qué modelos se pueden utilizar.
Fernández et al. (2011)	Es una revisión a la aplicación de los métodos formales en proyectos reales en la industria. Presenta los diversos casos de éxito y describe la situación actual en México en cuanto a la aplicación de métodos formales.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Serna (2011)	Hace un recorrido histórico por los métodos formales y describe sus aplicaciones y beneficios en la formalización de requisitos. Propone que la investigación en esta área se debe orientar a aplicaciones prácticas y a buscar reducir los costos de su utilización.
Ibrahim, Alagar y Mohammad (2011)	Aplica una base formal para la especificación y verificación de requisitos, presenta una definición formal y una teoría de composición formal para la Ingeniería de Requisitos. Aplica la verificación formal y describe los resultados.
Yan (2011)	Describe la aplicación de la especificación formal en Communications-Based Train Control (CBTC), obteniendo una descripción profunda y exacta del sistema, y lenguaje Z en el desarrollo. Concluye que al usar métodos formales se gana confianza para asegurar el sistema.
Ammar y Abdallah (2011)	Presenta un nuevo enfoque para la especificación y verificación formal basado en la reescritura la lógica. Luego lo aplica en caso de estudio con resultados prometedores en mejoramiento de la calidad.
Kaur, Gulati y Singh (2012)	Estudio comparativo entre los métodos de especificación formal y descripción de las razones de por qué usarlos en los procesos industriales.
You, Li y Xia (2012)	Muestra los logros y problemas de los métodos formales y valida su aplicación para garantizar la seguridad y estabilidad del sistema.
Barlas, Koletsos y Stefaneas (2012)	Presenta una comparación entre la especificación de requisitos tradicional y la formal. Con la segunda logra mayor comprensión, elimina ambigüedades y obliga a una mejor precisión de la especificación. Muestra aplicaciones y beneficios de su uso en la industria.
Wolff, S. (2012)	Utiliza la experiencia en la industria para combinar el desarrollo ágil Scrum con los métodos formales, pero desde una conceptualización teórica. Presenta una evaluación y discusión acerca de los beneficios de utilizarlos para formalizar los requisitos.
Bollin (2013)	Desde un marco teórico concluye que el proceso de la formalización de los requisitos es un problema de adaptación transcultural, analiza los pros y los contras del mismo y da a conocer un modelo refinado para un proceso de desarrollo de software formal.
Serna y Serna (2013)	Investigación teórica en la que se concluye se concluye que el método formal ha alcanzado significativos avances, lo que vislumbra aplicaciones ambiciosas en el futuro para apoyar rigurosos y coherentes planes de estudios en Ciencias Computacionales.
Pandey y Batra (2013)	Discute los diversos lenguajes de especificación formal y analiza sus fortalezas y debilidades. Realiza recomendaciones para el uso de métodos formales, especialmente en la Ingeniería de Requisitos.
Atlee et al. (2013)	Mediante una serie de recomendaciones para formalizar los requisitos, aporta al logro práctico de la selección de una técnica específica. Su objetivo es mejorar la aceptación de los métodos formales en el modelado de requisitos.
Serna y Serna (2013a)	La especificación formal todavía tiene usos limitados, pero la comunidad tiene una comprensión diferente acerca de su utilidad y necesidad. Hasta el momento, su desarrollo se focaliza principalmente en evaluar las herramientas relacionadas.
Chen (2013)	Analiza el dominio de los métodos formales para identificar defectos y disminuir fracasos. Aplica la formalización en la especificación, el modelado y la verificación.
Chan, Hexel y Wen (2013)	Propone una metodología de modelado de requisitos para convertir los informales en formales. Con ellos busca eliminar ambigüedades y defectos, refinar y perfeccionar el sistema y servir como un medio de comunicación entre las partes interesadas.
Diallo (2013)	Hace un estudio de los lenguajes de especificación formal, describe los modelos que se pueden aplicar y realiza un análisis general. Realiza una experimentación con estadística de depuración en un estudio de caso.
Noaman, Alsmadi y Jaradat (2013)	Aplica la especificación formal por medio de Z para potencializar la seguridad del sistema y reducir las amenazas. Los resultados son prometedores para ampliaciones y elaboraciones futuras.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Lockhart, Purdy y Wilsey (2013)	Demuestra la utilidad de los métodos formales para la especificación de requisitos en sistemas de seguridad críticos. La especificación formal reduce la ambigüedad del diseño, ayuda a probar la consistencia, e incrementa la confianza y el rendimiento.
Wu, Dong y Hu (2013)	Propone reglas de transformación desde el modelo B y describe los requisitos con máquinas abstractas. Para el autor, la especificación formal es un proceso intermedio entre la especificación de requisitos y la escritura del código.
Serna y Serna (2014a)	Realiza una revisión de la literatura, hace un recorrido por la esencia, la función, el uso y los inconvenientes de las técnicas de especificación formal y analiza criterios de valoración y evaluación a sus debilidades. Busca que la especificación formal se afiance como actividad básica de investigación y formación en la academia.
Bollin y Rauner (2014)	Muestra una serie de recomendaciones para realizar especificaciones formales, e identifica que la facilidad de lectura de una especificación es un factor clave para mejorar la calidad del software.
Schraps y Peters (2014)	Aplica gramática formal con anotaciones semánticas para formalizar requisitos, de tal forma que puedan ser analizados y procesados por el computador. Los requisitos formulados pueden ser analizados y procesados en las primeras etapas de desarrollo.
Serna y Serna (2014b)	Afirma que muchos de los problemas desafiantes en la construcción de sistemas requieren el apoyo formal para su modelado y análisis. En los procesos de investigación y aplicación de las Ciencias Computacionales existe un número creciente de aplicaciones. Pero todavía no constituyen una parte integral de los procesos formativos en pregrado y posgrado.
Azeem et al. (2014)	Utiliza especificación formal Z para mejorar la calidad y confianza del sistema. Muestra trabajos relacionados y aplica una propuesta en un estudio de caso en salud.
Li et al. (2014)	Aplica object-Z para describir sistemas complejos. Demuestra su utilidad en un caso de estudio de un sistema de suministro de gasolina. Los resultados obtenidos alientan el uso de la especificación formal.
Serna y Serna (2014c)	Describe aspectos relevantes de los métodos formales y realiza un marco del futuro de la investigación en el área. Los resultados invitan a la reflexión de este componente de las Ciencias Computacionales y de la necesidad de tener una comunidad más amplia y sólida dedicada a fomentar y aplicar la formalización de los requisitos.
Jeffery (2015)	Realiza un estudio empírico para encontrar la productividad de los proyectos que utilizan métodos formales, específicamente al formalizar los requisitos. Identifica una serie de preguntas sobre los resultados en productividad de estos proyectos.
Tamrakar y Sharma (2015)	Compara tres métodos para especificar formalmente: Z, B y VDM. Usar especificaciones formales no asegura un sistema completamente correcto, pero se aumenta la confianza en el mismo.
Singh y Yadav (2015)	Utiliza el método Event-B para escribir especificaciones formales, con el fin de garantizar la comprensión de los límites en los que un algoritmo puede ser usado. Para el autor, la especificación, la validación y la verificación formales con la clave para obtener un mejor diseño.
Walter (2015)	Realiza una representación de formalización de requisitos por medio de diagramas SysML usando la semántica RSL. Formaliza únicamente los requisitos no-funcionales y ese diseño lo aplica a un caso de estudio en el área automotriz.

ANÁLISIS DE RESULTADOS

De acuerdo con la revisión de la literatura en esta investigación, el trabajo en formalización de requisitos ha tenido un progreso lento, en la tabla 2 se puede evidenciar que la mayoría de los trabajos tenidos en cuenta para la presente investigación se enfocan más en definiciones teóricas que son representadas por el 63% y solo el 23% de estos en aplicación en la industria. Algunos autores han analizado este hecho y presentan sus conclusiones. Entre las causas se destacan: falta de formación en la academia (Serna, 2011), falta de aceptación en la industria (Serna, 2012), no hay

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

demostraciones realmente importantes de sus ventajas, los costos, falta de personal capacitado, los ingenieros les tienen fobia a las matemáticas (Atlee, J. et al. 2013), la industria no quiere experimentar con principios que no tienen comprobación suficiente y en la última década la investigación en métodos formales ha perdido interés y patrocinio (Serna y Serna, 2014).

Los investigadores de mediados del siglo pasado estaban convencidos que la única manera de mejorar la calidad de los productos software era a través de la matematización de la Ingeniería del Software (van der Poll, 2010), pero a medida que sus resultados comenzaron a publicarse, la industria comenzó a encontrar y colocar barreras al progreso de este trabajo. Solamente el software crítico los arropó como tabla de salvación para solucionar sus problemas de fiabilidad y seguridad, y son diversos los casos de éxito en estos desarrollos. El problema surge cuando las empresas de desarrollo de software comercial se enfrentan a situaciones de incumplimiento, que incrementan los costos del proceso y que las obligan a dedicar menos tiempo a la estructura misma de las actividades. La decisión es disminuir la mayor cantidad de tiempo posible y, como la forma tradicional de realizar la verificación y validación es como una barrera de contención al final del ciclo de vida, esta fase es la que se recorta para encontrar algo de tiempo (You, Li y Xia, 2012). El estudio sobre la formalización de requisitos se ha relegado a unos pocos investigadores que convencer a sus estudiantes de posgrado para que trabajen con ellos. Con el inconveniente analizado antes de que los profesionales le temen a todo lo que tenga que ver con matemáticas. La recomendación es a que las universidades incluyan cursos relacionados en los planes de estudios de los pregrados en Ciencias Computacionales, de tal forma que los estudiantes se interesen y profundicen en su asimilación y comprensión. De acuerdo con los autores analizados en esta investigación, la formalización de requisitos se investiga de forma mayoritariamente teórica, y de esta manera no se logrará convencer a la industria de su aplicación. Puede ser que la etapa de la teorización ya haya sido superada en los muchos trabajos que se presentaron el siglo pasado. La necesidad actual es masificar los métodos formales y encontrar la forma de aplicarlos en la matematización de la Ingeniería de Software, pero con costos que estén al alcance de las empresas desarrolladoras de software. De esta manera será posible reavivar el interés en esta área de trabajo y de encontrar el patrocinio necesario para ejecutar la investigación necesaria. Si el objetivo es lograr por fin superar la crisis del software, los métodos formales deben ser el centro de desarrollo. Porque sus bondades ya han sido suficientemente demostradas y porque el lenguaje con el que funciona el computador es el matemático.

CONCLUSIONES

Los resultados demuestran que el progreso y desarrollo de la formalización de requisitos son lentos, y que en los últimos años el interés por darle continuidad ha disminuido. Aunque se acepta que los métodos formales son una herramienta útil y necesaria para superar la crisis de la calidad del software, muy pocos investigadores quieren abordar la formalización de requisitos.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Dentro de las investigaciones obtenidas se puede evidenciar la importancia de la aplicación de los métodos formales durante el desarrollo de un producto de software, ya que evita ambigüedades y permite obtener productos de software más precisos y seguros, pero cabe resaltar que en los últimos 5 años la aplicación de la formalización de requisitos en la industria ha sido muy poca por factores como: desconocimiento de la formalización de requisitos por parte de los desarrolladores, la búsqueda de desarrollos de software ágiles o hechos en el menor tiempo posible y la falta de publicaciones por parte de las empresas que aplican la formalización de requisitos en sus desarrollos.

La industria necesita demostraciones de los beneficios que representa la formalización, pero hasta ahora no tiene inclinación directa por apoyar las iniciativas que en este sentido proponen la comunidad. Solamente en el desarrollo de sistemas críticos se aprecia una amplia participación de las empresas de desarrollo y la industria, porque el objetivo es que funcionen sin poner en riesgo la vida humana o las inversiones económicas. Se necesita mayor trabajo para reducir los costos actuales de la formalización, pero, sobre todo, es urgente contar con el personal capacitado para aplicarla y darle continuidad.

Esta es una Sociedad software-dependiente, porque muy pocas de sus actividades están por fuera del ámbito de este desarrollo tecnológico. Pero los productos que se liberan o entregan a los usuarios todavía no satisfacen aspectos como fiabilidad y seguridad. Las matemáticas ofrecen la posibilidad de superar este problema y su representación en los métodos formales es una alternativa prometedora.

REFERENCIAS

- Ambrosio, M. y Andrade, G. (2011). Métodos formales aplicados en la industria del software. *Temas de Ciencia y Tecnología* 15(43), pp. 3-12.
- Ammar, B, y Abdallah, K. (2011). Towards the formal specification and verification of multi-agent based systems. *International Journal of Computer Science Issues* 8(4), pp. 200–210.
- Atlee, J. et al. (2013). Recommendations for improving the usability of formal methods for product lines *Proceedings 1st FME Workshop on Formal Methods in Software Engineering* (pp. 43–49). San Francisco, USA.
- Azeem, M. et al. (2014). Specification of e-Health system using Z: A motivation to formal methods. *Proceedings International Conference for Convergence of Technology* (pp. 1–6). Pune, India.
- Azeem, M., Ahsan, M. y Noreen, K. (2014). Specification of e-Health System using Z: A motivation to formal methods. *Proceedings International Conference for Convergence of Technology* (pp. 1–6). Pune, India.
- Barbosa, L y Lumpe, M. (2014). Formal aspects of component software. *Lecture Notes in Computer Science* 7684, pp. 253–254.
- Barlas, K., Koletsos, G. y Stefaneas, P. (2012). Extending standards with formal methods: Open document architecture. *Proceedings International Symposium on Innovations in Intelligent Systems and Applications* (pp. 1-5). Trabzon, Turkish.
- Barringer, H. et al. (2011). An entry point for formal methods : Specification and analysis of event logs. *Proceedings Workshop on Formal Methods for Aerospace* (pp. 16–21). New Mexico, USA.
- Bhavsar, M. (2010). Analysis of Multiagent Based Interactive Grid Using Formal Methods - A Reliable Approach. *Proceedings 3rd International Conference on Emerging Trends in Engineering and Technology* (pp. 462-466). Goa, India.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Bishop, M., Hay, B. y Nance, K. (2011). Applying formal methods informally. Proceedings of the Annual Hawaii International Conference on System Sciences. Proceedings Annual Hawaii International Conference on System Sciences (pp. 1-8). Kauai, USA.
- Boehm, B. (1984). Model and metrics for software management and engineering. IEEE, USA.
- Boehm, B. (1987). Industrial software metrics top 10 list. IEEE Software 4(5), pp. 84–85.
- Bollin, A. (2013). Do you speak Z? Formal methods under the perspective of a cross-cultural adaptation problem. Proceedings 1st FME Workshop on Formal Methods in Software Engineering (pp. 8-14). San Francisco, USA.
- Bollin, A. y Rauner, D. (2014). Formal specification comprehension the art of reading and writing Z. Proceedings 2nd FME Workshop on Formal Methods in Software Engineering (pp. 3–9). Hyderabad, India.
- Chalupnik, M., Wynn, D. y Clarkson, J. (2009). Approaches to mitigate the impact of uncertainty in development processes. Proceedings 17th International Conference on Engineering Design (pp. 1-12). Stanford, USA.
- Chan, L., Hexel, R. y Wen, L. (2013). Rule-based behaviour engineering: Integrated, intuitive formal rule modelling. Proceedings 22nd Australian Software Engineering Conference (pp. 20-29). Melbourne, Australia.
- Chatterjee, R. y Johari, K. (2010). A simplified and corroborative approach towards formalization of requirements. Communications in Computer and Information Science 94, pp. 486-496.
- Chen, X. (2013). Research on the implementation of internal control in enterprise information system by domain analysis and formal methods - A case study of sales activities internal control under chinese enterprise environment. Proceedings Fourth World Congress on Software Engineering (pp. 171-175). Hong Kong, China.
- Cimatti, A. et al. (2010). Formalization and Validation of Safety-Critical Requirements. Proceedings Workshop on Formal Methods for Aerospace (pp. 68–75). Eindhoven, Netherlands.
- Dahlstedt, A. y Persson, A. (2003). Requirements interdependencies - Moulding the state of research into a research agenda. Proceedings Ninth International Workshop on Requirements Engineering (pp. 71–80). Klagenfurt/Velden, Austria.
- de Sousa, T. et al. (2010). Automatic analysis of requirements consistency with the B method. Software Engineering Notes 35(2), pp. 1-4.
- Diallo, R. (2013). The need for usable formal methods in verification and validation. Proceedings Winter Simulation Conference (pp. 1257-1268). Washington, USA.
- Fantechi, A. (1994). Assisting requirement formalization by means of natural language translation. Formal Methods in System Design 4(3), pp. 243-263.
- Fernández, C. et al. (2011). Métodos formales aplicados en la industria del software. Temas de Ciencia y Tecnología 15(43), pp. 3-12.
- González, C. y Cabot, J. (2014). Formal verification of static software models in MDE: A systematic review. Information and Software Technology 56(8), pp. 821–838.
- Gruner, S. (2011). Editorial: Special section on formal plus agile methods. ACM Software Engineering Notes 36(4), pp. 26.
- Hassan, R. et al. (2010). Formal analysis and design for engineering security automated derivation of formal software security specifications from goal-oriented security requirements. IET Software 4(2), pp. 149–160.
- Heimdahl, M. y Leveson, N. (1995). Completeness and consistency analysis of state-based requirements. IEEE Transactions on Software Engineering 22(6), pp. 3–14.
- Heitmeyer, C., Jeffords, R. y Labaw, B. (1996). Automated consistency checking of requirements specifications. ACM Transactions Software Engineering and Methodology 5(3), pp. 231–261.
- Honghao, G. (2010). Based on formal methods in trustable software requirements engineering. Proceedings International Conference on Internet Technology and Applications (pp. 1-4). Wuhan, China.
- Ibrahim, N., Alagar, V. y Mohammad, M. (2011). Specification and Verification of Context-dependent.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Proceedings of the 20th international conference on World Wide Web (pp. 17-33). Hyderabad, India.
- Jeffery, R. et al. (2015). An empirical research agenda for understanding formal methods productivity. *Information and Software Technology* 60, pp. 102–112.
- Jureta, I., Faulkner, S. y Schobbens, P. (2008). Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering* 13, pp. 87–115.
- Kaur, A., Gulati, S. y Singh, S. (2012). A comparative study of two formal specification languages: Z-notation & B-method. *Proceedings Second International Conference on Computational Science, Engineering and Information Technology* (pp. 524-531). Coimbatore, India.
- Krad, H. (2011). Formal methods and automation for system verification. *Proceedings 4th International Conference on Modeling, Simulation and Applied Optimization* (pp. 1–12). Kuala Lumpur, Malaysia.
- Li, Y et al. (2014). Specifying complex systems in object-Z: A case study of petrol supply systems. *Journal of Software* 9(7), pp. 1707–1717.
- Lockhart, J., Purdy, C. y Wilsey, P. (2013). Formal methods for safety critical system specification. *Proceedings 57th International Midwest Symposium on Circuits and Systems* (pp. 201-204). College Station, USA
- Mandrioli, D. y Milano, P. (2015). On the heroism of really pursuing formal methods - Title inspired by Dijkstra's "On the Cruelty of really Teaching Computing Science". *Proceedings of the Third FME Workshop on Formal Methods in Software Engineering* (pp. 1–5). Austin, USA.
- Mathiassen, L. y Munk, A. (1985). Formalization in systems development. *Lecture Notes in Computer Science* 186, pp. 101-116.
- Morimoto, et al. (2008). Classification, formalization and verification of security functional requirements. *Lecture Notes in Computer Science* 4910, pp. 622-633.
- Noaman, M., Alsmadi, I. y Jaradat, A. (2013). The specifications of E-Commerce secure system using Z language. *The Research Bulletin of Jordan ACM II(III)*, pp. 127-131.
- Pandey, S. y Batra, M. (2013). Formal methods in requirements phase of SDLC. *International Journal of Computer Applications* 70(13), pp. 7–14.
- Parnas, D. (2010). Really rethinking "formal methods". *Computer* 43 (1), pp. 28–34.
- Peres, F, Yang, J. y Ghazel, M. (2012). A formal framework for the formalization of informal requirements. *The International Journal of Soft Computing and Software Engineering* 2 (8), pp. 14-27.
- Post, A., Hoenicke, J. y Podelski, A. (2011). Vacuous of real-time requirements. *Proceedings 19th IEEE International Requirements Engineering Conference* (pp. 153–162). Trento, Italy.
- Post, A., Hoenicke, J. y Podelski, A. (2011a). Rt-inconsistency: A new property for real-time requirements. *Lecture Notes in Computer Science* 6603, pp. 34–49.
- Post, A., Menzel, I. y Podelski, A. (2011b). Applying restricted English grammar on automotive requirements - Does it work? A case study. *Lecture Notes in Computer Science* 6606, pp. 166–180.
- Schraps, M. y Peters, M. (2014). Semantic annotation of a formal grammar by Semantic Patterns. *Proceedings 4th International Workshop on Requirements Patterns* (pp. 9–16). Karlskrona, Sweden.
- Seceleanu, C. (2011). Formal methods applied in industry : Success stories, limitations, perspectives – panel introduction. *Proceedings IEEE 35th Annual Computer Software and Applications Conference* (pp. 448-449). Munich, Germany.
- Serna, A.A. (2012). Formal Methods in industry. *Los Métodos Formales en la Industria. Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)* 2(2), pp. 44-51.
- Serna, M.E. & Serna, A.A. (2014). Formal specification in context: Current and future. *Ingeniare - Revista Chilena de Ingeniería* 22(2), pp. 243-256.
- Serna, M.E. (2010). Formal methods and Software Engineering. *Revista Virtual Universidad Católica del Norte* 30, pp. 158-184.
- Serna, M.E. (2011). Formal Methods: Perspective and future application. *Memorias III Jornadas de Investigación de la Facultad de Ingenierías* (pp. 64-68). Medellín, Colombia.
- Serna, M.E. y Serna, A.A. (2013). Challenges and opportunities of research in formal methods. *Memorias XII Conferencia Iberoamericana en Sistemas, Cibernética e Informática* (pp. 1-6). Orlando, USA.
- Serna, M.E. y Serna, A.A. (2013a). Formal Specification - Present and Future. *Proceedings Informática 2013* (pp. 1-13). La Habana, Cuba.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Serna, M.E. y Serna, A.A. (2014). Methodology for perform reliable literature reviews. Revista Información, cultura y sociedad. In press.
- Serna, M.E. y Serna, A.A. (2014a). Formal specification in context: Current and future. *Ingeniare - Revista Chilena de Ingeniería* 22(2), pp. 243-256.
- Serna, M.E. y Serna, A.A. (2014b). Formal methods in context. *Memorias XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática* (pp. 61-66). Orlando, USA.
- Serna, M.E. y Serna, A.A. (2014c). Perspective and application of the formal methods. *Memorias XIII Conferencia Iberoamericana en Sistemas, Cibernética e Informática* (pp. 67-72). Orlando, USA.
- Shirali, S. y Shirali, M. (2010). Using formal methods in component based software development. In Sobh, T. (Ed.), *Innovations and advances in computer sciences and engineering* (pp. 429-432). London, Springer.
- Singh, A. y Yadav, D. (2015). Formal specification and verification of total order broadcast through destination agreement using e vent-B. *International Journal of Computer Science & Information Technology* 7(5), pp. 85–95.
- Singh, M. (2010). Formal Specification of Common Criteria Based Access Control Policy Model *International Journal of Network Security* 11(3), pp. 139-148.
- Tamrakar, S. y Sharma, A. (2015). Comparative study and performance evaluation of formal specification language based on Z, B and VDM tools. *International Journal of Scientific & Engineering Research* 6(9), pp. 1540–1543.
- van der Poll, J. (2010). Formal methods in software development : A road less travelled. *South African Computer Journal* 45, pp. 40-52.
- Vilkomir, S., Bowen, J. y Ghose, A. (2006). Formalization and assessment of regulatory requirements for safety-critical software. *Innovations in Systems and Software Engineering* 2(3), pp. 165–178.
- Walter, S. (2015). Towards formalized model-based requirements for a seamless design approach in safety-critical systems development. *Proceedings 18th International Symposium on Real-Time Distributed Computing Workshops* (pp. 111–115). Auckland, USA.
- Wolff, S. (2012). Scrum goes formal : Agile methods for safety-critical systems. *Proceedings First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches* (pp. 23–29). Zurich, Switzerland.
- Wu, T., Dong, Y. y Hu, N. (2013). Formal specification and transformation method of system requirements from B Method to AADL Model. *Proceedings 17th International Conference on Computational Science and Engineering* (pp. 1621-1628). Chengdu, China.
- Yan, F. (2011). Studying formal methods applications in CBTC. *Proceedings International Conference on Management and Service Science* (pp. 1-3). Wuhan, China.
- You, J., Li, J. y Xia, S. (2012). A survey on formal methods using in software development. *Proceedings International Conference on Information Science and Control Engineering* (pp. 1–4). Shenzhen, China.
- Yu, L. et al. (2008). Completeness and consistency analysis on requirements of distributed event-driven systems. *Proceedings 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering* (pp. 241–244). Washington, USA.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Carmen Eduvina Bermúdez
Estudiante

Prof. Edgar Serna M.
Asesor

FECHA ENTREGA: 28 de abril de 2016

<p>_____</p> <p>Comité trabajos de grado de la Facultad</p> <p> ACEPTADO____ RECHAZADO _____ ACEPTADO CON MODIFICACIONES_____ </p> <p style="text-align: right;">ACTA NO. _____</p>

FECHA ENTREGA: _____

<p>_____</p> <p>Consejo de Facultad</p> <p style="text-align: right;">ACTA NO. _____</p>
--

FECHA ENTREGA: _____