

Capítulo 2

Diseño de un sistema de información que implementa el modelo automatizado para asegurar la calidad de los requerimientos en proyectos de software

Luis Eduardo Peláez Valencia - Luiseduardo.pelaez@gmail.com

Universidad Tecnológica de Pereira

Miguel A. Cohuo Ávila - Macohuo@itescam.edu.mx

Instituto Tecnológico Superior de Calkiní en el estado de Campeche (ITESCAM)

Alonso Toro Lazo - Alonso.toro@ucp.edu.co

Juan Luis Arias Vargas - Juan.arias@ucp.edu.co

Universidad Católica de Pereira

Daniel Eduardo Rodríguez Franco - Danieleduardorodriguez09@gmail.com

Tecnológico Nacional de México

Yulieth Gómez Osorno - Yulieth.gomez@ucp.edu.co

Universidad Católica de Pereira

Diego Alejandro Pérez Correa - Dalejandropc@gmail.com

Universidad Católica de Pereira

I. INTRODUCCIÓN

Abordar los proyectos de software para usuarios finales conlleva un ejercicio riguroso y responsable que debe asegurar el cumplimiento de las expectativas del cliente, además de las recomendaciones y especificaciones que hayan sido establecidas en las etapas iniciales de un determinado proyecto, de tal manera que la disposición y el cumplimiento de las actividades llevadas a cabo durante el proceso de desarrollo concluyan en atributos de calidad que posea el producto software resultante, de modo que pueda ser

calificado como de calidad. Esto, normalmente configurado en la triple restricción de alcance o funcionalidad, tiempo y costo [1], variables propias del proceso de aseguramiento de la calidad del software.

La ingeniería del software establece una serie de criterios indispensables para lograr procesos y productos software de calidad mediante el desarrollo de actividades de análisis y diseño previos y de integración y verificación posteriores [2], distribuyéndolas en el ciclo de vida del desarrollo del software. En las etapas iniciales del

desarrollo, el análisis y diseño permiten comprender los requerimientos del cliente y plasmarlos fielmente en especificaciones concretas [3] realizada como parte del proyecto de investigación denominado “Procedimiento para especificar y validar requisitos de software en MIPYMES desarrolladoras de software, basado en estudios previos en la región”. Lo anterior responde a la necesidad de indagar sobre los problemas de calidad del sector del software relacionado con el tratamiento que se les hace a los requisitos como un punto clave para lograr productos y procesos de calidad, con el ánimo de desarrollar propuestas que contribuyan a mejorarla, inicialmente en la industria que se representa desde la ciudad de Pereira. Para ello, se hace un recorrido por algunos trabajos que permiten dar cuenta de los logros que se han alcanzado en investigaciones similares y que tienen como objeto de estudio los requisitos, reconociendo, a partir de unos antecedentes regionales, nacionales e internacionales los orígenes de ese problema y las propuestas que se han desarrollado para ayudar a corregirlo. Finalmente, y como conclusión, se encuentra que han sido muchas las propuestas que de una u otra manera pretenden contribuir al mejoramiento de la calidad del software a partir de los requisitos, pero en ninguna de las revisadas se ha propuesto un método, proceso o lineamiento claramente definido para elaborar una completa, consistente,

modificable y trazable, específicamente de requisitos de software en las micro, pequeñas y medianas empresas (MIPYMES), con lo cual los expertos de desarrollo cumplen con las expectativas de calidad.

De acuerdo con [4], estos requerimientos “expresan las necesidades y limitaciones impuestas a un producto de software que contribuyen a la solución de algún problema en el mundo real”. En este sentido, la adopción de una disciplina como RQA (*Requirement Quality Assurance*) aporta

[un] conjunto de instrumentos y buenas prácticas que proporcionan la seguridad de que los productos y procesos relacionados con la definición de un problema que se resolverá con software nuevo o modificado, cumplirán con lo especificado previamente, para generar la confianza necesaria que los lleve a ser considerados como de calidad [5].

Lo anterior, unido a un proceso de SQA (*Software Quality Assurance*),

proporciona la seguridad de que los productos y procesos en el ciclo de vida del proyecto cumplan con sus requisitos especificados mediante la planificación, promulgando y llevando a cabo un conjunto de actividades de suficiente confianza, e indicando la calidad que radica en el software [6].

Esto permite adoptar un proceso de calidad en todas las etapas del ciclo de vida del software, que comprende desde las etapas iniciales de requerimientos hasta las pruebas finales del sistema, lo que promueve la utilización y adaptación de técnicas y buenas prácticas que ayudarán a la consecución de la calidad.

De esta manera, y para efectos del desarrollo del sistema de información que permita automatizar el “modelo para el aseguramiento de la calidad de los requerimientos en proyectos de software”, llevado a cabo por la línea Ingeniería de Software del Grupo de Investigación entre Ciencia e Ingeniería, este capítulo pretende aportar a la aplicación práctica de los conceptos RQA y SQA evidenciando algunos de los artefactos y técnicas de especificación, diseño y modelado propuestos por la ingeniería de software para el desarrollo de la aplicación denominada CHAMI, que implementa el modelo automatizado antes mencionado.

Por consiguiente, se detallan algunos procesos, técnicas y buenas prácticas de ingeniería de software implementadas, específicamente las que hacen parte de la etapa de diseño. Es importante aclarar que este artículo no se enfoca en el modelo para el aseguramiento de la calidad de los requerimientos, sino en el diseño del sistema de información que lo

implementa; más concretamente en la modelación de los procesos.

El desarrollo y la especificación de requerimientos en requisitos es quizá la etapa más importante y determinante del proceso de desarrollo de software, pues permite garantizar, desde las primeras etapas del proceso, la calidad de las aplicaciones resultantes [7] [8].

Así, dentro de la etapa de requerimientos es indispensable aplicar técnicas y procedimientos que permitan la elaboración de un documento de especificación de requisitos de software (SRS, por sus siglas en inglés [9]) completo, consistente, modificable, trazable [10] y lo suficientemente detallado como para permitir, en primera instancia, una ayuda fundamental al criterio del analista y/o profesional encargado del proceso de requerimientos en su labor.

Por ello, el presente artículo se enfoca, principalmente, en explicar y documentar la fase de diseño de software, para la cual, a partir de la especificación de requisitos de software entregada, se elaboran diferentes diagramas estáticos y dinámicos de acuerdo con el estándar UML [11], para modelar la estructura y el comportamiento del sistema que se va a desarrollar, así como algunos prototipos y otros procesos que hacen parte del ciclo de vida del desarrollo del software seleccionado para el proyecto.

El capítulo se estructura de la siguiente manera: en la sección 2 se presentan los materiales y métodos contemplados en el proyecto; en la sección 3 se describen los resultados y artefactos obtenidos de la ejecución de cada una de las fases llevadas a cabo en el desarrollo del sistema de información; la discusión de resultados se presenta en la sección 4, en la sección 5, las conclusiones y, por último, la bibliografía consultada.

II. MATERIALES Y MÉTODOS

La definición de las técnicas de diseño y modelado del sistema

de información, que implementa el modelo automatizado para el aseguramiento de la calidad de los requerimientos en proyectos de software, se justifica a partir del ejercicio y la exploración de bibliografía y levantamiento de información que sustenten la validez y pertinencia del modelo, con base en el aseguramiento de la calidad de los requerimientos, así como la inclusión y utilización de técnicas de diseño y modelado. De esta forma, se proponen las fases para la ejecución del proyecto (Figura 1), considerando que la fase 6, a pesar de haber sido parte del proyecto, no se presenta en este documento:

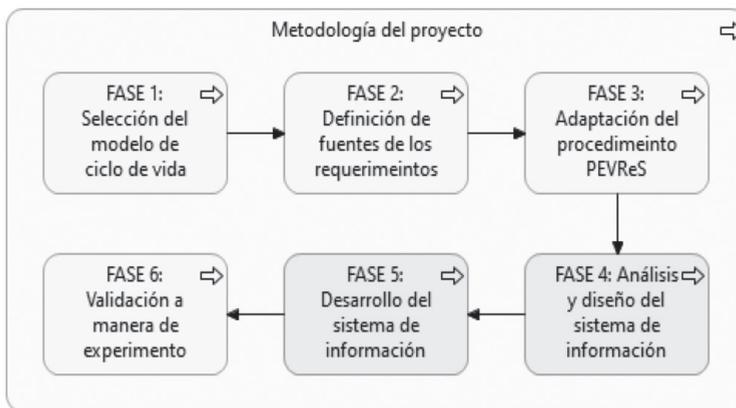


Fig. 1. Fases de la metodología de ejecución del proyecto

A. Fase 1: Selección del modelo de ciclo de vida del software (MCVS)

Dentro del ejercicio del diseño e implementación del modelo automatizado CHAMI, y con la intención de establecer un marco de trabajo que ayude a la estructura,

planificación y control del desarrollo del producto software, se hace importante seguir un modelo de ciclo de vida del desarrollo de software que aborde cada uno de los aspectos antes considerados y que, a su vez, logre sintetizar y justificar la construcción basada en el diseño.

Actualmente, existe una amplia variedad de metodologías y modelos dispuestos a abordar desarrollos de software, sin embargo, tienen limitaciones inherentes al contexto de aplicación. Es decir que no existe una metodología que pueda ser aplicada perfectamente en todos los contextos de trabajo y todos los tamaños de proyecto [12].

En este sentido, de acuerdo con el estudio y la clasificación de modelos de ciclo de vida para el desarrollo de software (MCVDS), el rumbo de la investigación se encamina hacia la utilización del modelo de prototipos, el cual, a partir de la experiencia del equipo de desarrollo y la participación del usuario en la elaboración de prototipos, su evaluación y desarrollo, debe permitir la definición de especificaciones completas y acordes con el producto definitivo, lo que en consecuencia se ajusta al propósito y los objetivos del proyecto respecto a la calidad en el proceso de desarrollo y el producto software.

El modelo de prototipos inicia con la actividad de comunicación, continúa con la realización de un plan rápido y modelado o diseño rápido, para luego construir el prototipo y

desarrollarlo [13], lo que permitirá una retroalimentación por parte del cliente/usuario que aclare las funcionalidades y requerimientos que debe poseer el sistema, lo cual tendrá sin dudas un impacto positivo al obtener mejores resultados en el producto final entregado, asegurando que cumpla con la mayor especificidad de las necesidades del cliente.

De acuerdo con la caracterización realizada, el modelo prototipo se centra en una representación de aquellos aspectos del software que serán visibles para el cliente y/o usuario final, de esta manera, el modelo permite realizar pruebas y modificaciones cuando es necesario. La clasificación y las etapas consideradas en el modelo se especifican en la Figura 2, las fases 1, 3, 4 y 5 se adaptan al esquema propuesto por el modelo para el aseguramiento de la calidad de los requerimientos en proyectos de software, en el desarrollo de este proyecto.

Sin embargo, por el enfoque de este documento, se presentarán solo algunos elementos correspondientes a las fases de modelado-diseño y construcción del prototipo, aunque todas las fases propuestas fueron llevadas a cabo.



Fig. 2. Modelo de ciclo de vida de prototipos. Adaptado de [13]

B. Fase 2: Recopilación y fuentes de los requerimientos

De acuerdo con el proceso de definición e identificación de los requerimientos como etapa inicial dentro del proceso de desarrollo de software, y con el fin de comprender y establecer las necesidades en función del sistema, es de vital importancia definir claramente los requerimientos del software, de tal manera que se tenga una especificación completa y consistente que sea, como lo indica [14], una base sólida para el proceso de desarrollo de software que, en consecuencia con el modelo desarrollado, tenga la calidad esperada para el diseño y la construcción del sistema de información CHAMI.

La ingeniería de requisitos plantea la división de la fase de requisitos en

cuatro etapas: identificación, análisis, especificación y validación de requisitos [15]. Para la identificación de requisitos del proyecto, se establecieron cuatro fuentes principales de información: primera, la recolección de datos sustentada sobre la caracterización del sector y la industria, específicamente en las Mipymes [16], [5]; segundo, identificación de fuentes documentales de trabajos previos y el estado de la técnica [10]; tercero, exploración de antecedentes relacionados con investigaciones y propuestas de mejora de procesos para la industria del software, y cuarto, el análisis y experiencia del grupo de trabajo, junto con la vinculación del sector educativo (estudiantes y graduados de las carreras relacionadas con las TIC y el software).

La Figura 3 contempla el trabajo colectivo y colaborativo dispuesto sobre las fuentes para determinar los requerimientos del sistema de información. Seguidamente, se

presenta una breve explicación de cada una de las fuentes de información y los instrumentos empleados en algunas de ellas.



Fig. 3. Fuentes de los requerimientos

1) *Caracterización de la industria:* el instrumento de medición diseñado y aplicado a las empresas Mipymes dedicadas a la producción de software en la ciudad de Pereira (Colombia), tuvo el objetivo de determinar y caracterizar su proceso de gestión en la etapa de requerimientos, y que además permitiera recolectar la información suficiente para generar preguntas a la industria del software sobre la manera como están siendo tratados los requisitos, y en consecuencia conocer el estado de la apropiación del aseguramiento de la calidad del producto software [12]. De esta manera, se tomaron como entradas las necesidades representadas por el

sector, que se convirtieron en una de las fuentes de requerimientos más determinantes para el desarrollo del sistema de información CHAMI.

2) *Estado de la técnica y caracterización de herramientas:* mediante una metodología de exploración teórica se hizo hincapié en un aspecto de gran importancia, como es la fiabilidad, con el fin de no desconocer el ejercicio realizado por autores y organizaciones que son autoridad en temas de ingeniería de software, ingeniería de requisitos y aseguramiento de la calidad del software. En tal sentido, en este estudio se tuvieron en cuenta expertos en la disciplina como Pressman

[17], Sommerville [18], Wiegers [19], entre otros, y organizaciones como International Requirements Engineering Board (IREB) [20], International Standards Organization (ISO) [21], IEEE Computer Society [4] y Software Engineering Institute (SEI).

Respecto a la caracterización de herramientas para el desarrollo y la gestión de requisitos, se hace una exploración teórica y práctica de las herramientas más usadas en el entorno mundial por las compañías desarrolladoras de software a la hora de desarrollar y gestionar los requerimientos. Esta caracterización se realiza considerando factores como tipo de licenciamiento (de pago y libres), plataforma, facilidad de soporte o contacto con el fabricante, fecha de la última actualización, vigencia en el mercado, posibilidad de prueba, etc.

Finalmente, se hace un análisis comparativo para determinar sus principales características, funcionalidades y limitaciones, además de identificar los elementos de entrada requeridos, las restricciones, los detalles del proceso que se ha de seguir, los atributos que permite definir, sus ventajas y desventajas, entre otros aspectos [12].

3) *Trabajos previos:* por medio de una exploración teórica de antecedentes, a partir de la cual las deficiencias encontradas por [10] y la caracterización de la industria reportada

en el punto número 1, respecto a un deficiente o nulo aseguramiento de la calidad de los requisitos en la industria local del software, se realiza una exploración de antecedentes regionales, nacionales e internacionales, con la intención de evidenciar proyectos de investigación, desarrollos e iniciativas propuestos para contribuir a una solución a este problema. A partir de allí, se logran adquirir algunos requerimientos y lecciones aprendidas importantes para el desarrollo del sistema de información.

4) *Consideraciones del grupo de investigación:* basados en la experiencia del equipo del proyecto de investigación y la articulación con otros actores vinculados con el proyecto por parte del grupo de Investigación Entre Ciencia e Ingeniería de la Universidad Católica de Pereira, y a partir de las investigaciones previas y desarrollos realizados, se establece un instrumento para la inclusión de requerimientos y consideraciones prácticas que deben ser contempladas por el sistema que se va a desarrollar. De esta manera, se logran requerimientos adicionales que, desde la práctica y la experiencia, aportan al mejoramiento de la propuesta de desarrollo.

C. Fase 3: Adaptación del modelo PEVReS para la especificación y validación de requisitos de software

Una vez identificados los requerimientos, se efectúan su

especificación y validación de acuerdo con las técnicas propuestas por el procedimiento para especificar y validar requisitos de software (PEVReS) [8], tal como se muestra en la Figura 4, el cual facilita la creación de un documento SRS que cumpla con las características de una buena

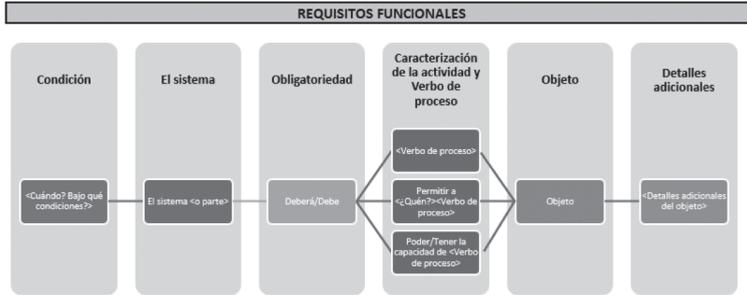
especificación [22], [10], y además se validen mediante un trabajo conjunto entre el analista y el cliente y/o usuario del sistema para asegurar que los requisitos que se identificaron y se especificaron reflejen realmente lo que los usuarios necesitan [23].



Fig. 4. Procedimiento PEVReS [24]

El procedimiento aporta además los formatos y plantillas necesarios para lograr la aplicación de cada una de las técnicas, así como el documento SRS basado en una ampliación del estándar IEEE Std. 830. En la Figura 5, por ejemplo, se representa la técnica de especificación de requisitos de usuario (DRU) en lenguaje natural sugerida por el procedimiento, mediante la cual

se establece la sintaxis adecuada de escribir los requerimientos funcionales y no funcionales que luego serán llevados al documento SRS, de tal manera que se cumplan los criterios de evaluación de un buen requisito definidos por [19]: completo, correcto, realizable, necesario, priorizable, no ambiguo y verificable.



Requisitos funcionales - Ejemplo

Escenario condicional	Obligatoriedad	Interacción	Objeto	Detalles
	El sistema debe	Permitir a los estudiantes <matricular>	Un curso	Con el cual cumplan los prerrequisitos
	El sistema debe	Permitir a los profesores <consultar>	La lista de estudiantes	Registrado en cada uno de sus cursos
Cuando se ha cumplido fecha de inscripción a un curso o cuando se ha completado el cupo máximo de estudiantes	El sistema debe	<Cerrar>	El curso	
	El sistema debe	Tener la capacidad de <notificar>	Al sistema de pagos	Cuando un estudiante haya realizado el proceso de pago de matrícula
	El sistema debe	Permitir a los administradores <crear>	Cursos	
Tan pronto como un estudiante haya realizado el pago de matrícula		<Imprimir>	La factura del curso	

Fig. 5. Esquema de especificación de requisitos funcionales en lenguaje natural [25]

De igual forma, se utilizan instrumentos para la gestión de requisitos, con la intención de determinar aspectos que ayuden con la tipificación, identificación, priorización, descripción y control

de los requerimientos, una vez se haya capturado por parte del cliente, lo que garantiza la documentación y trazabilidad. Uno de estos instrumentos, elaborado por el grupo de investigación, se presenta en la Figura 6.

Formato de levantamiento de requerimientos												
(Grupo de investigación Entre Ciencia e Ingeniería - UCP)												
Proyecto						Responsable						
Identificador	Nombre					Responsable						
Metodología a utilizar												
Empresa												
Nombre de la empresa responsable				Dirección				Teléfono				
Solicitante del requerimiento												
Nombre del solicitante				Dirección				Teléfono				
Dependencia												
Requerimiento												
Identificador	fecha de levantamiento			Versión		Medio por el cual fue recibido		Telefónica		Oral		
Tipo de requerimiento	Funcional	<input type="checkbox"/>	No funcional	<input type="checkbox"/>	Proceso	<input type="checkbox"/>	Infraestructura	<input type="checkbox"/>	Descripción:			
	Sistema	<input type="checkbox"/>	Interfaz	<input type="checkbox"/>	Usuario	<input type="checkbox"/>	Seguridad	<input type="checkbox"/>				
¿Se relaciona con otros sistemas?	Si	<input type="checkbox"/>	No	<input type="checkbox"/>	¿Cuál?							
Prioridad según el cliente	Alta	<input type="checkbox"/>	Media	<input type="checkbox"/>	Baja	<input type="checkbox"/>	Prioridad según el desarrollador		Alta	<input type="checkbox"/>	Media	<input type="checkbox"/>
Estado	Recibido	<input type="checkbox"/>	Detallado	<input type="checkbox"/>	Aprobado	<input type="checkbox"/>	Rechazado	<input type="checkbox"/>	En ejecución	<input type="checkbox"/>	Entregado	<input type="checkbox"/>
Restricciones	Descripción											
	Técnica	<input type="checkbox"/>										
	Económica	<input type="checkbox"/>										
	Ambiental	<input type="checkbox"/>										
	Equipo de trabajo	<input type="checkbox"/>										
Tipo de usuario que utiliza	legal	<input type="checkbox"/>										
Especializado	<input type="checkbox"/>	Potencial	<input type="checkbox"/>	General	<input type="checkbox"/>	Casual	<input type="checkbox"/>					
Complemento												
Complemento	Interfaz de usuario				Requerimiento de hardware				Requerimiento de software			
¿Existen riesgos? Si <input type="checkbox"/> No <input type="checkbox"/> Descripción: _____												
Plan de trabajo												
Fecha de inicio		Fecha fin		Responsable								
Recursos												
¿Se necesita personal adicional? Si <input type="checkbox"/> No <input type="checkbox"/>												
Cliente						Nombre y firmas						
Nombre			Nombre			Analista						
C.C.			C.C.									

Fig. 6. Formato para identificación de requisitos.
Fuente: Grupo de investigación Entre Ciencia e Ingeniería

D. Fase 4: Análisis y diseño del sistema de información

Se presenta en esta sección un resumen, a manera de ejemplo demostrativo, de algunos de los diagramas UML y modelos desarrollados como parte del análisis estructural y dinámico del sistema de información CHAMI. Con ellos, se pretende adquirir un mayor dominio del problema y de la solución, de tal manera que los resultados correspondan con los requerimientos y cumplan las expectativas del proyecto.

1) *Modelo de datos:* la mayoría de los sistemas de software robustos utilizan bases de datos de información de altos volúmenes de datos. En

algunos casos, la base de datos es independiente del sistema de software, en otros se crea para el sistema que se está desarrollando. Es una parte importante del modelado de sistemas para la definición de la forma lógica de los datos procesados por el sistema. Estos se denominan a menudo modelos semánticos de datos [17]. La Figura 7 muestra el esquema general del modelo de datos elaborado para una arquitectura multi-tenant, capaz de soportar múltiples equipos de trabajo, laborando en múltiples proyectos para múltiples empresas del sector, donde los diferentes usuarios no dependen de la empresa y pueden ir y venir entre diferentes empresas, ligados a ellas solamente por sus proyectos de interés.

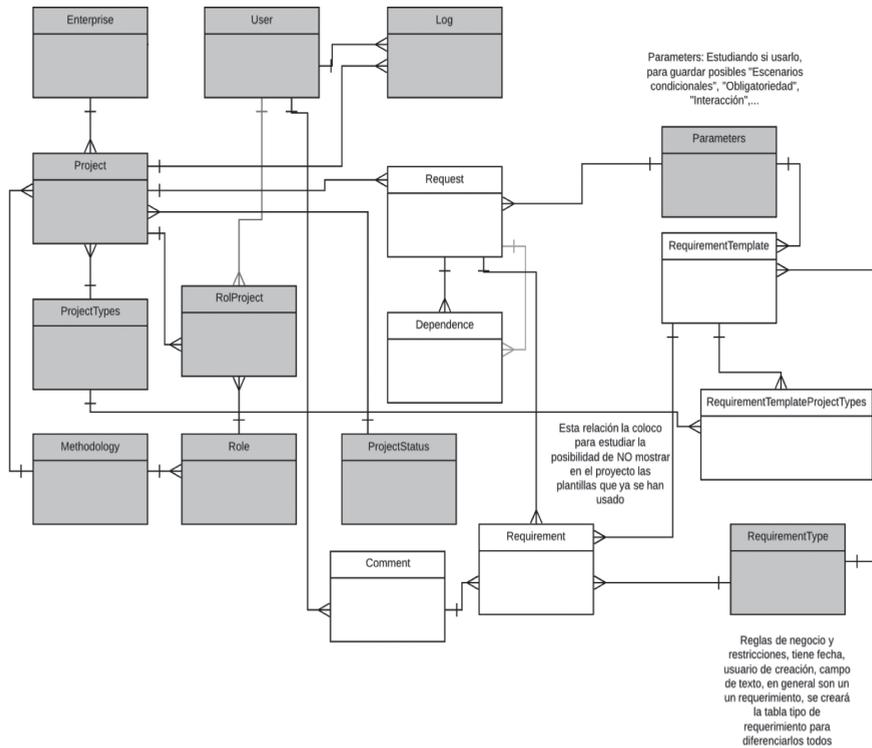


Fig. 7. Diagrama del Modelo de Datos

2) *Modelo de clases*: representa la estructura estática del sistema de información, indicando las clases que lo componen junto con sus atributos, operaciones, y las relaciones entre los objetos, tales como asociaciones, herencias, composición o agregación,

entre otras. De acuerdo con las técnicas de diseño, para este tipo de proyectos se hace indispensable considerar el diagrama de clases, por lo que en la Figura 8 se presenta el diagrama respectivo.

- *Request*: se encarga de la especificación del requisito de acuerdo con los parámetros entregados por la técnica de especificación en lenguaje natural antes presentada, y garantiza que la descripción del requerimiento siga la sintaxis adecuada y que todos los atributos relacionados con su estructura hayan sido correctamente diligenciados (escenario condicional, obligatoriedad, caracterización de la actividad y verbo de proceso, objeto, detalles adicionales).
- *Dependence*: se considera como fuente de control que justifique y sustente la dependencia entre grupos de requerimientos, que dentro de un esquema tradicional podrían agruparse por Caso de Uso o por disponibilidad del cliente y/o Usuario, ayudando a garantizar la trazabilidad de los requerimientos.

Sobre estas clases se establecen los métodos que se pueden ejecutar sobre los objetos pertenecientes a cada clase, los cuales representarán funciones de la aplicación que estarán disponibles

para el usuario mediante la aplicación.

3) *Modelo relacional*: dentro del proceso de diseño se considera indispensable la realización del modelo relacional, el cual permite diseñar la base de datos con las tablas en las cuales será almacenada la información del sistema. Dicha estructura, que puede entenderse como un conjunto de relaciones bidimensionales entre tablas constituidas por filas (tuplas) y columnas (atributos) [26], contempla principalmente las entidades **Request** y **Requirement**, en las cuales reposan los registros correspondientes a cada requerimiento y su especificación.

Asimismo, se incluyen otras tablas en función de la operatividad y parametrización del sistema, que permiten obtener un conjunto de elementos que aportan a la automatización del proceso de desarrollo y gestión de cada requisito, por supuesto al interior de un determinado proyecto que a su vez pertenece a una empresa y posee usuarios particulares. La estructura mencionada se presenta a continuación en la Figura 9.

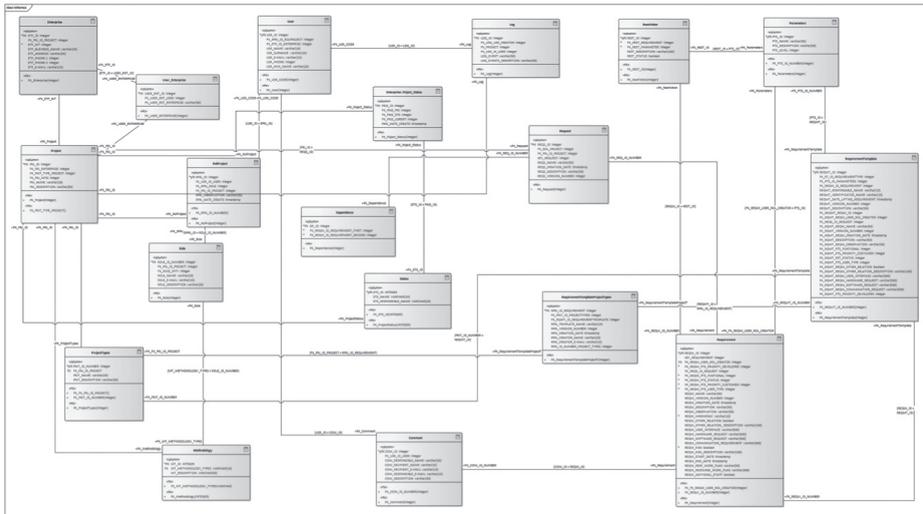


Fig. 9. Modelo Relacional

4) *Diagramas de casos de uso:* existen diferentes maneras de clasificar los requisitos, sin embargo, de acuerdo con [24], basta una clasificación en requisitos de negocio, requisitos de usuario y requisitos del sistema para describir de forma suficiente las funcionalidades del sistema que representan las necesidades del cliente. Asimismo, para cada tipo de requisito existe un conjunto de técnicas que ayudan a representarlo. El diagrama de casos de uso es adoptado

en este proyecto como la técnica para representar los requisitos funcionales que corresponden al nivel usuario y su comportamiento dinámico [27].

Este diagrama permite representar la forma en que un cliente (Actor) opera con el sistema, y además la forma, el tipo y orden en que los elementos interactúan entre sí. Un ejemplo de los diagramas realizados durante el diseño del sistema se muestra en la Figura 10.

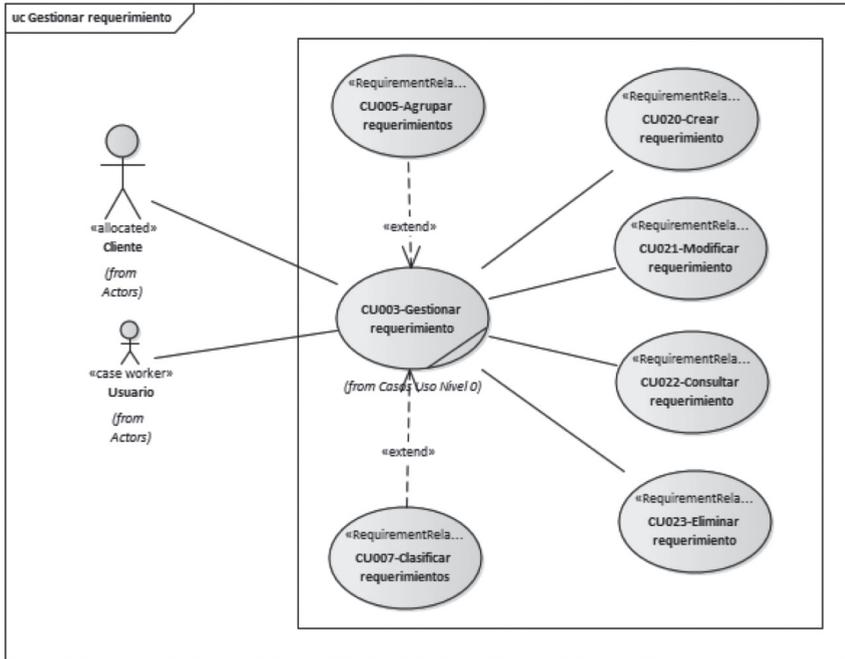


Fig. 10. Caso de uso gestionar requerimiento

5) *Diagramas de secuencia:* el diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos [27]. Estos diagramas ayudan a definir los métodos u operaciones que se deben codificar al interior de las clases, así como los mensajes de

interacción con el usuario y la secuencia lógica de las funcionalidades.

El diagrama presentado en la Figura 11 muestra una interacción del actor Mipyme con la interfaz gestión de proyectos, la cual a su vez se comunica con el controlador y la entidad responsable de los proyectos, de tal manera que pueda crear y consultar información de un determinado proyecto.

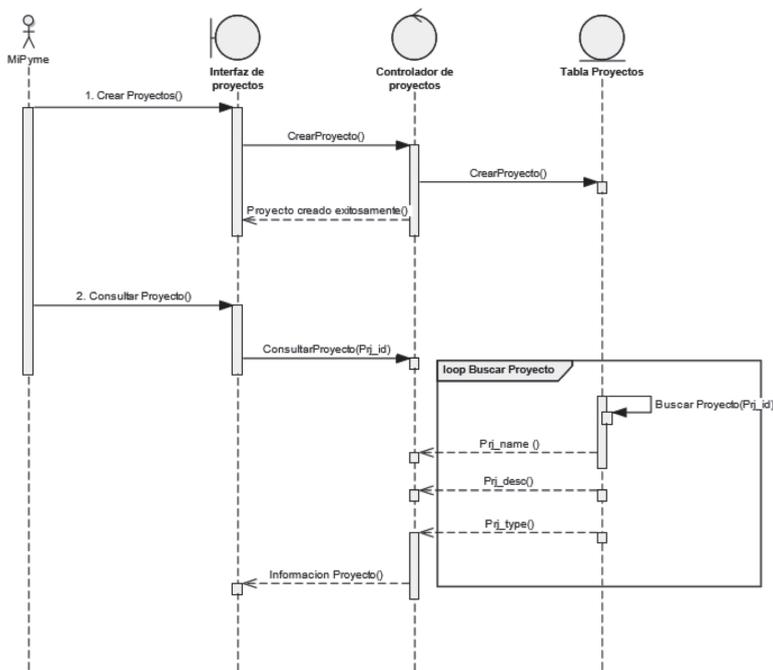


Fig. 11. Diagrama de secuencia gestión de proyecto

III. RESULTADOS

En esta sección se presentan los artefactos software desarrollados de acuerdo con la Fase 5 de la metodología propuesta (desarrollo del sistema de información), presentada en la Figura 1, así como las etapas 2 y 3 del modelo de ciclo de vida guiado por prototipos, presentado en la Figura 2. Esta labor se realizó a partir de los requerimientos especificados y corresponden a los *mockups* entregados al ingeniero desarrollador como parte del proceso de maquetación de vistas.

En esta etapa del proyecto, el documento SRS cumple un papel supremamente importante, ya que es

el insumo principal para el proceso de desarrollo, toda vez que es el documento en el cual se describen con claridad y sin ambigüedades las necesidades y expectativas del cliente, de cuya calidad depende en gran medida la calidad del producto software resultante.

En la Figura 12 se puede observar un ejemplo de requisitos de usuario escritos mediante la técnica de lenguaje natural (ver Figura 5). Posteriormente, estos requisitos se clasifican, ponderan, priorizan y especifican mejor mediante la plantilla de identificación y análisis (ver Figura 6) con el propósito de facilitar su gestión y trazabilidad a lo largo del proyecto.

REQUERIMIENTOS FUNCIONALES					
Identificador	Escenario condicional	Obligatoriedad	Interacción	Objeto	Detalles
REQF001		El sistema debe	permitir al administrador <gestionar*>	un usuario	
REQF002	Siempre y cuando no exista un proyecto con el mismo nombre	El sistema debe	permitir al administrador <gestionar>	un proyecto	
REQF003	Siempre y cuando no exista un requerimiento con el mismo nombre en el proyecto	El sistema debe	permitir solo a un usuario <gestionar>	un requerimiento	
REQF004		El sistema debe	permitir al usuario cliente <describir>	un requerimiento	en un máximo de 150 caracteres
REQF005	Al crear un requerimiento	El sistema debe	<asignar>	fecha del requerimiento	usando la fecha actual del sistema
REQF006	Al modificar un requerimiento	El sistema debe	<asignar> una fecha de última actualización	al requerimiento	de acuerdo con la fecha del sistema
REQF007		El sistema debe	permitir al usuario <asignar> prioridad	a un requerimiento	de acuerdo con las siguientes categorías: alta, media y baja
REQF008	Al crear o modificar un requerimiento	El sistema debe	<versionar>	el requerimiento	

Fig. 12. Ejemplo de requerimientos funcionales

A. Elaboración de prototipos

Una vez finalizadas las etapas de educación, especificación y documentación de requerimientos, se procedió a la maquetación del sistema mediante la elaboración de los primeros *Mockups* y la promoción de una comunicación continua con el cliente que permitiera una oportuna retroalimentación, lo cual aportó algunas mejoras al diseño del sistema y facilitó al cliente una mejor comprensión de requisitos.

Este involucramiento del cliente en el proceso de diseño (Diseño Centrado en el Usuario —DCU—) y desarrollo constituye una característica fundamental de la metodología de prototipos, no hacerlo se podría entender, según [10], como una de las principales causas de fracaso de los proyectos de software atribuibles a los requisitos. La Figura 13 muestra uno de los primeros prototipos de alta fidelidad realizados como parte del proyecto.

Modificar_Requerimiento_Analista

MODIFICAR REQUERIMIENTO

Nombre Requerimiento

Tipo Usuario

Prioridad

Medio

Solicitado por:

Descripcion
 Escribir un max. de 150 Caracteres

El contenido de este formulario ha sido modificado. ¿Desea guardar los cambios?

NOMBRE COMPLETO DEL ANALISTA
 Analista

REQUERIMIENTO
 Recibe Requerimiento
 Restaurar Requerimiento
 Añadir Requisito a Requerimiento

REQUISITOS
 Crear Requisito
 Indicadores Requisito
 Colocar dependencia Requisito
 Restaurar Requisito

Fig. 13. Mockup Modificar requerimiento analista

B. Desarrollo del sistema de información CHAMI

El sistema de información desarrollado se comporta en forma de plataforma web, permite ingresar y gestionar los requisitos de un proyecto de software y llevar a cabo sobre ellos un conjunto de actividades que permiten asegurar su calidad (ver Figura 14), esto es, cumplir los

criterios de evaluación de un requisito para que sea considerado como de calidad en el proceso de gestión del proyecto. El sistema fue desarrollado con la intención de facilitar a las Mipymes desarrolladoras de software de la región el proceso de desarrollo y gestión de los requisitos para sus proyectos.

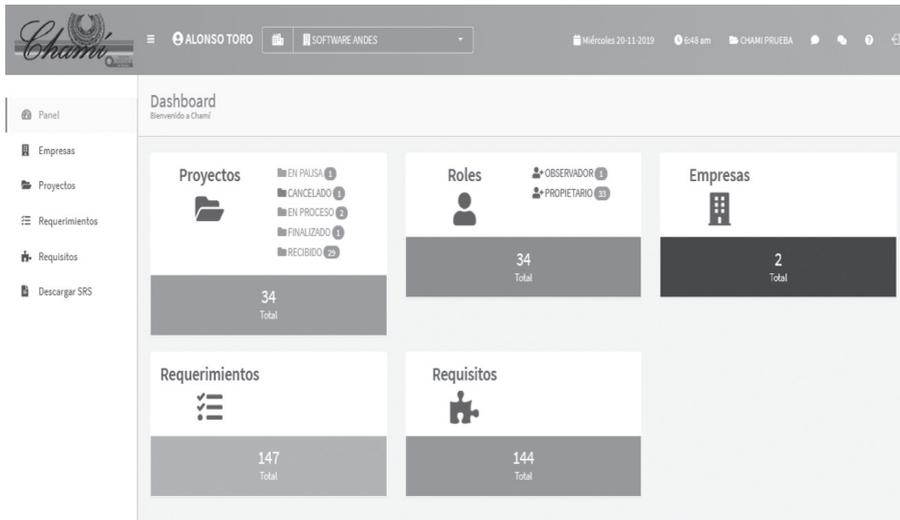


Fig. 14. Vista general de la plataforma CHAMI

El sistema está conformado por seis diferentes módulos, mediante los cuales se puede realizar toda la gestión de las empresas, los usuarios y proyectos y los requerimientos/requisitos asociados a ellos. De igual forma, el sistema permite la generación del documento SRS de un proyecto una vez finalizado el proceso de desarrollo y gestión de requisitos. A continuación, se presenta una descripción general de algunos de los módulos principales:

- 1) *Panel:* corresponde al *Dashboard* de la aplicación, donde se presentan estadísticas generales de los módulos subsecuentes.
- 2) *Empresa:* permite el ingreso al administrador de empresas, donde el usuario podrá gestionar las empresas que estarán asociadas a un determinado proyecto.
- 3) *Proyectos:* este módulo permite administrar los proyectos de una empresa.
- 4) *Requerimientos:* permite ingresar a la administración de requerimientos de un proyecto; se mostrarán únicamente los requerimientos de los proyectos de los cuales el usuario sea propietario o analista.
- 5) *Requisitos:* una vez ingresados los requerimientos a un proyecto, mediante este módulo se puede gestionar la especificación de requisitos.
- 6) *Descargar SRS:* esta funcionalidad de la aplicación permite generar y descargar en formato PDF el documento SRS del proyecto en el cual se esté trabajando.

La Figura 15 corresponde a una parte del documento SRS que permite generar la plataforma.



SRS PROYECTO PR2			
Empresa:		EMPRESA COLOMBIA	
ID de proyecto:	13	Metodología:	AGILES
Tipo:	CRM	Fecha creado:	2019/08/30 14:09 horas
Estado Actual:		RECIBIDO	
Descripción del Proyecto:			
PESQUERA			
PARTICIPANTES			
NOMBRES	ALIAS	EMAIL	ROL
USUARIO 2 COLOMBIA	usuario2.col	usuario2.col@chami.com	ANALISTA
DIEGO ALEJANDRO PEREZ CORREA	dalejandropc	dalejandropc@gmail.com	PROPIETARIO
ALONSO TORO	ALONSO TORO	alonso.toro@ucp.edu.co	PROPIETARIO
REQUERIMIENTOS			
ASIGNACIÓN DE NÚMEROS PROPIOS			
Fecha:		2019/09/14 12:26	
Prioridad cliente:	MEDIA	Tipo usuario:	ESPECIALIZADO
medio recepción	OTRO	Estado	RECIBIDO
Descripción:		Versión 0	
ESPECIFICACIÓN DEL REQUISITO			
Nombre Requisito especificado:			
ASIGNACIÓN DE NÚMEROS PROPIOS			
Fecha Creado:	2019/09/14 12:27	Prioridad Desarrollador:	BAJA
Estado:		RECIBIDO	
Descripción:			
El sistema debe añadir numero de lote propio			
Observaciones:			
SIN OBSERVACIÓN			

Fig. 15. Módulo de generación del documento SRS

Los módulos “Requerimientos” y “Requisitos” permiten, además de la gestión, la especificación de algunos procedimientos adicionales que se mencionan a continuación:

- *Exportar e imprimir:* permite exportar en formatos CSV y PDF, o imprimir el listado de requerimientos o requisitos correspondientes a un determinado proyecto.
- *Dependencias:* el sistema facilita el reporte de dependencias entre requerimientos, con la finalidad de facilitar su priorización y posterior desarrollo.
- *Buscar:* en todos los módulos se tiene la posibilidad de realizar una búsqueda por palabra clave, de tal manera que se pueda ubicar fácilmente un elemento cuando el listado es muy grande.
- *Gestión de la configuración:* permite el control de versiones de un requerimiento o requisito, de tal manera que se puedan bajar o subir sus versiones de acuerdo con las actualizaciones realizadas sobre el mismo.
- *Gestionar estados:* permite controlar los estados de avance de un determinado requerimiento o requisito. Al cambiar de un estado a otro se genera una nueva versión del elemento.

- *Añadir sistemas foráneos*: este procedimiento, que aplica solo a los requisitos, permite indicar la existencia de un sistema externo al sistema en desarrollo con el cual se requiera interactuar.
- *Añadir restricción*: permite indicar las restricciones (*constraints*) relacionadas con el desarrollo de un requisito en particular.
- *Comentarios*: permite adicionar notas al administrador del proyecto o analista asignado, sobre un determinado requerimiento, requisito o proyecto en particular. Son visibles en todo el *workspace* y facilitan la realización de ajustes o cambios sobre los elementos en desarrollo.

IV. DISCUSIÓN

Los resultados obtenidos en este proyecto permiten describir los logros alcanzados en las etapas de la metodología propuesta, en lo que tiene que ver con el diseño del sistema de información [3] realizada como parte del proyecto de investigación denominado “Procedimiento para especificar y validar requisitos de software en MIPYMES desarrolladoras de software, basado en estudios previos en la región”. Lo anterior responde a la necesidad de indagar sobre los problemas de calidad del sector del software relacionado con el tratamiento

que se les hace a los requisitos como un punto clave para lograr productos y procesos de calidad, con el ánimo de desarrollar propuestas que contribuyan a mejorarla, inicialmente en la industria que se representa desde la ciudad de Pereira. Para ello, se hace un recorrido por algunos trabajos que permiten dar cuenta de los logros que se han alcanzado en investigaciones similares y que tienen como objeto de estudio los requisitos, reconociendo, a partir de unos antecedentes regionales, nacionales e internacionales los orígenes de ese problema y las propuestas que se han desarrollado para ayudar a corregirlo. Finalmente, y como conclusión, se encuentra que han sido muchas las propuestas que de una u otra manera pretenden contribuir al mejoramiento de la calidad del software a partir de los requisitos, pero en ninguna de las revisadas se ha propuesto un método, proceso o lineamiento claramente definido para elaborar una completa, consistente, modificable y trazable, específicamente de requisitos de software en las micro, pequeñas y medianas empresas (MIPYMES). La primera de ellas tuvo como propósito implementar un modelo de ciclo de vida que guiara el desarrollo mediante el uso de prototipos, lo cual fue indispensable para tener una línea clara en la ejecución del proyecto. Si bien no se presenta en este documento el seguimiento de todas las fases, es

importante al menos indicar cuál fue el transcurso metodológico del proyecto de diseño que implementaría el modelo de aseguramiento de la calidad de los requisitos ya elaborado por el grupo de investigación referenciado [28].

Algunos pequeños productores de software llevan a cabo sus proyectos según se lo indican las herramientas que utilizan y acompañados de sus ideas de lo que es mejor para el proyecto; sin embargo, es necesario comprender los principios y el conocimiento que hay alrededor de cada fase de un proyecto de desarrollo de software para, de esta manera, reconocer y apropiarse los métodos y las metodologías existentes en cada proyecto, más allá de la herramienta [29], por lo que llegar a este nivel de detalle con los requerimientos puede entregar ideas diferentes a las Mipymes de cómo hacerlo para asegurar mayor éxito en sus procesos y productos.

De igual forma, se presentan las fases propuestas por el modelo de ciclo de vida seleccionado, pero en este documento solo se desarrollan aquellas relacionadas con el diseño del sistema.

Un segundo aspecto relevante tiene que ver con las fuentes de información empleadas para la identificación de requisitos [30], pues el proyecto no se limitó únicamente al punto de vista de los integrantes del equipo, sino que consideró el estado de la técnica,

los antecedentes del orden regional, nacional e internacional y, aún más importante, se involucraron los sectores productivo (representado por la industria del software) y académico [5], logrando una sinergia que difícilmente se logra encontrar en el desarrollo de un sistema de software.

Por otro lado, la adaptación de técnicas y buenas prácticas sugeridas por el Procedimiento para especificar y validar requisitos de software (PEVReS) constituyó un insumo importante para facilitar las actividades correspondientes en las fases de requisitos y modelado del proyecto (Fase 4 de la metodología), a partir del cual se elaboraron los modelos conceptuales (estáticos y dinámicos) necesarios para lograr un mejor entendimiento del problema y de la solución para desarrollar.

Teniendo presente que los proyectos, en este caso los de software, deben concluir en un valor de negocio para quienes los desarrollan y para quienes los utilizan [31], abordar este proyecto rigurosamente desde la modelación, también se propone dejar un valor agregado para la comunidad académica y para la industria del software.

Igualmente, se entrega información que describe aspectos del desarrollo y resultados finales del proyecto, pasando por el proceso de educación y especificación de requisitos y la elaboración de prototipos relacionados

con la maquetación de las vistas, y el sistema de información desarrollado junto con algunas de sus principales funcionalidades.

Finalmente, la aplicación web resultante exhibe características particulares que la hacen diferente de otras —cuyo enfoque también es la gestión de requisitos—, puesto que implementa un modelo de RQA producto de varios procesos de investigación, además de varias de las técnicas propuestas por el procedimiento PEVReS para guiar a las Mipymes en el proceso de definición y especificación de requisitos de usuario que cumplan con criterios de calidad, para, finalmente, generar de manera automática un documento SRS ajustado a las necesidades de cada proyecto y empresa. De igual forma, la aplicación simplifica notablemente la redacción de los requisitos al entregar al usuario elementos gramaticales preestablecidos reutilizables y reducir las posibilidades de ambigüedad debidas al lenguaje natural, a la vez que lo guía por una secuencia de fases en las que, además de lograr que el requisito sea completo, correcto y priorizable, pase por un proceso de gestión de la configuración y se establezcan las dependencias, sistemas foráneos y restricciones a las que haya lugar. Lo anterior permite, entonces, identificar estos elementos diferenciadores como rasgos que distinguen esta aplicación de las citadas y que existen para labores similares.

V. CONCLUSIONES

Se presentó en este capítulo el resumen de un conjunto de actividades de ingeniería de software relacionadas con el diseño de un sistema de información, desarrollado por el Grupo de Investigación Entre Ciencia e Ingeniería de la Universidad Católica de Pereira, que implementa un modelo *cuasi*-automatizado para el aseguramiento de la calidad de los requisitos en empresas de software, el cual pretende ser un instrumento útil para las Mipymes desarrolladoras de software a la hora de llevar a cabo los procesos de desarrollo y gestión de requisitos. El concepto de *cuasi*-automatizado hace alusión a su diferenciación de los procesos automatizados en sentido estricto, dada la necesidad de intervención humana en la incorporación de los datos de entrada y la búsqueda de las salidas.

Los resultados del proyecto permiten obtener valiosa información de una experiencia de hacerle ingeniería de software a la ingeniería de software, con el pretexto de buscar asegurar la calidad del proceso redundando en la calidad del producto.

Asimismo, la aplicación de buenas prácticas de ingeniería de software, RQA y SQA, contribuye a adoptar un enfoque sistemático y disciplinado, lo cual materializa el propósito de diferentes metodologías para la gestión

de proyectos que están en mora de aportar herramientas para proyectos muy específicos, como los de desarrollo de software. Esto contribuye a elevar el nivel de calidad del proceso desde las primeras etapas del proyecto y, en consecuencia, la calidad del producto software resultante.

En este caso, la aplicación de una propuesta de aseguramiento de calidad de requerimientos sobre una propuesta de un MCVS, donde ambos se convierten en alternativas para mejorar la calidad del proceso de desarrollo, en una simbiosis interesante de actualizar y sistematizar mejor las metodologías y las propuestas clásicas para hacer software.

Por otro lado, se logra el desarrollo de un sistema de información adecuado a las necesidades de los productores de software de la región, que tiene la capacidad de acompañar los proyectos de software en la gestión de requisitos de calidad, y que desde etapas tempranas del proyecto asegura la elaboración de especificaciones de software acordes con las características de calidad establecidas por los autores y organizaciones internacionales expertas en la disciplina.

Entre los industriales consultados, la gestión de requerimientos constituye, en algunos casos, una tarea dentro del ejercicio de análisis y diseño. Sin embargo, el trabajo elaborado permite concluir que, al

darle un papel más protagónico a la gestión de requerimientos, se logra una modelación más acertada y en correspondencia con el problema abordado, lo que, necesariamente, redundará en que las fases siguientes tengan un mayor nivel de acierto.

También, resulta muy importante involucrar al cliente/usuario en las etapas de requisitos y modelado del proyecto, ya que de esta manera se logra comprender mejor las necesidades del cliente, su retroalimentación inmediata y la garantía en el cumplimiento de sus expectativas a través del desarrollo de un software que satisface sus necesidades, y que además exhibe atributos de calidad competitivos. Esto no solo se logra por el MCVS elegido, sino también por la plataforma que permite la automatización de los requerimientos y exige la interacción entre los especialistas en la gestión de estos y el usuario.

En la misma línea, se describen los logros alcanzados en cada una de las etapas de la metodología propuesta, haciendo énfasis en el diseño del sistema de información, y los resultados permiten evidenciar el cumplimiento de los objetivos del proyecto. Asimismo, se da cuenta del proceso de gestión de requisitos funcionales y no funcionales con la ayuda del procedimiento PEVReS, de la realización de los modelos conceptuales bajo el estándar UML

haciendo uso de herramientas CASE, y del desarrollo del sistema de información web a partir de prototipos.

Finalmente, es importante mencionar que ya se hizo la validación del modelo, a manera de experimento, y se encuentra en proceso de publicación, razón por la cual no se presenta en este capítulo.

VI. REFERENCIAS

- [1] R. S. Pressman, *Ingeniería del Software. Un Enfoque Práctico*. México: McGrawHill, 2010.
- [2] S. R. Gómez Palomino y E. Moraleda Gil, *Aproximación a la ingeniería del software*. Madrid: Editorial Centro de Estudios Ramón Areces S. A., 2014.
- [3] A. Toro y J. G. Gálvez, “Especificación de requisitos de software: una mirada desde la revisión teórica de antecedentes”, *Entre Cienc. e Ing.*, vol. 10, n.º 19, pp. 108-113, 2016.
- [4] IEEE Computer Society, *Swebok V3.0*, n.º V3.0, 2014.
- [5] L. Peláez, A. Toro, J. López, y A. Medina, “Caracterización del proceso de desarrollo de software en Colombia: una mirada desde las PYMES productoras”, *Rev. PÁGINAS*, vol. 92, pp. 89-98, 2012.
- [6] P. Bourque, R. E. Richard, E. Fairley, and IEEE Computer Society, *Guide to the software engineering body of knowledge (SWEBOK)*, 2014.
- [7] F. Ramírez García y P. Puello Marrugo, “Gestión de Requisitos en la Ingeniería de Software”, *Ingeniator Rev. Virtual los Programas Ing.*, vol. 1, n.º 1, pp. 57-65, 2009.
- [8] A. Toro Lazo, J. G. Gálvez Botero, y L. E. Peláez-Valencia, “Procedimiento para especificar y validar requisitos de software”, en *Investigación e Innovación en Ingeniería de Software*, 2.ª ed., Medellín: Sello Editorial TdeA, 2018.
- [9] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, and L. Tripp, “Guide to the software engineering body of knowledge, Version 3.0”, Piscataway, NJ, USA, 2014.
- [10] A. Toro Lazo, J. G. Gálvez Botero, y S. V. Hurtado Gil, “Procedimiento para especificar y validar requisitos de software en mipymes desarrolladoras de software de la ciudad de Pereira, basado en estudios previos en la región”, *Universidad Autónoma de Manizales*, 2017.
- [11] U. J. Bonaparte, “Proyectos UML Diagramas de clases y

- aplicaciones JAVA en NetBeans 6.9.1.”, Cátedra Paradigmas de Programación, Univ. Tecnológica Nacional, Argentina, 2012.
- [12] L. E. Peláez-Valencia, A. Toro-Lazo, L. D. Ledesma-Mariño, D. E. Rodríguez-Franco, y E. H. Largo-Bueno, “Revisión de herramientas para la gestión de requisitos: un aporte al aseguramiento de la calidad del software”, *Vent. Inform.*, n.º 38, Ene. 2018. doi: 10.30554/ventanainform.38.2865.2018
- [13] E. M. Méndez Nava, “Modelo de evaluación de metodologías para el desarrollo de software”, Universidad Católica Andrés Bello, 2006.
- [14] M. Arias Chávez, “La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software”, *InterSedes Rev. las Sedes Reg.*, vol. 6, n.º 10, pp. 1-13, 2011.
- [15] IEEE Computer Society, *SWEBOK V3.0. Guide to the Software Engineering Body of Knowledge*. NJ, 2014.
- [16] L. E. Peláez Valencia, A. Toro Lazo, D. E. Rodríguez Franco, y L. D. Ledesma Mariño, “El aseguramiento de la calidad de los requerimientos en la industria local del software : caso Pereira”, en *Nuevas tendencias investigativas en computación, informática y educación en ingeniería*, 2019, pp. 48-69.
- [17] R. Pressman y B. R. Maxim, *Ingeniería de Software, un enfoque práctico*, 8.ª ed. Nueva York: McGraw-Hill, 2015.
- [18] I. Sommerville, *Software Engineering*, 10th Ed. London: Pea, 2016.
- [19] K. Wiegers and J. Beatty, *Software Requirements*. Washington, 2013.
- [20] International Requirements Engineering Board (IREB), *International Requirements Engineering Board*, 2020.
- [21] ISO, “International Organization for Standardization”, *The International Organization for Standardization*, 2020.
- [22] A. N. Camacho Zambrano, “Herramienta para el análisis de requerimientos dentro de la pequeña empresa desarrolladora de software en Bogota”, Pontificia Universidad Javeriana, 2005.
- [23] O. C. Medina, M. M. Marciszack, y M. A. Groppo, “Trazabilidad y validación de requerimientos funcionales de sistemas informáticos mediante

- la transformación de modelos conceptuales”, ReCIBE. Revista Electrónica de Computación, Informática, Biomédica y Electrónica, vol. 5, n.º 1, 2016 [En línea]. Disponible en: <http://recibe.cucei.udg.mx/ojs/index.php/ReCIBE/article/view/53>
- [24] A. Toro Lazo y J. G. Gálvez Botero, “Procedimiento PEVReS”, 2017.
- [25] J. G. Gálvez, A. Toro, y S. V. Hurtado, “Procedimiento para especificar y validar requisitos de software en MiPymes desarrolladoras de software, basado en estudios previos de la región”, Universidad Autónoma de Manizales, 2017.
- [26] J. Quiroz, “El modelo relacional de bases de datos”, Boletín de Política Informática, n.º 6, pp. 53-61, 2010.
- [27] B. Unhelkar, Software Engineering with UML. Boca Raton, Florida: Taylor & Francis Group, 2018.
- [28] L. E. Peláez, “Validación de un modelo propio para el desarrollo y la gestión de proyectos de software”, Pereira, 2014.
- [29] IEEE, Guide to Enterprise IT Body of Knowledge (EITBOK). 2017.
- [30] S. L. Buitrón, B. L. Flores-Ríos, y F. J. Pino, “Elicitación de requisitos no funcionales basada en la gestión de conocimiento de los stakeholders”, Ingeniare. Rev. Chil. Ing., vol. 26, n.º 1, pp. 142-156, 2018. doi: 10.4067/S0718-33052018000100142.
- [31] Project Management Institute (PMI), Guía de los FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (Guía del PMBOK) 6.^a ed., Newtown Square, Pennsylvania: PMI, 2017.