



# Investigación e Innovación en Ingeniería de Software

Volumen 2



# Investigación e Innovación

en Ingeniería de Software

Volumen 2



**Tecnológico**  
**de Antioquia**  
**Institución Universitaria**

*Investigación e Innovación en Ingeniería de Software / volumen 2*  
Fabio Alberto Vargas Agudelo, Darío Enrique Soto Duran, Juan  
Camilo Giraldo Mejía, compiladores. – Medellín : Publicar T,  
Sello Editorial TdeA, 2018.

178 páginas. ; Ilustraciones a color.  
Incluye referencias bibliográficas.

**ISBN: 978-958-59925-8-0**

**Corrección de estilo, diseño, diagramación y animación**

Divegráficas S.A.S.

divegraficas@gmail.com

Derechos reservados del Tecnológico de Antioquia – Institución  
Universitaria

Los capítulos publicados incorporan contenidos derivados de procesos de investigación académica, que cumplen una función social, sin embargo, no representan los criterios institucionales del Tecnológico de Antioquia – Institución Universitaria. Los contenidos son responsabilidad exclusiva de los autores, y cualquier observación o cuestionamiento sobre la originalidad de los textos puede ser notificada al correo de los autores. El Tecnológico de Antioquia - Institución Universitaria autoriza la reproducción parcial de los textos con fines exclusivamente académicos, dando estricto cumplimiento a las normas de referenciación bibliográfica en favor de los autores y de las instituciones editoras. Cualquier uso diferente requerirá autorización escrita de los compiladores, y su omisión inducirá a las acciones legales dispuestas por las leyes internacionales sobre la propiedad intelectual y derechos de autor.

Hecho en Colombia





*Investigación e Innovación en Ingeniería de Software*  
Tecnológico de Antioquia – Institución Universitaria

Publicar T – Sello Editorial TdeA  
Tecnológico de Antioquia – Institución Universitaria  
Calle 78B No. 72A – 220  
PBX: (+57)(4) 444 37 00  
[www.tdea.edu.co](http://www.tdea.edu.co)  
Medellín - Colombia

**Dr. Lorenzo Portocarrero Sierra**  
Rector

**Dr. Fabio Alberto Vargas Agudelo**  
Director de Investigación – Coordinación del Sello Editorial

**Dr. Darío Enrique Soto Duran**  
Decano de la Facultad de Ingeniería

**Dr. Lorenzo Portocarrero Sierra**

Rector

**Dr. Fabio Alberto Vargas Agudelo**

Director de Investigación – Coordinación del Sello Editorial

**Dr. Dario Enrique Soto Duran**

Decano de la Facultad de Ingeniería

**Dr. Juan Camilo Giraldo Mejía**

Lider de Grupo GIISTA

***Autores:***

Juan Carlos Valdés Quintero  
Juan Camilo Parra Toro  
John Fernando Escobar Martínez  
Gloria Yaned Tobón Cataño  
Mónica María Rojas R  
Adriana Xiomara Reyes Gamboa  
Jovani Alberto Jimenez Builes  
Dario Enrique Soto Duran  
Jorge Andrick Parra Valencia  
Byron Cuesta Quintero  
Víctor Daniel Gil Vera  
Ramiro Antonio Giraldo Escobar  
Lina María Montoya Suarez  
Évelin Cristina Quiceno Sosa  
Ricardo de J. Botero Tabares  
Roberto Guevara Calume  
Carlos Arturo Castro Castro  
Leon Daniel Jaramillo Ramírez  
Raúl Ivan Mazo Peña  
Gloria Lucía Giraldo Gómez  
Paola Andrea Noreña Cardona  
Carlos Mario Zapata Jaramillo  
Aixa Eileen Villamizar Jaimes  
Milton Jesús Vera Contreras  
Matías Herrera Cáceres  
Oscar Alberto Gallardo Pérez

Beatriz Eugenia Marín  
Antonio Pantoja  
Daniel Ernesto Osorio  
Miguel Andrés Idrobo  
Edwin Núñez Ortiz  
Alonso Toro Lazo  
Juan Guillermo Gálvez Botero  
Luis Eduardo Peláez Valencia  
Sandra Victoria Hurtado Gil  
Juan Luis Arias Vargas  
Iván Andres Delgado  
John A. Bohada Jaime  
Geiver Vianey Ríos Gutiérrez  
Katerine Villamizar Suaza  
Juan Camilo Giraldo Mejía  
Mauricio Amariles Camacho  
Fabio Alberto Vargas Agudelo  
Claudia Elena Durango Vanegas

**ISBN: 978-958-59925-8-0**

*Investigación e Innovación en Ingeniería de Software*  
/ volumen 2 Fabio Alberto Vargas Agudelo, Dario Enrique Soto, Juan Camilo Giraldo, compiladores. –  
Medellín : Publicar T, Sello Editorial TdeA, 2018.

---

7	<b>PREFACIO</b>
.....	
9	<b>Capítulo I UN FRAMEWORK PARA LA DEFINICIÓN DE MÉTRICAS EN INGENIERÍA DE LÍNEAS DE PRODUCTOS</b>
.....	
19	<b>Capítulo II ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE: ACERCAMIENTO AL MARCO TEÓRICO Y AL ESTADO DE LA CUESTIÓN DESDE LA DISCIPLINA DE LA INGENIERÍA DE SOFTWARE</b>
.....	
31	<b>Capítulo III METODOLOGÍAS CERCANAS AL DESARROLLO DE SOFTWARE LIBRE Y DE CÓDIGO ABIERTO (FOSS). EVALUANDO EL PROCESO DE DESARROLLO DESDE LA COOPERACIÓN</b>
.....	
43	<b>Capítulo IV REVISIÓN SISTEMÁTICA A LOS CONTENIDOS DIGITALES INTERACTIVOS ELABORADOS CON PRÁCTICAS ÁGILES</b>
.....	
55	<b>Capítulo V GESTIÓN DE RIESGOS ASOCIADOS A LAS PERSONAS EN PROYECTOS CON SCRUM: UNA REVISIÓN BIBLIOGRÁFICA</b>
.....	
69	<b>Capítulo VI REPRESENTACIÓN DE EVENTOS A PARTIR DE ESTRUCTURAS LINGÜÍSTICAS BASADAS EN ROLES SEMÁNTICOS: UNA EXTENSIÓN AL ESQUEMA PRECONCEPTUAL</b>
.....	
81	<b>Capítulo VII REPRESENTACIÓN EN EL NÚCLEO DE SEMAT DEL MODELO MADCE-TVD</b>
.....	
89	<b>Capítulo VIII OBJETOS VIRTUALES DE APRENDIZAJE EN LA EDUCACIÓN SUPERIOR: UN CASO DE APLICACIÓN</b>
.....	
101	<b>Capítulo IX LA MODA DE CLOUD COMPUTING: DESAFÍOS CURRICULARES</b>
.....	

---

111	<b>Capítulo X</b> <b>GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN EN LAS ORGANIZACIONES</b>
123	<b>Capítulo XI</b> <b>SISTEMAS DE SOPORTE A LAS DECISIONES ESPACIALES CASO DE APLICACIÓN: MAPIFICACIÓN DE SERVICIOS ECOSISTÉMICOS EN LA GESTIÓN TERRITORIAL URBANA</b>
139	<b>Capítulo XII</b> <b>MODELO DE MONITOREO Y MEJORA DE PROCESOS PARA PYMES DEL SECTOR JURÍDICO</b>
147	<b>Capítulo XIII</b> <b>PROCEDIMIENTO PARA ESPECIFICAR Y VALIDAR REQUISITOS DE SOFTWARE</b>
161	<b>Capítulo XIV</b> <b>MANEJO DE INFORMACIÓN GEOGRÁFICA A TRAVÉS DE HERRAMIENTAS INTEGRADAS. CASO DE ESTUDIO: CENSO ARBÓREO MUNICIPIO DE TULUÁ</b>
169	<b>Capítulo XV</b> <b>REVISIÓN SISTEMÁTICA DE LITERATURA: FACTORES QUE AFECTAN EL PROCESO DE PRUEBAS DE SOFTWARE</b>



La Facultad de Ingeniería y el grupo de investigación GIISTA del Tecnológico de Antioquia - Institución Universitaria, conscientes de las necesidades de desarrollo de las nuevas tecnologías de la información y las comunicaciones, sus métodos y estrategias de investigación y sus nuevos desarrollos, compilan resultados de investigaciones de la comunidad académica e investigativa del país y exponen diferentes enfoques de la disciplina a través de la segunda edición del libro denominado: Investigación e innovación de la ingeniería de software.

En la actualidad, la frecuente interacción con las redes sociales y la operatividad de dispositivos móviles han contribuido a que el volumen de datos cada vez sea mayor, por lo tanto, la necesidad de métodos y herramientas más óptimos es fundamental para la gestión de la información y la obtención de nuevo conocimiento, por lo indispensable que ambos han llegado a ser en un mundo globalizado, y por la cada vez mayor producción de nuevo conocimiento y el almacenamiento de más y más información. Sin embargo, aún no se sabe específicamente en qué nivel de desarrollo se encuentra. La Ingeniería de Software contribuye de manera efectiva a la materialización de estas nuevas estrategias, lo cual, sumado a la gestión de conocimiento, genera un amplio campo de estudio para los investigadores, dado que los procesos de desarrollo de software, innovación tecnológica y, en general, de las tecnologías de la información experimentan grandes transformaciones y lo seguirán haciendo en el futuro cercano.

Esta segunda edición presenta quince capítulos dirigidos a la comunidad académica en general, estudiantes, docentes, semilleros, investigadores, desarrolladores, etc., y al sector empresarial enfocado en el desarrollo de software, con temáticas que contribuyen al interés en las áreas de la ingeniería de software, los ambientes virtuales de aprendizaje, la gestión del conocimiento y la seguridad de la información. El libro está conformado por capítulos autónomos e inéditos resultado de investigaciones, cada uno estructurado en varias secciones, a saber: contextualización, desarrollo, resultados, conclusiones y trabajos futuros de cada temática abordada.

Teniendo como premisa la cooperación académica, el libro Investigación e innovación de la ingeniería de software genera un espacio para la disertación, la transferencia y la apropiación del conocimiento en contextos de investigación asociados a los programas académicos en los niveles técnico, tecnológico, profesional y posgradual, de los programas del área de informática que las IES ofertan en concordancia con sus procesos misionales. Confiamos en que los resultados plasmados en este nuevo volumen contribuyan a estimular nuevos desarrollos y proyectos de investigación en el área de informática y de ingeniería de software de las IES del país.

**Fabio A. Vargas**  
**Darío E. Soto**  
**Juan C. Giraldo.**  
Compiladores



# Capítulo I

## UN FRAMEWORK PARA LA DEFINICIÓN DE MÉTRICAS EN INGENIERÍA DE LÍNEAS DE PRODUCTOS

**León Daniel Jaramillo Ramírez** - [leon.jaramillo@tdea.edu.co](mailto:leon.jaramillo@tdea.edu.co)

Docente, Tecnológico de Antioquia – Institución Universitaria, Medellín, Colombia

**Raúl Ivan Mazo Peña** - [rimazop@eafit.edu.co](mailto:rimazop@eafit.edu.co)

Profesor asociado de la Université Pantheon-Sorbonne, París, Francia

**Gloria Lucía Giraldo Gómez** - [glgiraldog@unal.edu.co](mailto:glgiraldog@unal.edu.co)

Profesora asociada de la Universidad Nacional de Colombia, Medellín, Colombia

### I. INTRODUCCIÓN

Una línea de productos es un conjunto de productos relacionados que cuentan con características comunes, las cuales posibilitan su producción masiva y características variables que facilitan su diferenciación, orientada a la satisfacción de requisitos particulares [1]. Las líneas de productos (LP) son una práctica exitosa y ampliamente utilizada en dominios tales como el aeronáutico, el automotriz y los sistemas embebidos<sup>1</sup>. Esta práctica se utiliza también en la ingeniería de software, y da lugar a las líneas de productos de software. Una línea de productos de software (LPS) es un conjunto de aplicaciones de software similares definidas para un segmento de mercado, las cuales comparten unos requisitos comunes y además cuentan con unos requisitos variables entre ellas [2].

El desarrollo de líneas de productos y de líneas de productos de software implica aplicar ingeniería de líneas de productos. La ingeniería de líneas de productos (ILP) surgió como un paradigma que busca aprovechar eficientemente las similitudes y las variaciones presentes en el conjunto de productos que constituyen una línea [1]. Dentro de la ILP se distinguen dos procesos: la ingeniería del dominio y la ingeniería de aplicación. La ingeniería del dominio es el proceso en el cual se realiza el análisis del dominio de una línea de productos y se desarrollan los artefactos reutilizables [3]. Por su parte, la ingeniería

de aplicación tiene como objetivo el desarrollo de productos específicos, atendiendo requisitos particulares [3] y haciendo uso de los artefactos reutilizables obtenidos en la ingeniería del dominio.

Por otro lado, la ILP se puede abordar desde dos perspectivas: la perspectiva de los clientes o interesados (*stakeholders*) y la del vendedor o desarrollador. Estas perspectivas se denominan, respectivamente, el espacio del problema y el espacio de la solución. El espacio del problema se refiere a especificaciones del sistema definidas durante el análisis del dominio y el proceso de ingeniería de requisitos de la línea de productos [4]. Por otro lado, el espacio de la solución se refiere a los artefactos creados durante las fases de especificación de la arquitectura, de diseño y de implementación de los componentes de la LP [4].

Un artefacto importante en el espacio del problema son los modelos de variabilidad. Un modelo de variabilidad es una abstracción que representa los requisitos del dominio de los productos de la línea de productos [1]. Dichos modelos se construyen inicialmente en el proceso de ingeniería del dominio y permiten representar la variabilidad de una línea de productos, así como sus elementos comunes. En la literatura se proponen diferentes formalismos o notaciones para el modelado. Entre otros, se encuentran los modelos de características (*o feature models* - FM) [5], los modelos de variabilidad ortogonal (*o Orthogonal Variability Model* -OVM-) [1] y los modelos de variabilidad basados en restricciones [6].

<sup>1</sup> Product Line Hall of Fame: <http://splc.net/fame.html>

Un concepto que atañe al presente trabajo es el proceso de medición. La medición tiene una gran importancia en la ciencia y la ingeniería. Su uso es intensivo en áreas de estudio maduras, lo que permite en ellas una evaluación efectiva de productos, procesos y experimentos [7], [8]. La medición se puede definir como un proceso en el cual se asignan números (o símbolos) a atributos de entidades en el mundo real, de tal manera que los describan de acuerdo con unas reglas claramente definidas [9].

La medición tiene un rol cada vez más importante en la ingeniería de software, por ejemplo, en la evaluación de proyectos y productos y en la predicción de algunas variables como el costo y la calidad. También es la piedra angular de iniciativas como SW-CMM (*Capability Maturity Model for Software*), ISO/IEC 15504 (SPICE, *Software Process Improvement and Capability Determination*) y CMMI (*Capability Maturity Model Integration*) [8]. Dentro de la ingeniería de software, los atributos para medir suelen ser atributos de calidad y las entidades para medir suelen ser artefactos del proceso de ingeniería de software [10].

La medición es también un proceso de gran importancia en la ILP [7]. Éste permite el control, monitoreo, entendimiento y predicción [8] de diferentes atributos de una línea de productos. El control de estos atributos es muy importante en este paradigma, incluso más que en la ingeniería de software tradicional, ya que un diseño inadecuado o un defecto en un artefacto de la línea de productos tiene el potencial de propagarse a diferentes productos simultáneamente [1].

A pesar de la importancia de la medición en la ingeniería de software y en la ILP, una revisión de la literatura muestra que no hay un consenso general respecto a la terminología de la medición dentro de dicho marco. Si bien existen propuestas para un marco conceptual común [8], [11], [12] y estándares relacionados con la medición [13–15], aún no se llega a un consenso ni en la terminología ni en el marco conceptual. Además, tampoco logran facilitar la definición de métricas en la ILP. Por eso en el presente trabajo se propone un *framework* para la definición de métricas en ILP.

El *framework* propuesto tiene como propósito proveer un marco conceptual que facilite la definición y aplicación de métricas en la ingeniería de líneas de productos. Para esto se especifican de forma clara y precisa los conceptos relacionados con la medición, y su lugar dentro de la ingeniería de líneas de productos. También busca guiar el proceso de definición de una métrica mediante la adaptación de enfoques que faciliten su claridad conceptual y su aplicación [11], [16].

En la literatura se pueden encontrar diferentes trabajos orientados a facilitar el proceso de medición, tanto en la ingeniería de software como en la ILP. Zuse [17] y Fenton y Pfleeger [9] proponen marcos conceptuales amplios y rigurosos para la medición en la ingeniería de software. Sin embargo, precisamente su rigor dificulta su aplicación en entornos no académicos. Se pueden encontrar también estándares internacionales de medición. Tanto la IEEE [13] como el *Joint Committee for Guides in Metrology* (JCGM) [18] proponen vocabularios estándar, y en el primer caso, pautas para el proceso de medición. Aunque en ambos casos se definen claramente terminologías para la medición, éstas no constituyen *frameworks* que faciliten la aplicación de dicho proceso. Por otro lado, Basili [16], Kitchenham et al. [11], Purao y Vaishnavi [19], García et al. [8] y Habra et al. [20] proponen diferentes marcos conceptuales para la definición de métricas y de procesos de medición en la ingeniería de software. Sin embargo, dichos trabajos presentan dos limitantes con respecto al *framework* propuesto en el presente trabajo. En primer lugar, en ellos se busca generar un consenso sobre una terminología unificada para la medición en la ingeniería de software, siendo a menudo contradictorios entre sí. Un ejemplo de lo anterior se da al definir los términos *metric* y *measure*: Kitchenham et al. [11] utilizan ambos términos de manera intercambiable, Purao y Vaishnavi toman *metric* como un concepto central en su propuesta [19], mientras que García et al. [8] descartan el término *metric* en favor de *measure*. Sin embargo, para la IEEE [13] y Fenton y Pfleeger [9] se trata de dos conceptos diferentes. En segundo lugar, los trabajos anteriores se enfocan en la definición de métricas en la ingeniería de software tradicional, y dejan por fuera aspectos importantes de la ILP, como son la medición de varios productos en lugar de uno solo o su aplicación en el ciclo de vida de las líneas de

productos. Esto es relevante, ya que en la ILP no solo se definen métricas en función de un producto, sino de una familia de productos [7]. Finalmente, Sant'Anna et al. [21], Her et al. [22], Ding [23] y Venkitachalam et al. [24] proponen *frameworks* de medición aplicables en ILP, y aunque están orientados a la medición de determinados atributos en algunos de los artefactos de las líneas de productos, no proveen un marco conceptual general que permita definir métricas a lo largo del ciclo de vida de las líneas de productos.

Dadas las limitantes encontradas en los trabajos mencionados previamente, en el presente documento se propone un *framework* para la definición de métricas en ILP. Éste facilita la definición de métricas a lo largo del ciclo de vida de una línea de productos, proveyendo un marco conceptual y pautas que faciliten dicha definición. Lo anterior, teniendo en cuenta conceptos del área de la medición en general y de la ILP en particular.

El resto del presente documento se organiza de la siguiente manera: en la sección II se describe la metodología utilizada para la construcción del *framework*; luego, en la sección III se detalla el *framework* propuesto, y, finalmente, en la sección IV se listan las conclusiones del presente trabajo.

## II. METODOLOGÍA

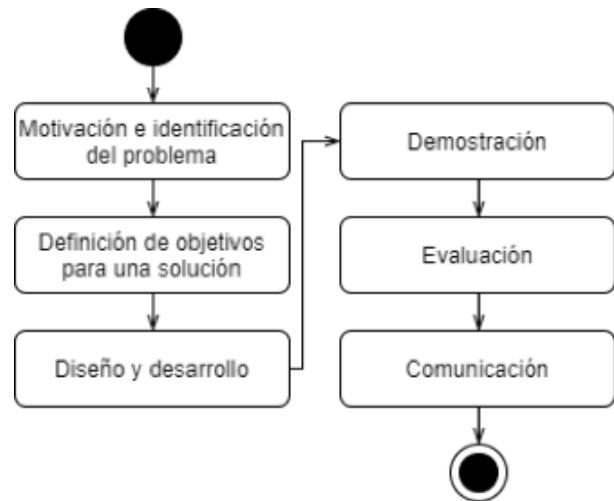
Para la definición del *framework* se usa una adaptación de la metodología *design science*. Ésta es una metodología utilizada para realizar procesos de investigación y desarrollo en diferentes áreas, incluyendo el área de TI [25]. A grandes rasgos, *design science* consiste en la realización de los siguientes procesos [25]:

1. Diseñar un artefacto que mejore algo dentro de un contexto determinado.
2. Evaluar el rendimiento de dicho artefacto dentro del contexto.

En *design science* un artefacto puede ser un modelo, un método, una construcción, un recurso técnico, social o informacional, entre otros [26]. En el presente trabajo el artefacto lo constituyen el *framework* y las pautas propuestas que se deben seguir para la

definición de métricas en ILP.

Peffer et al. [26] proponen una secuencia de actividades para realizar en un proyecto en el cual se aplique la metodología *design science*. La Figura 1 presenta la secuencia de dichas actividades.



**Fig. 1.** Actividades para la metodología *design science* propuestas por Peffer et al. [26]

En el presente trabajo se desarrollan las cuatro primeras actividades. *La motivación e identificación del problema* consiste en la definición del problema de investigación específico y la justificación de su solución [26]. *La definición de objetivos* comprende la inferencia, desde la definición del problema, de los objetivos por alcanzar para una solución específica [26]. *El diseño y desarrollo* es la actividad que incluye determinar la funcionalidad deseada en el artefacto, diseñar su arquitectura y finalmente crearlo [26]. Por último, se procede con las actividades de *demonstración* y de *evaluación*. En el presente trabajo se realizó únicamente la actividad de demostración. Esta actividad es aquella en la cual se demuestra el uso del artefacto para resolver una o más instancias del problema, y puede involucrar el uso de experimentación, de simulación, de un caso de estudio, de una prueba de concepto o de otra actividad apropiada [26].

La elaboración del presente documento hace parte de la actividad de comunicación, la cual se ocupa explícitamente de la comunicación del problema que se desea resolver, del artefacto obtenido y de cómo éste resuelve dicho problema [26].

### III. RESULTADOS

En esta sección se presenta el framework propuesto para la definición de métricas en ingeniería de líneas de productos. Se trata específicamente de un framework conceptual. Smyth [27] define un framework conceptual como un marco de trabajo estructurado mediante “un conjunto de ideas y teorías que ayudan a un investigador a identificar apropiadamente el problema que se está buscando, a enmarcar sus preguntas y a encontrar una literatura previa apropiada”. El framework propuesto a continuación tiene como objetivo proveer un marco conceptual, el cual facilite la definición y aplicación de métricas en ILP.

#### A. Framework Conceptual

En la Figura 2 se presenta el framework propuesto para la medición en la ingeniería de líneas de productos. El framework muestra los conceptos que se deben tener en cuenta al momento de definir una métrica en ILP, así como la relación entre éstos. Se trata de un modelo de dominio, representado con la notación del diagrama de clases de UML. En el framework propuesto se pueden observar dos perspectivas: La primera, son los conceptos clave en la medición en la ingeniería de software. La segunda perspectiva, delimitada mediante una línea discontinua verde, son los conceptos relacionados con la ILP que se deben tener en cuenta en el proceso de medición.

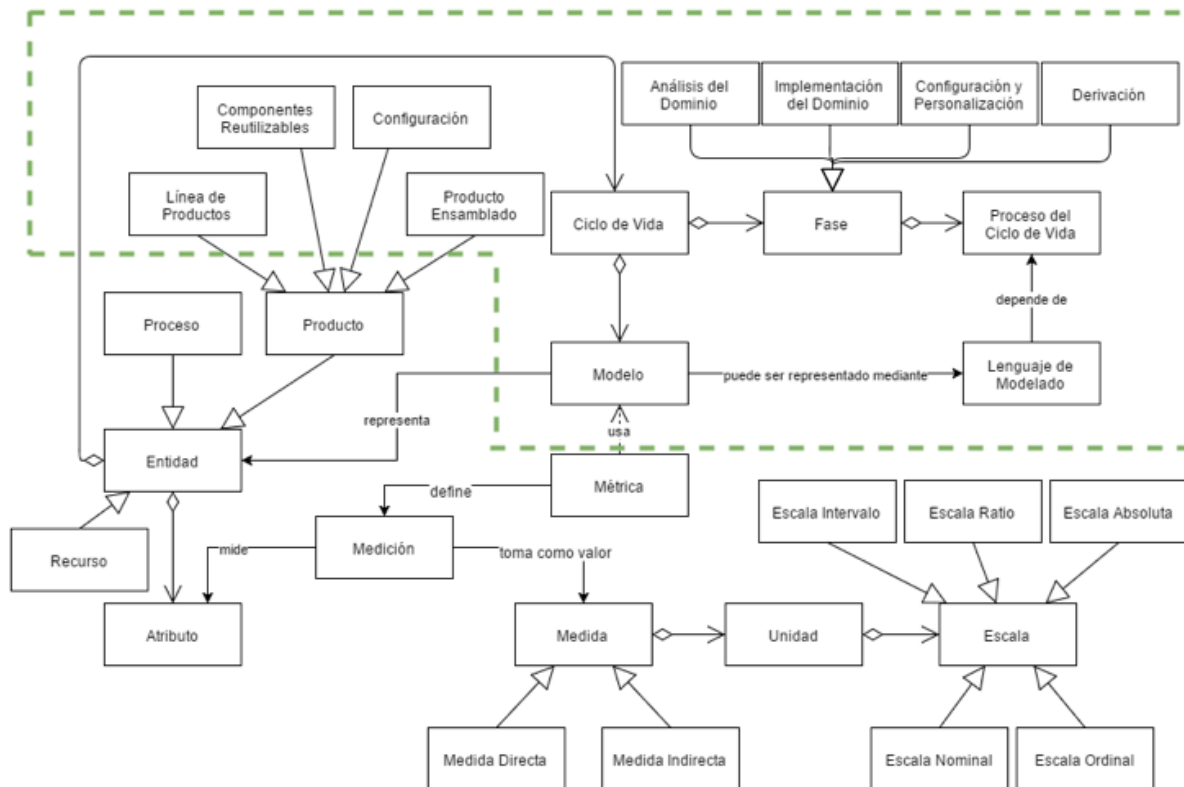


Fig. 2. Framework para la elaboración de métricas en ingeniería de líneas de productos

Durante el proceso de medición es necesario tener claridad conceptual; esto ayuda no solo a la definición de métricas válidas [11], sino también a su uso efectivo [7]. Sin embargo, en muchos casos este último aspecto no se considera. Con el fin de facilitar la comprensión y la aplicación del *framework* propuesto, a continuación se explican los conceptos presentes en él.

Al momento de definir una métrica, en primer lugar es necesario definir la entidad. Una *entidad* es el objeto o evento del mundo real que se va a medir [9]. Así como en un entorno de manufactura, una entidad puede ser un bien fabricado o una de sus piezas, en ILP, una entidad puede ser una línea de productos completa, un componente reutilizable o incluso el proceso de desarrollo de éste.

En el proceso de medición, una entidad puede ser de tipo *producto*, proceso o recurso [28]. El proceso describe las actividades que se llevan a cabo durante el desarrollo de un producto o, en este caso, de una línea de productos [28]. El *producto* es cualquier resultado concreto del proceso, en ILP éste puede ser: la línea de productos, una configuración, un conjunto de componentes reutilizables o un producto ya terminado. Finalmente, los *recursos* son las entradas o insumos necesarios para ejecutar el proceso, tales como hardware, software o personal [28].

Una entidad de cualquier tipo posee atributos. Un *atributo* es una propiedad física o abstracta medible de una entidad [13]. Por ejemplo, una línea de productos posee atributos como el grado de variación, que es el grado en el cual la línea cuenta con características variables, en contraste con sus características comunes [29]. También se pueden observar de manera más intuitiva los atributos de entidades físicas como son el peso, el volumen o la temperatura de un objeto.

A continuación, se debe definir cómo se mide un atributo. Esto se lleva a cabo mediante la medición. La *medición* es el proceso mediante el cual se le asigna una medida a la entidad, describiendo el grado con el cual ésta posee un atributo dado [13]. Esta *medida* puede ser un número o símbolo [28], y puede ser directa o indirecta dependiendo de cómo se obtenga. Una medida directa es aquella que se puede obtener directamente, ya que no requiere ni se relaciona con

la medición de otros atributos [9]. Un ejemplo de una *medida directa* es el número de características de un modelo de características o el número de restricciones presentes en él. Por otro lado, una medida indirecta está relacionada con la medición de otros atributos, y por lo general se deriva de cálculos realizados sobre las medidas obtenidas de aquéllos [9]. Un ejemplo de una *medida indirecta* es la densidad de defectos de un modelo de variabilidad, la cual es producto de un cálculo que involucra el número de variables del modelo y el número de defectos presentes en él [1]. Una medida puede ser, además, subjetiva u objetiva. Una *medida subjetiva* es aquella en la que hay un juicio o criterio por parte del sujeto que la hace, influyendo en su resultado [9]. Por otro lado, en una *medida objetiva* no existe influencia de ningún tipo de juicio por parte del sujeto y, por el contrario, depende completamente de la entidad y el proceso de medición [9].

Una propiedad importante a tener en cuenta al definir una métrica válida es su unidad [11]. La *unidad* de una medida provee valiosa información de ella [11]. Por ejemplo, no es posible determinar de manera confiable la longitud de un objeto sin expresar su unidad, que podría ser centímetros, pies o kilómetros. Conocer la unidad de una medida permite saber, entre otras cosas, las posibilidades de transformación entre sus unidades, lo cual viene determinado por su escala [9]. La escala es el modo en el que se hace el mapeo del atributo desde la entidad medida (mundo real) al mundo formal o matemático [9]. La *escala* puede ser de uno de cinco tipos. De tipo *nominal*, es decir, que mapea el atributo de una entidad a un nombre o símbolo predefinido sin que este tenga una magnitud asociada. De tipo ordinal, que clasifica las entidades de acuerdo con un criterio de ordenamiento sin que haya un valor de magnitud significativo asociado a cada clase. De tipo *intervalo*, se usa cuando la diferencia entre dos medidas es significativa, mas no su valor ni la proporción entre dos valores dentro de la escala. De tipo ratio, es cuando existe un valor cero, es decir que es posible la ausencia del atributo medido, y la proporción entre dos escalas es significativa. Y de tipo *absoluta*, es un caso especial del tipo ratio, en el cual se mide mediante el conteo de elementos en la entidad; esto implica que el valor en sí de la medida es la única transformación significativa posible [9].



Cabe resaltar que, aunque la escala es una propiedad importante de una medida, en la mayor parte de las métricas estudiadas en la revisión de literatura del presente trabajo la escala no se hace explícita.

Finalmente, con respecto a la medición tenemos el concepto de *métrica*, que se define como una función que cuenta con entradas y una salida definidas, y mediante la cual se asigna un valor que describe el atributo seleccionado de una entidad dada [13].

Como puede apreciarse en la Figura 2, el concepto métrica se relaciona con conceptos pertenecientes al dominio de la ILP. Nuestra propuesta los considera necesarios en la definición de una métrica dentro de este dominio.

En la definición de una métrica es común el uso de modelos. Un *modelo* es una representación abstracta de la entidad, la cual facilita su medición. El modelo permite la definición de una métrica que caracterice, efectivamente, un atributo de la entidad, independientemente del significado intuitivo (y a veces ambiguo) que pueda tener ésta [11]. Como en otras ramas de la ingeniería, en ILP un modelo puede ser representado mediante un *lenguaje de modelado*. Por ejemplo, en un modelo de variabilidad se pueden usar diferentes lenguajes de modelado (como *feature models* u OVM) y en el modelo de componentes se puede usar la notación UML.

En este punto es conveniente retomar el concepto de entidad. Toda entidad que se va a medir cuenta con un *ciclo de vida*, que en la ILP son las fases o etapas por las que atraviesa una línea de productos, desde su concepción hasta la obtención de productos concretos [1]. Para el presente capítulo se tienen en cuenta cuatro *fases*: el análisis del dominio, la implementación del dominio, la fase de configuración y personalización y la fase de derivación [3].

A su vez, las fases comprenden, cada una, un conjunto de *procesos del ciclo de vida*. La fase de *análisis del dominio* comprende los procesos de ingeniería de requisitos del dominio, de definición de la arquitectura de referencia de la línea de productos y de modelado de la variabilidad. La fase de *implementación del dominio* comprende los procesos de ingeniería

de requisitos de los componentes reutilizables de dominio, de implementación de dichos componentes, de realización de pruebas unitarias y de enlace de esos componentes con sus respectivos requisitos de dominio. La fase de *configuración y personalización* comprende los procesos de ingeniería de requisitos, de análisis, de definición de la arquitectura y de personalización del producto. Y la fase de *derivación* comprende los procesos de ensamblaje y de pruebas de integración del producto derivado [3].

El lenguaje de modelado del modelo usado en la definición de la métrica, depende en gran medida del proceso del ciclo de vida en el que se desea usarla. Por ejemplo, una métrica definida para un proceso de la fase de análisis del dominio, por lo general, hace uso de un modelo de variabilidad, el cual a su vez puede ser un modelo de características (feature model) o un modelo OVM, entre otros.

## **B. Pautas para la Definición de Métricas en Ingeniería de Líneas de Productos**

A continuación se establecen unas pautas para la definición de métricas en ingeniería de líneas de productos, las cuales tienen en cuenta los conceptos del *framework* propuesto que facilitan la definición y aplicación de métricas válidas.

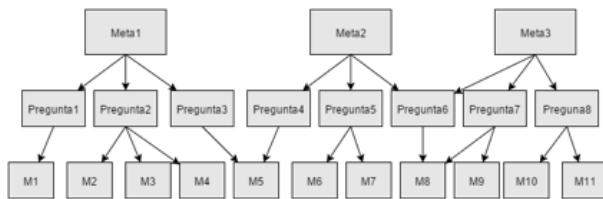
En primer lugar, es necesario definir el porqué de la métrica, es decir, el motivo o el objetivo que se desea lograr al definirla y aplicarla. Uno de los métodos más usados [30–32] para este fin es el paradigma *Goal/Question/Metric* (GQM) [16]. El paradigma GQM es un mecanismo para la definición de metas u objetivos operacionales y su evaluación usando métricas. Se trata de un enfoque sistemático en el que se establece una relación entre las metas y los modelos de procesos y productos industriales [33] y de software [13], con base en necesidades específicas de un proyecto o de la organización [13]. Si bien el paradigma GQM fue desarrollado para su aplicación en un proyecto específico, su uso se ha expandido a un contexto más amplio [34].



Aplicar el paradigma GQM implica [13], [16]:

1. Desarrollar un conjunto de metas de calidad y productividad en el nivel corporativo, de división o de proyecto.
2. Generar preguntas que definan las metas anteriores de forma tan completa y cuantificable como sea posible.
3. Especificar las métricas necesarias para responder dichas preguntas y rastrear su conformidad con las metas inicialmente planteadas.
4. Desarrollar mecanismos de recolección de datos.
5. Obtener, validar y analizar los datos en tiempo real, de tal modo que se pueda obtener retroalimentación hacia los proyectos y se pueda validar la conformidad de los resultados con las metas.

La transición entre las metas y las métricas puede verse como un grafo dirigido (Figura 3), donde el flujo va de las metas a las preguntas y de las preguntas a las métricas.



**Fig. 1.** Representación del paradigma Goal/Question/Metric (GQM). **Fuente:** Adaptado de Basili [16]

La definición de metas y su refinamiento en preguntas cuantificables es un proceso complejo y requiere experiencia [13]. Para facilitar este proceso, el paradigma GQM propone una plantilla útil al momento de definir metas [16]. Esta plantilla posee cinco dimensiones: el objeto, el propósito, el enfoque, el punto de vista y el entorno. Un ejemplo de una meta definida mediante esta plantilla es: “Monitorear un proyecto de software (el *objeto*) objetivamente durante la duración del proyecto para obtener su estado (*propósito*) respecto al tamaño, complejidad, costo y calidad (enfoque) desde la perspectiva de los interesados (*punto de vista*) dado un alto índice de fallos debido a una falta de visibilidad de la gestión (*entorno*)”.

Una vez se establecen la meta y las preguntas que se desean resolver mediante la métrica, se procede a la definición de dicha métrica. A continuación se presenta una serie de 5 pasos que se deben seguir para definir una métrica:

**Paso 1. Definición de la entidad que se va a medir.** En este punto, además, se define de qué tipo de entidad se trata, si se trata de un producto, de un proceso o de recursos.

**Paso 2. Definición de la fase y el proceso del ciclo de vida para los cuales se desea definir la métrica.** Esto permite tener una mayor claridad a la hora de determinar cuándo y qué atributos se van a medir con ella. Así, dependiendo de la fase, la entidad puede ser una línea de productos, una configuración, un conjunto de componentes reutilizables o un producto ensamblado. Además, tener claridad con respecto a cuándo se va a medir la entidad da una pista sobre qué atributos se pueden medir.

**Paso 3. Definición del atributo que se va a medir.** En este paso se define el atributo que se desea medir que, como ya se definió en esta sección, es una propiedad física o abstracta medible de la entidad. Dichos atributos corresponden, a menudo, a atributos de calidad del software, pero también los hay específicos de la ILP [7].

**Paso 4. Definición del lenguaje de modelado que se va a usar en la definición de la métrica.** El lenguaje de modelado determina en gran medida el procedimiento de medición, ya que de éste dependen, por ejemplo, los elementos que se tienen en cuenta en procedimientos operativos y fórmulas matemáticas. Como se mencionó anteriormente, es muy útil saber en qué fase y proceso del ciclo de vida se usará la métrica, ya que tanto las fases como los procesos tienen uno o varios lenguajes de modelado asociados.

**Paso 5. Definición del procedimiento de medición.** Este paso incluye los aspectos operativos de la métrica, los cuales son:

1. Definir qué tipo de medida se va a tomar (directa o indirecta).
2. Definir las unidades de la medida, lo que implica determinar su escala. Esto a su vez permite saber

el tipo de escala, la cual puede ser nominal, ordinal, intervalo, ratio o absoluta.

3. Definir paso a paso el procedimiento de toma de la medida, el cual se puede expresar mediante fórmulas matemáticas, conjuntos de pasos, descripciones formales, etc.

### C. Definición de Métricas Usando el *Framework* Propuesto

Para ilustrar y demostrar la usabilidad del *framework* propuesto a pequeña escala, a continuación se define una métrica para medir el grado de reusabilidad de una característica de un *feature model* de una línea de productos. Para la definición de dicha métrica se usó el *framework* propuesto y se siguieron las pautas expuestas previamente.

En primer lugar, se define el propósito de la definición de la métrica. Esto se logra estableciendo una meta por alcanzar y una pregunta por responder con la métrica, apuntando a la meta. Para ello, se hace uso del paradigma GQM [16] explicado previamente en el numeral B.

**Meta:** Monitorear el modelado de la variabilidad de una línea de productos (el objeto) durante el análisis del dominio, que permita un diseño apropiado y válido (propósito) con respecto a la reusabilidad de características (enfoque) desde la perspectiva del ingeniero encargado del modelado del dominio (punto de vista), dada la necesidad de asegurar constantemente el diseño correcto del modelo de variabilidad (entorno).

**Pregunta:** ¿Cómo conocer la reusabilidad de una característica del modelo de variabilidad de una línea de productos?

A continuación se define la métrica usando el *framework* propuesto.

#### **Paso 1. Definición de la entidad que se va a medir.**

La entidad a medir es una característica perteneciente a un modelo de características (*feature model*) que es una entidad de tipo Producto.

#### **Paso 2. Definición de la fase y el proceso del ciclo de vida para los cuales se desea definir la métrica.**

La métrica se aplica en la fase de análisis del dominio,

específicamente en el proceso de modelado de la variabilidad.

#### **Paso 3. Definición del atributo que se va a medir.**

El atributo a medir es la reusabilidad.

**Paso 4. Definición del lenguaje de modelado que se va a usar en la definición de la métrica.** El lenguaje de modelado usado es el modelo de características (*feature model*).

#### **Paso 5. Definición del procedimiento de medición.**

El grado de reusabilidad de una característica (GRC) es la relación entre las configuraciones posibles de un modelo de variabilidad en las cuales esta característica está seleccionada, con respecto al total de configuraciones que es posible

$$GRC = \left( \frac{\#CC}{\#CT} \right) \times 100 \quad (1)$$

Donde #CC es el número de configuraciones posibles de la línea de productos en las cuales la característica está presente, y #CT es el número total de configuraciones posibles de la línea de productos. El número de configuraciones posibles con y sin la característica seleccionada se puede obtener mediante operaciones de análisis en ILP presentes en herramientas como VariaMos [35]. Mediante este procedimiento se toma una medida indirecta, cuya unidad es por ciento (porcentaje) y cuyo tipo de escala es *ratio*.

## IV. CONCLUSIONES

La medición es un proceso de gran importancia en las ciencias y la ingeniería, ya que permite una evaluación efectiva de productos, procesos y experimentos. Además, como escribió DeMarco [36]: “No se puede controlar lo que no se puede medir”. Esta importancia se extiende, de manera general, a la ingeniería de software, y de manera particular, a la ingeniería de líneas de productos. En la ingeniería de líneas de productos dicha importancia se acentúa, dado que un defecto o una falla en la calidad de una línea de productos puede, potencialmente, extenderse a diferentes productos simultáneamente.

En la literatura se pueden encontrar propuestas que apuntan a facilitar el proceso de medición en la ingeniería de software y en la ingeniería de líneas de productos. Desde marcos conceptuales amplios y rigurosos hasta *frameworks* o marcos de trabajo para la medición de aspectos específicos de las líneas de productos de software, pasando por estándares internacionales de medición y por marcos conceptuales para la medición en la ingeniería de software. Sin embargo, cada una de las propuestas encontradas presenta diferentes limitantes en la definición de métricas en ingeniería de líneas de productos.

Teniendo en cuenta las limitantes encontradas en propuestas anteriores, en el presente artículo se propone un *framework* conceptual para la definición de métricas en ingeniería de líneas de productos. Este *framework* comprende un conjunto de conceptos que se deben tener en cuenta al momento de definir una métrica enmarcada en el ciclo de vida de una línea de productos, así como las relaciones existentes entre dichos conceptos. Además, se propone un conjunto de pautas que se deben seguir en la definición de las métricas, y así facilitar la implementación del *framework* en el nivel operativo.

En la presente propuesta se suplen las limitantes de trabajos previos. No solo se propone una terminología claramente definida para el proceso de medición, sino que también se complementa con pautas y pasos para seguir en la definición de las métricas. Además, el *framework* propuesto incluye conceptos generales del área de la medición y conceptos particulares de la ingeniería de líneas de productos, de tal modo que no se pasen por alto a la hora de definir métricas que se van a usar en el ciclo de vida de una línea de productos.

Tomando como base el trabajo expuesto en el presente artículo, se plantean trabajos futuros. En primer lugar, es pertinente la definición de una ontología para la medición en la ingeniería de líneas de productos. Un trabajo de este tipo puede contribuir a reforzar el *framework* propuesto, apuntando no solo a la formalización de su enfoque, sino también a una aplicación más precisa y sin ambigüedades del proceso de medición. En segundo lugar, se encuentran pertinentes trabajos dirigidos a la automatización del proceso de medición. Este aspecto es importante,

ya que la automatización de una métrica facilita su uso en diferentes entornos, tanto académicos como industriales.

## AGRADECIMIENTOS

Este trabajo hace parte de los resultados del proyecto “Desarrollo de soluciones para soportar la completitud y la corrección de líneas de productos con aplicación a la ingeniería de software”, financiado por el Departamento Administrativo de Ciencia, Tecnología e Innovación (Colciencias).

## REFERENCIAS

- [1] K. Pohl, G. Böckle y F. Van Der Linden, Software Product Line Engineering. Foundations, Principles, and Techniques, vol. 49, N.º 12, 2005.
- [2] J. Bosch, Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach, 2000.
- [3] S. Apel, D. Batory, C. Kästner y G. Saake, Feature-Oriented Software Product Lines: Concepts and Implementation, 2013.
- [4] K. Czarnecki y U. W. Eisenecker, “Generative programming: methods, tools, and applications”, en Object-Oriented Technology ECOOP 2002 Workshop Reader, ACM Press/Addison-Wesley Publishing Co., 2000.
- [5] K. C. Kang, S. G. Cohen, J. a Hess, W. E. Novak y S. Peterson, “Feature-Oriented Domain Analysis (FODA) Feasibility Study,” Distribution, N.º 17, p. 161, nov.1990.
- [6] R. Mazo, C. Salinesi, D. Diaz, O. Djebbi y A. Michiels, “Constraints: the Heart of Domain and Application Engineering in the Product Lines Engineering Strategy,” Int. J. Inf. Syst. Model. Des. IJISMD, vol. 3, N.º 2, pp. 33-68, 2011.
- [7] S. Montagud, S. Abrahão y E. Insfran, “A systematic review of quality attributes and measures for software product lines”, Softw. Qual. J., vol. 20, N.º 3-4, pp. 425-486, 2012.
- [8] F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini y M. Genero, “Towards a consistent terminology for software measurement”, Inf. Softw. Technol., vol. 48, N.º 8, pp. 631-644, 2006.
- [9] N. E. Fenton y S. L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, vol. 2, 1997.
- [10] C. Ebert, “Correspondence visualization techniques for analyzing and evaluating software measures”, IEEE Trans. Softw. Eng., 1992.

- [11] B. Kitchenham, S. L. Pfleeger y N. Fenton, "Towards a framework for software measurement validation", *IEEE Trans. Softw. Eng.*, vol. 21, N.º 12, 1995.
- [12] L. Briand, K. El Emam y S. Morasca, "On the application of measurement theory in software engineering", *Empir. Softw. Eng.*, vol. 1, N.º 1, pp. 61-88, 1996.
- [13] IEEE Computer Society, "IEEE Standard for a Software Quality Metrics Methodology - IEEE Std 1061TM-1998 (R2009)", vol. 1998, 2009.
- [14] International Organization for Standardization (ISO), ISO/IEC 15939, *Systems and Software Engineering - Measurement Process*, vol. 2007, January 2007.
- [15] International Organization for Standardization (ISO), ISO/IEC 14598-1, "Information Technology - Software Product Evaluation", 1999.
- [16] V. R. Basili, "Software modeling and measurement: the Goal/Question/Metric paradigm", *Quality*, p. 24, 1992.
- [17] H. Zuse, *A Framework of Software Measurement*, Walter de Gruyter & Co., 1997.
- [18] J. C. F. G. I. M. Jcgm, "JCGM 200 : 2008 International vocabulary of metrology — Basic and general concepts and associated terms (IVM), Vocabulaire international de métrologie — Concepts fondamentaux et généraux et termes associés (VIM)", *Int. Organ. Stand. Geneva ISBN*, vol. 3, N.º Vim, p. 104, 2008.
- [19] S. Puroo y V. Vaishnavi, "Product metrics for object-oriented systems," *ACM Comput. Surv.*, vol. 35, N.º 2, pp. 191-221, 2003.
- [20] N. Habra, A. Abran, M. Lopez y A. Sellami, "A framework for the design and verification of software measurement methods", *J. Syst. Softw.*, vol. 81, N.º 5, pp. 633-648, 2008.
- [21] C. Sant'Anna, E. Figueiredo, A. Garcia y C. J. P. Lucena, "On the Modularity of Software Architectures: A Concern-Driven Measurement Framework", *Softw. Archit.*, N.º 4758, pp. 207-224, 2007.
- [22] J. S. Her, J. H. Kim, S. H. Oh, S. Y. Rhew y S. D. Kim, "A framework for evaluating reusability of core asset in product line engineering," *Inf. Softw. Technol.*, vol. 49, N.º 7, pp. 740-760, 2007.
- [23] J. J. Ding, "Software product line measurement process capability maturity model", *Appl. Mech. Mater.*, vol. 536-537, N.º 2014, pp. 673-677, 2014.
- [24] H. Venkitachalam, C. Granrath, B. Gopalakrishnan y J. Richenhagen, "Metric-based Evaluation of Powertrain Software Architecture", *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 10, N.º 1, pp. 2017-01-1615, marzo 2017.
- [25] R. J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*, Berlín, Heidelberg: Springer Berlin Heidelberg, 2014.
- [26] K. Peffers, T. Tuunanen, M. A. Rothenberger y A. S. Chatterjee, "A Design Science Research Methodology for Information Systems Research", *J. Manag. Inf. Syst.*, vol. 24, N.º 3, pp. 45-77, 2008.
- [27] R. Smyth, "Exploring the Usefulness of a Conceptual Framework as a Research Tool: A Researcher's Reflections", *Issues Educ. Res.*, vol. 14, N.º 2, pp. 167-180, nov. 2003.
- [28] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell y A. Wesslén, *Experimentation in Software Engineering*, Berlín, Heidelberg: Springer Berlin Heidelberg, 2012.
- [29] T. von der Maßen and H. Lichter, "Determining the Variation Degree of Feature Models", en *Software Product Lines*, 2005, pp. 82-88.
- [30] A. Fuggetta, L. Lavazza, S. Morasca, S. Cinti, G. Oldano y E. Orazi, "Applying GQM in an industrial software factory", *ACM Trans. Softw. Eng. Methodol.*, vol. 7, N.º 4, pp. 411-448, 1998.
- [31] R. van Solingen y E. Berghout, "Integrating goal-oriented measurement in industrial software engineering: industrial experiences with and additions to the Goal/Question/Metric method (GQM)", en *Proceedings Seventh International Software Metrics Symposium*, 2001, pp. 246-258.
- [32] P. Berander y P. Jönsson, "A goal question metric based approach for efficient measurement framework definition", en *Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06*, 2006, p. 316.
- [33] M. G. Mendonça y V. R. Basili, "Validation of an approach for improving existing measurement frameworks", *IEEE Trans. Softw. Eng.*, vol. 26, N.º 6, pp. 484-499, 2000.
- [34] L. Briand, S. Morasca y V. Basili, "An operational process for goal-driven definition of measures", *IEEE Trans. Softw. Eng.*, vol. 28, N.º 12, pp. 1106-1125, 2002.
- [35] R. Mazo, J. Muñoz-Fernández, L. Rincón, C. Salinesi y G. Tamura, "VariaMos: an extensible tool for engineering (dynamic) product lines", *19th Int. Softw. Prod. Line Conf.*, 2015, pp. 1-6.
- [36] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates*, Prentice Hall PTR, 1986.

## Capítulo II

# ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE: ACERCAMIENTO AL MARCO TEÓRICO Y AL ESTADO DE LA CUESTIÓN DESDE LA DISCIPLINA DE LA INGENIERÍA DE SOFTWARE

*Luis Eduardo Peláez Valencia - luiseduardo.pelaez@gmail.com*

Docente e investigador, Universidad Católica de Pereira, Pereira

*Sandra Victoria Hurtado Gil - sandra.hurtado@ucaldas.edu.co*

Docente Universidad de Caldas, Manizales

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

Docente e investigador, Universidad Católica de Pereira, Pereira

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

Docente e investigador, Universidad Católica de Pereira, Pereira

## I. INTRODUCCIÓN

El Eje Cafetero, en Colombia, es una región geográfica y cultural que comprende los departamentos de Caldas, Quindío y Risaralda, el norte del Valle del Cauca y el noroccidente del Tolima. Esta región se caracteriza por el desarrollo común en sus dimensiones económica, ecológica, política y social.

Su denominación de Eje Cafetero se debe precisamente a que su economía se soporta en el café, cuya importancia lo hizo trascender más allá de un fruto o un cultivo hasta generar la llamada Cultura Cafetera que llegó a ser la base de su desarrollo social y económico por muchos años. Sin embargo, se acercaría el momento para resignificar la manera de seguir desarrollando la región.

En los años ochenta, el vertiginoso declive de la industria cafetera en la región y en el país, dejó a la primera relegada a un nivel secundario de importancia en la economía local, y aunque aún representa un importante renglón en la economía nacional, no es ni sombra de lo que algún día representó para este país. Lo que sí es claro en el subconsciente colectivo de los habitantes de la Región Cafetera, es que el café ha significado prosperidad, riqueza y desarrollo para todos.

En la actualidad otros sectores emergentes, entre ellos las tecnologías de la información, comenzaron a proponer alternativas productivas y de calidad que ayudaran a consolidar una industria basada en el conocimiento como soporte al desarrollo económico y social de la región [1], uno de esos sectores ha sido la Industria del Software.

En este sentido, un grupo de investigadores en la disciplina de la Ingeniería del Software radicados en la Región Cafetera han adelantado proyectos de investigación alrededor del aseguramiento de la calidad del software (SQA, por sus siglas en inglés), con el propósito de entregar a la Industria elementos que permitan contribuir al desarrollo económico y social a través de la construcción de productos software de calidad.

Para el caso que nos ocupa, se presentaron los acercamientos teóricos, conceptuales y de diagnóstico para fundamentar el proyecto de investigación, que busca problematizar alrededor de un modelo para el aseguramiento de la calidad de los requerimientos en proyectos de software, radicado y aprobado en la convocatoria de 2016 de la Universidad Católica de Pereira.



El artículo se estructura de la siguiente forma: en la sesión 2 se presenta un acercamiento al marco teórico que soporta conceptualmente la calidad del software; en la sesión 3, el estado de la cuestión con elementos del orden internacional, regional y local; en la sesión 4, la caracterización de la industria en la ciudad de Pereira; en la sesión 5, las conclusiones y, finalmente, las referencias.

## II. MARCO TEÓRICO

Actualmente nos encontramos en lo que varios expertos llaman la sociedad o la era del conocimiento, donde “la difusión de las nuevas tecnologías y la aparición de la red pública Internet parecen abrir nuevas perspectivas a la ampliación del espacio público del conocimiento” [2]. Aunque existe cierta discusión entre los términos sociedad del conocimiento y sociedad de la información [3], es innegable que las transformaciones que han llevado a los expertos a hablar de estas nuevas sociedades se basan en las tecnologías de la información y las comunicaciones (TIC).

Uno de los elementos más importantes en las TIC es el software, el corazón de los miles de dispositivos y redes que existen en la actualidad, y que incluso se encarga de crear y controlar otro software [4].

Considerando la gran importancia del software en la actualidad, se vuelve de especial interés para quienes hacen parte de esta industria, tener en cuenta altos criterios de calidad que garanticen su correcto funcionamiento y comportamiento en las diferentes circunstancias en la cuales es utilizado.

La preocupación por desarrollar software de alta calidad no es nueva, pues desde los años sesenta se encontró que “producir buen software es muy difícil, muy costoso y muy necesario” [5]. Es entonces como en el año 1968 se acuñó el término Ingeniería de Software, como una meta a la que se deseaba llegar: “Aplicar la ingeniería al software” [6].

La Ingeniería de Software ha tenido grandes avances desde el nacimiento del término, e incluso cuenta con una definición estándar: “Aplicación sistemática de conocimiento científico y tecnológico, métodos y experiencia al diseño, implementación,

pruebas y documentación del software para optimizar su producción, soporte y calidad” [7]. No obstante, vale la pena mostrar otra definición que enfatiza la importancia de esta disciplina para la sociedad: “La rama de las ciencias de la computación que crea soluciones rentables a problemas prácticos de computación mediante la aplicación de conocimiento científico al desarrollo de sistemas de software al servicio de la humanidad” [8].

En todos los casos, la ingeniería de software se fundamenta en el compromiso con la calidad [4], pero ¿qué se entiende por calidad (en general) y por calidad de software?

Aunque el concepto de calidad puede ser subjetivo [9], existen varias definiciones de calidad, entre las cuales se pueden destacar: “La calidad es el cumplimiento de los requisitos” [10]. “Calidad es adecuación al uso del cliente” [11]. “Grado en el que un conjunto de características inherentes de un objeto cumple con los requisitos” [12]. En estas tres definiciones se ve cómo el concepto está directamente relacionado con la idea de los requisitos y el uso que el cliente le dará a un producto o servicio.

El concepto de *calidad de software* no está muy lejano del concepto de calidad en general, pero establece un elemento adicional de gran importancia: las necesidades del usuario. Esto se puede ver en dos definiciones de calidad de software tomadas de estándares internacionales: “Grado en el cual un producto de software satisface las necesidades establecidas e implícitas, cuando es utilizado en condiciones específicas” [13], y la otra es: “Grado en que un sistema, componente o proceso cumple las necesidades o deseos de clientes y usuarios” [14].

Es decir que la calidad en el software va un poco más allá de los requisitos, y debe apuntar a solucionar las necesidades reales de los usuarios, para contribuir así a mejorar su bienestar o a lograr sus objetivos. Esto está totalmente alineado con las tendencias actuales del desarrollo de software, en las que el software es visto como un componente estratégico para los negocios, que apoya o incluso amplía sus metas.

Los ingenieros de software tienen una gran responsabilidad en los proyectos en los cuales participan, pues deben contribuir a desarrollar y entregar productos de software de alta calidad [15]. Es aquí donde la ingeniería de software participa con todo el conocimiento que ha construido desde su nacimiento, y que se lleva a la práctica principalmente mediante técnicas, métodos, procesos y herramientas.

La mayoría de los autores coinciden en los elementos de la ingeniería de software relacionados con la calidad, pero difieren en la forma de agruparlos. Por ejemplo, en la guía al cuerpo de conocimiento de la ingeniería de software [6], se identifican los siguientes temas en el área de conocimiento Calidad de Software:

**Fundamentos de la calidad de software:** Incluye aspectos como cultura de la calidad, costos, atributos de calidad y mejora de la calidad de software.

**Procesos de gestión la calidad de software:** Incluye aseguramiento de la calidad del software, validación y verificación y revisiones y auditorías.

**Consideraciones prácticas y herramientas:** Incluye requisitos de calidad de software, caracterización de defectos y técnicas de gestión y métricas de calidad de software.

Un reconocido autor [4] menciona cuatro actividades principales:

**Métodos de la ingeniería de software:** Incluye actividades de requisitos, análisis y diseño.

**Técnicas de administración de proyectos:** Incluye la planeación y seguimiento de la calidad en cada proyecto.

**Control de calidad:** Acciones que ayudan a asegurar que los productos cumplan con las metas de calidad.

**Aseguramiento de la calidad:** Infraestructura general que soporta las diferentes acciones y funciones de calidad.

Otros autores relacionados con el tema [16] establecen los siguientes ámbitos:

**Calidad de realización:** Mejora de las capacidades de las personas. Implica conocer técnicas y competencias de desarrollo y mantenimiento.

**Organización de las actividades a nivel de proyecto:** Contempla la planeación y el seguimiento de actividades de calidad en proyectos de desarrollo y mantenimiento. Incluye actividades de: Aseguramiento de la calidad y Evaluación y control de calidad.

**Organización de las actividades a nivel de empresa:** Contempla acciones de nivel organizacional, que incluyen la mejora de procesos.

**Correcto análisis de necesidades del usuario y buena especificación de requisitos.**

En todos los casos es posible observar que no existe solo un aspecto que logre o garantice la calidad, sino que hay una amplia gama de actividades y elementos que deben tenerse en cuenta desde la ingeniería de software para lograr productos de calidad.

Todas las actividades presentadas son relevantes, pero hay una que se menciona con frecuencia y por ello amerita una definición concreta, se trata del *aseguramiento de la calidad del software* (SQA, por sus siglas en inglés): “Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad” [17].

Este concepto se resalta porque su introducción en el mundo del software ha permitido hacer énfasis en que la calidad no es algo que se logre al final del proceso [18], sino que requiere acciones concretas desde el comienzo y durante todo el desarrollo de cada proyecto, es decir que abarca estándares, administración de configuración, auditorías, revisiones, pruebas, capacitación, administración de riesgos, procesos de requisitos, procesos de diseño, etc.

En otra referencia, donde se resalta constantemente la integración de actividades de calidad en el desarrollo, se dice que el aseguramiento de la calidad

da soporte a la entrega de productos de alta calidad, proporcionando al personal del proyecto y a los gerentes, en todos los niveles, la visibilidad apropiada y la realimentación sobre los procesos y los productos de trabajo asociados, durante toda la vida del proyecto. [19].

El aseguramiento de la calidad del software tiene dos aspectos o áreas: 1) Aseguramiento del producto, que busca asegurar que éste cumple con los requisitos establecidos, y 2) aseguramiento del proceso, que busca asegurar que los procesos son efectivos y se aplican correctamente en cada proyecto [18].

Estas dos áreas de SQA están alineadas con dos de los componentes de la ingeniería de software: el producto y el proceso. Los otros dos componentes son las personas y el proyecto [20], los cuales tienen una innegable influencia en la calidad del software, pero las actividades que tienen que ver con estos dos componentes por lo general están inmersas en las áreas de aseguramiento del producto y del proceso.

A continuación se presenta un breve recuento de los aspectos de calidad que hacen parte de la ingeniería de software, relacionados con el producto y con el proceso:

### A. Calidad del Producto

Cuando se habla de producto software no se hace referencia solo al código, sino también a su documentación y a todos los productos intermedios asociados, como los requisitos, el diseño, las pruebas, etc.

La calidad del producto, en este caso el software, está estrechamente relacionada con los requisitos, y por lo tanto la satisfacción del cliente puede considerarse un indicador de la calidad. Sin embargo, no es conveniente dejar para el final del proceso la evaluación de la calidad, pues la corrección de defectos en etapas tardías de desarrollo, o con el software en producción, es altamente costosa [21]. Por este motivo se cuenta con diferentes estrategias para evaluar y mejorar la calidad del producto de software, y las más conocidas son las pruebas, los modelos de calidad y las métricas.

**Pruebas:** Es interesante señalar que el concepto de prueba ha evolucionado mucho en los últimos años, ha pasado de un concepto destructivo a uno constructivo, y que además está presente desde el comienzo del desarrollo. Una de las definiciones más recientes de pruebas es:

Proceso conformado por todas las actividades del ciclo de vida, tanto estáticas como dinámicas, relacionadas con la planeación, preparación y evaluación de los productos de software y relacionados, para determinar que satisfagan los requisitos especificados, para demostrar que cumplen su propósito y para detectar defectos. [22].

**Modelos de calidad:** Teniendo en cuenta que la calidad debe verse desde un punto de vista multidimensional [23], se consideró que una forma de evaluar la calidad en el software era mediante diferentes características o propiedades que individualmente fueran más fáciles de medir [16]. Esta descomposición en factores, atributos o características de calidad se conoce como modelos de calidad. Algunos modelos de calidad ampliamente conocidos son: el modelo de MacCall [24], el modelo de Bohem [25] y el estándar ISO/IEC 25010:2011 [13].

También existe la posibilidad de que no se utilice un modelo estándar de calidad, sino que se definan para cada proyecto los criterios que determinarán la calidad del producto. En estos casos se cuenta con referencias que dan lineamientos para la selección y definición de dichos criterios y sus correspondientes métricas, como el manual de medición orientada a metas [26] y el enfoque GQM [27].

**Métricas de producto:** Las métricas son una forma de obtener valores cuantificables para diferentes aspectos de un producto. En el caso de la calidad del producto, por lo general las métricas están relacionadas con los criterios de calidad definidos en un modelo de calidad. Al dar valores concretos a cada criterio, es posible evaluar de manera un poco más objetiva la calidad del producto, y si es del caso tomar acciones correctivas o de mejora.

La aplicación de métricas es un área que tiene cierta controversia, sobre todo después de la amplia difusión



de los métodos ágiles de desarrollo, pues se cuestiona el verdadero valor que pueden aportar en relación con el esfuerzo que demanda obtenerlas. En general se ha identificado que es adecuado usar métricas, pero éstas deben ser realmente efectivas y se debe capacitar a las personas sobre su correcto uso [28].

## B. Calidad del Proceso

Una premisa importante que se derivó de los sistemas de calidad en empresas industriales es que “la calidad de un sistema o producto está altamente influenciada por la calidad del proceso utilizado para desarrollarlo y mantenerlo” [18]. Lo anterior se complementa indicando que “es esencial identificar métodos y aproximaciones para estudiar la madurez de un proceso, e identificar estrategias y procedimientos para mejorarlo” [29].

La ingeniería de software no es ajena a estas premisas, y también aporta elementos para contribuir a la calidad de los procesos, entre los cuales se encuentran: modelos de referencia y de evaluación de procesos, modelos de mejora y métodos ágiles.

### Modelos de referencia y evaluación de procesos:

En este grupo se incluyen estándares, normas y modelos que establecen un conjunto de elementos básicos con que deben contar los procesos de desarrollo de software para considerar que tienen buena calidad. Este tipo de modelos son utilizados por las empresas para revisar sus procesos y determinar cuáles son objeto de mejora, o para contar con una certificación que muestre que cumplen con el referente seleccionado.

Existe una gran cantidad de modelos de referencia y evaluación de procesos, con diferentes niveles de detalle y áreas de aplicación, que les permiten a las empresas poder seleccionar aquel que sea más adecuado a sus necesidades. Estos modelos definen el “qué” debe tener el proceso, más no el “cómo”, que puede ser diferente de una empresa a otra. Aunque algunos modelos incluyen elementos del “cómo”, lo presentan siempre como algo opcional o complementario, y no es obligatorio, por ejemplo, para una posible certificación. Algunos de los modelos más conocidos son:

**ISO 9000 – Gestión de la calidad:** Es una familia de estándares relacionados con varios aspectos de la gestión de la calidad. En particular, el estándar ISO 9001:2015 es certificable. Para obtener esta certificación en el área de ingeniería de software se cuenta con la guía ISO/IEC 90003:2014.

### CMMI - Capability Maturity Model®

**Integration:** “Son colecciones de buenas prácticas que ayudan a las empresas a mejorar sus procesos” [18]. Este modelo está conformado por “áreas de proceso, objetivos o metas, prácticas y material informativo” [30].

El CMMI fue creado por el Instituto de Ingeniería de Software (SEI), permite evaluar la madurez o la capacidad de los procesos que tenga una organización de software. En cuanto a la madurez, se definen cinco niveles: El nivel 1 o inicial, donde el proceso es impredecible; el nivel 2 o gestionado, donde el proceso es aplicado en proyectos; el nivel 3 o definido, donde el proceso es aplicable a toda la organización; el nivel 4 o gestionado cuantitativamente, donde el proceso es predecible, y el nivel 5 o en optimización, donde se hace énfasis en la mejora continua [31]. Cada nivel de madurez incluye un conjunto de áreas de proceso, de manera que la organización va mejorando paulatinamente sus procesos al cumplir cada vez con los correspondientes objetivos y prácticas.

La versión 1.3 del modelo CMMI-DEV (para desarrollo de software) incluye 22 áreas de proceso. Las áreas que tienen más relación con los conceptos que se han venido presentando en este marco teórico, son: Medición y Análisis (MA), Aseguramiento de la Calidad del Proceso y del Producto (PPQA), Desarrollo de Requisitos (RD), Validación (VAL) y Verificación (VER).

**Otros modelos:** Modelos como ISO 9000 y CMMI tienen una amplia trayectoria y son reconocidos internacionalmente, pero la inversión que representa su adopción hace que las pequeñas empresas tengan reservas para implementarlos [32]. Varios estudios muestran que las pequeñas empresas tienen una percepción negativa de estos modelos, basada en la idea de que son costosos y burocráticos [33].

Por eso se han creado modelos específicamente para pequeñas empresas y organizaciones, que por lo general cuentan con menos procesos y, además, están acompañados de guías de implementación. En Latinoamérica se destacan Moprosoft [34] y MPS.BR [35].

**Modelos de mejora:** El mundo del software se encuentra en constante evolución y los procesos de desarrollo no son ajenos a esta tendencia, por lo que es necesario revisarlos y ajustarlos constantemente, o dicho de otra forma: “La mejora del proceso es un mecanismo de mejora continua de la calidad” [16].

Realizar cambios en los procesos no es una tarea fácil, sobre todo porque se está transitando hacia aspectos nuevos y muchas veces desconocidos en la forma de realizar las tareas. Por este motivo se presentan varios modelos tomados, la mayoría, de disciplinas administrativas, que establecen un conjunto de lineamientos y tareas para llevar a cabo un proyecto de mejora. Entre los modelos más reconocidos se encuentran: modelo Planear-Hacer-Verificar-Actuar [6], el modelo IDEAL [36] y los modelos de Six Sigma [37].

**Métodos ágiles:** Como se comentó previamente, existen otros factores, además del proceso, que afectan la calidad del software. Tener en cuenta los aspectos sociales, es decir, a las personas, como elemento central en el desarrollo de software, fue la idea que impulsó la creación del manifiesto ágil, que a su vez incentivó la aplicación de métodos ágiles [29].

A diferencia de los procesos prescriptivos, los métodos ágiles se basan más en principios, valores y técnicas de coordinación y colaboración de equipos de personas, que en la definición de actividades y tareas que se deben llevar a cabo. Desde este punto de vista, se pueden considerar procesos empíricos que son diferentes para cada equipo y tienen suficiente flexibilidad para ajustarse a los continuos cambios.

Los métodos ágiles se caracterizan por ciclos cortos de desarrollo iterativos, equipos autoorganizados, participación activa de los clientes y énfasis en crear productos de trabajo funcionales de manera frecuente para obtener realimentación temprana [6].

Se mencionan a continuación algunos métodos ágiles, solo a manera de ejemplo, dada la gran variedad que existe: *Adaptative Software Development (ASD)*, *Agile Modeling (AM)*, *Crystal Methods*, *Extreme Programming (XP)*, *Feature-Driven Development (FDD)*, *Lean Development*, y *Scrum*.

### C. Ingeniería de Requisitos

Al comienzo de este marco teórico se señaló la importancia de los requisitos y necesidades del usuario para obtener productos de calidad. Se puede afirmar que no es posible hablar de la calidad de un producto, ni medirla, si no se tienen claramente establecidos los requisitos del producto. De allí que se pueda decir que “un buen producto no requiere solamente conformidad a la especificación sino una buena especificación en primer lugar” [16].

Considerando la relevancia de los requisitos en los procesos de software, se acuñó el término Ingeniería de requisitos, de manera similar a como nació el término Ingeniería de software, como una meta hacia la cual se desea llegar. Algunas definiciones de ingeniería de Requisitos son:

“Función interdisciplinaria que media entre los dominios del adquirente y el proveedor para establecer y mantener los requisitos que debe cumplir el sistema, software o servicio de interés” [38]. “Proceso que comprende todas las actividades requeridas para crear y mantener un documento de requerimientos del sistema” [39].

La ingeniería de requisitos tiene dos grandes subdisciplinas que son: el desarrollo de requisitos y la gestión de requisitos; a su vez, el desarrollo de requisitos se subdivide en las siguientes actividades: Obtención (también llamada adquisición, educación o “elicitación”), Análisis, Especificación y Validación [40].

Si se considera que es mucho más económico encontrar y corregir defectos desde requisitos [41], cobra importancia el aseguramiento de la calidad en la ingeniería de requisitos. Mediante procesos de aseguramiento de la calidad se busca verificar que los requisitos cumplan con unas características básicas de calidad: necesario, factible, no ambiguo, consistente, completo, correcto, rastreable y verificable [38].

Así como que el aseguramiento de la calidad en software está presente en todas las etapas de un desarrollo, en los requisitos el aseguramiento de la calidad está presente en todas las actividades, no solo en la validación.

Por lo anterior, es importante definir una estrategia de calidad que tenga en cuenta el contexto organizacional, los recursos disponibles, los riesgos y el nivel de calidad deseado en los requisitos. Esta estrategia deberá tener en cuenta diferentes técnicas para lograr el objetivo deseado. Se deben considerar: técnicas de obtención, técnicas de especificación y técnicas de validación y verificación (estáticas y dinámicas) [42].

### III. ESTADO DE LA CUESTIÓN

En investigaciones adelantadas con el propósito de conocer la manera como este Marco teórico ha sido conocido y apropiado en la industria, se logra evidenciar la diferencia entre la preocupación de los autores por aportar desde la literatura y la realidad del sector [1], [43], [44].

#### A. Internacional

La literatura existente en el área provee una gran variedad de definiciones de Ingeniería de software, Calidad de software, Aseguramiento de la calidad del software, pruebas, modelos de calidad, etc. Todas ellas conducentes a reconocer la importancia de definir el problema que se pretende resolver con el desarrollo de un producto software y lo atinado de contar inicialmente con un buen proceso de aseguramiento de la calidad en el desarrollo y la gestión de los requisitos [40]. Sin embargo, como parte del estado de la cuestión, también se encuentran investigaciones relevantes con respecto al campo de los requerimientos de software y que hacen alusión a la manera como la industria del software de los países, incluso desarrollados, cuenta con un sinnúmero de proyectos adelantados y fracasados, la mayoría de ellos por problemas relacionados con: la captura, el análisis y la especificación de requisitos, así como con la identificación de los stakeholders o interesados, las técnicas y herramientas utilizadas en el proceso y el alto nivel de desconocimiento de modelos y métricas

de calidad que acompañan un proceso de desarrollo adecuado y satisfactorio [45], [46]. Lo anterior lo confirma el Informe del Caos [47] que se publica anualmente para conocer el comportamiento de los proyectos.

La Tabla 1 permite visualizar el impacto de los requerimientos frente al nivel de éxito, fracaso o transformación de los proyectos; cerca del 25 % de los proyectos se han impactado por tener incompletos los requerimientos o por cambios en ellos.

**Tabla 1.** Impacto de los Requerimientos con respecto al Fracaso de los Proyectos [48]

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

El acelerado avance en las tecnologías que soportan el desarrollo de proyectos en la industria del software ha incidido en la calidad de los mismos desarrollos y la manera como se van liberando los productos resultantes. Sin embargo, a pesar de los crecientes aportes del software al mundo actual, durante su desarrollo continuamente se incurre en el mismo tipo de problemas que generan retrasos sistemáticos en los plazos de entrega, excesos en presupuesto, entregas con una alta tasa de errores y utilidad inferior a la esperada.

En gran medida, esta problemática es atribuible a defectos en los procesos utilizados para recoger, documentar, acordar y modificar los requisitos del sistema. Los requisitos son los cimientos sobre los cuales se construye un producto software, sin embargo, la incapacidad de gestionar sus cambios es una de las principales causas por las que un producto software se entrega fuera de tiempo, se excede en costo y no cumple con la calidad esperada por el cliente [48].

En Latinoamérica, por ejemplo, la norma MoproSoft estuvo a la vanguardia en el desarrollo de modelos de procesos. Esta norma tiene la particularidad de adaptarse a la realidad de las empresas constructoras de software de la región. En general, estas empresas tienen características diferenciadas con las empresas de los países desarrollados, básicamente en tres grandes aspectos: volumen de negocios de la organización, cantidad de personal involucrado en el proceso software y costos económicos asociados. De esta forma, la implantación de normas como CMMI o ISO 9000 resulta demasiado onerosa y complicada para la organización. Así, se reconoce un gran interés de las organizaciones a nivel internacional de soportar con literatura y con modelos el aseguramiento de la calidad del software, pero, por otro lado, la realidad de la industria imposibilita su implementación [44].

## B. Regional y Local

En Colombia, particularmente en la ciudad de Pereira, se reconocen los esfuerzos de la academia por investigar sobre la calidad del software con marcada preocupación e intención de acercarse más a la industria, lo que genera que se trascienda del mero desarrollo de actividades o artefactos al reconocimiento del marco teórico y el estado de la cuestión, con el fin de estimular la adopción de buenas prácticas que lleven a la misma industria local a ser más competitiva en el orden nacional e internacional [1], [43], [44], [49].

La industria de software de Colombia está dominada por microempresas y pequeñas empresas que se dedican especialmente al desarrollo de software a medida, a la intermediación entre las multinacionales y los clientes finales, a la compra-venta de equipos y a la oferta de diferentes servicios relacionados con el sector de Tecnologías de la Información. El sector de software cuenta hoy con cuatro grupos de empresas bien definidas: empresas desarrolladoras de software, empresas distribuidoras y comercializadoras de productos informáticos, empresas proveedoras de acceso y servicios de Internet y empresas productoras de hardware [50].

Las empresas netamente colombianas son reconocidas como “artesanas del software”, pues se

han concentrado en el desarrollo de software a medida, es decir, adecuado especialmente a las necesidades del cliente. Por su parte, la industria local en Pereira no ha sido ajena a este fenómeno y no ha logrado incursionar en el desarrollo de software empaquetado, a causa de la atención inmediata que las multinacionales instaladas en la región ofrecen al mercado.

Para esta investigación, la población de interés son las empresas productoras de software que estén categorizadas como pequeñas y medianas (MiPymes). El muestreo utilizado para la investigación se basa en muestreo no probabilístico [51]. Para el caso de la muestra de empresas desarrolladoras de software en el mundo y en el país se optó por un muestreo por juicio [51], el cual consistió en revisar las páginas web de las empresas más importantes del sector y de allí extraer la información pertinente, para el caso de las empresas de nivel mundial se seleccionaron 22, y para las del nivel nacional, 28. Los resultados se encuentran en la Tabla 2.

**Tabla 2.** Uso de Metodologías versus Certificaciones

	Nacional	Internacional
Total empresas	28	22
N.º de empresas que usan metodologías	27	21
Evaluación/Certificación/Reconocimiento	25	21

Una vez analizada la información, se encuentra que tanto en el orden internacional como en el nacional, hay una correspondencia entre el uso de metodologías y la obtención de certificaciones o reconocimientos, lo cual refiere una cultura de calidad asociada al proceso de desarrollo. El 95,5 % de las empresas analizadas fuera de Colombia que usan algún tipo de metodología de desarrollo, también han obtenido certificaciones o reconocimientos; mientras que para las empresas locales el porcentaje es del 89,3 %.

Con respecto a los parques tecnológicos, Parquesoft es el conglomerado de ciencia y tecnología más grande del país; consiste en un innovador modelo de asociación que busca consolidar un corredor de desarrollo de ciencia y tecnología informática. En la actualidad, esta entidad sin fin de lucro, “cuyo

propósito es facilitar la creación y desarrollo de empresas que provean al mercado de productos y servicios de tecnología informática”, ha logrado, mediante el apoyo a proyectos de emprendimiento, integrar sedes en las ciudades de Cali, Popayán, Pasto, Buga, Tuluá, Palmira, Armenia, Manizales, Pereira y Sincelejo, y está comenzando su expansión hacia la costa Atlántica del país.

Los requisitos de software representan una parte fundamental en la consecución de la calidad. Sin embargo, se detecta en la industria local, investigado particularmente en la ciudad de Pereira, que en lo relacionado con el tratamiento que se les hace a los requisitos, considerándolos como un punto clave para lograr productos y procesos de calidad, de 59 Mipymes consultadas sobre su metodología para el desarrollo de proyectos de software, 28 manifestaron no tener como punto de partida un proceso o una fase relacionados con la definición, especificación y desarrollo de requerimientos o requisitos [52].

Cuesta [1] considera al respecto, que si la región le quiere apostar seriamente al desarrollo soportado transversalmente en las TIC como fundamento principal para su ejecución y desarrollo, de tal manera que se pueda favorecer a la industria del software basada en el conocimiento, entonces las apuestas regionales deberán hacer especial énfasis en el mejoramiento de la calidad del software a través de la perfección de sus prácticas de desarrollo o la adaptación de nuevos modelos para la industria del software de Mipymes.

Aun cuando se está trabajando para el fortalecimiento de la agremiación de las empresas de software, la industria de software en Colombia se encuentra desarticulada, falta camino por recorrer. La desarticulación no sólo está presente entre las empresas locales sino entre el Estado y las federaciones de software y entre éstas y las empresas. Existen principalmente dos federaciones: una es Business Software Alliance (BSA), que tiene fuertes nexos con las compañías internacionales y que concentra su trabajo en la lucha contra la piratería, y la segunda es la Federación Colombiana de la Industria de Software (Fedesoft), que representa principalmente a las pequeñas empresas locales de software [50].

Todo lo anterior ha llevado a la industria del software local a relegarse de las tendencias mundiales en cuanto a la manera de desarrollar productos de calidad, y aunque en determinados sectores la fabricación de software a medida permite altos niveles de satisfacción del usuario, su desempeño no es comparable con el de empresas internacionales y, por tanto, pierden competitividad en los mercados local y extranjero [53].

En este sentido, propone la necesidad de trabajar en pro del software a la par con las tecnologías relacionadas, de tal manera que el sector tenga una mirada holística de la penetración de la tecnología en los procesos organizacionales, y de esta forma se pueda mejorar la industria a partir de esfuerzos divergentes [54].

Las razones por las que no se consolida la implementación de buenas prácticas en el aseguramiento de la calidad del software con el propósito de obtener un producto que genere confianza y permita competitividad a los industriales son: falta de personal capacitado para conocer, interpretar y apropiar las propuestas del orden internacional, falta de convicción de los directivos de la industria para incorporar profesionales en cada área que impacta el proceso de software (gestión, requerimientos, diseño, desarrollo), el tamaño de las empresas que conforman la industria del software local (en su mayoría Mipymes con poca capacidad de inversión) y la falta de reconocimiento de la ingeniería de software como disciplina que soporte todos los proyectos que emprenden [31], [43], [52].

#### **IV. CARACTERIZACIÓN DE LA INDUSTRIA EN LA CIUDAD DE PEREIRA**

Pereira presenta una situación particularmente desfavorable en términos competitivos respecto de la industria nacional e internacional. En el estudio se buscaba conocer las metodologías de desarrollo adoptadas por las empresas locales, los productos que han logrado posicionar, los reconocimientos y certificaciones obtenidos, entre otros elementos propios de su razón de ser.



Para el caso de estas empresas locales se realizó un muestreo por conveniencia [51], el cual consistió en contactar a 25 de las 52 empresas productoras de software que cumplen las condiciones de la población de interés para el estudio en la ciudad de Pereira, a las cuales se les aplicó el instrumento de medición (Encuesta). En el instrumento se indagó a las empresas sobre las metodologías de desarrollo de software utilizadas, así como las certificaciones y reconocimientos adquiridos por ellas.

La información obtenida muestra que 17 de las 25 empresas encuestadas (68 %) hacen uso de metodologías para el desarrollo de sus productos de software. Sin embargo, solo 11 de ellas (44 %) han obtenido certificaciones, reconocimientos o evaluaciones de calidad favorables. Esto permite concluir que a la industria del software de la ciudad aún le falta madurez en sus procesos de desarrollo, lo cual se refleja en un menor grado de reconocimiento de la calidad por parte de organismos de certificación, lo que a su vez plantea mayores retos para incursionar y posicionarse en escenarios internacionales.

Algunas de las metodologías más representativas encontradas en el estudio fueron, entre otras: SCRUM, KANBAN, RUP, LEAN, XP, Métrica V3, PSP/TSP, RAD, Angular, Secuencial. Por su parte, las certificaciones están relacionadas con CMMiDEV, ITIL, ITMark, PMP, ISO 1400, UNE EN ISO 50001, UNE-EN ISO 9001, PECAL 2210, UNE-EN 9110, SAP-AQM. PCOE, ISO 9001-2000, ISO 27001, ISO/IEC 19770-1:2006 SAM Processes.

## V. CONCLUSIONES, A MANERA DE DISCUSIÓN

Si se comprende entonces la Ingeniería de Software como disciplina, como una manera de comportarse, se encuentra una industria de software local que adopta actitudes que difieren del marco teórico y del mismo estado de la cuestión.

Resulta prioritario, incluso para el mismo desarrollo económico y social de la región, promover que la industria del software responda a sus proyectos con la implementación de buenas prácticas de aseguramiento de la calidad, hasta el momento de entrega de un producto confiable al usuario final.

En este sentido, adoptar los modelos y las metodologías actuales al menos le garantizaría

iniciar con las condiciones que le conduzcan a reconocimientos o certificaciones, como una variable importante que el consumidor tendrá en cuenta para optar por un producto software u otro.

El marco teórico y el estado del arte conducen a pensar en la necesidad de formar buenos profesionales, haciendo énfasis en la apropiación de conceptos y metodologías, al tiempo que se debe acompañar al industrial para que vincule personal preparado y decida invertir en los costos necesarios para la implementación de un buen sistema de gestión de calidad.

Todos los autores y todas las apuestas internacionales, nacionales y locales no serán suficientes sin un proceso de culturación hacia la industria, que conduzca a la aceptación de las buenas prácticas y las buenas referencias en el mundo como punto de partida para la industria regional.

Entonces, una forma adecuada de reconocer la importancia de la ingeniería de software en los proyectos adelantados por la industria local, es asumir con disciplina y responsabilidad el proceso mediante el cual se llega a un producto tecnológico como el que nos ocupa en este capítulo.

Como línea de trabajo actual y a mediano plazo, los autores se encuentran trabajando en un modelo para el aseguramiento de la calidad de los requerimientos en proyectos de software, a partir de reconocer la situación actual como línea de base y con la intención de conducir a una apropiación sistemática y metodológica que permita abordar un proyecto software con calidad desde el inicio.

## REFERENCIAS

- [1] A. Cuesta, Modelo para la Industrialización de Software en el Triángulo del Café - Colombia. Salamanca: Universidad Pontificia de Salamanca, 2011.
- [2] Unesco, Informe Mundial de la UNESCO: Hacia las sociedades del conocimiento, Ediciones Unesco, 2005. Disponible en: <http://unesdoc.unesco.org/images/0014/001419/141908s.pdf>
- [3] J. López i Amat, "De la Sociedad de la información a la(s) Sociedad(es) del Conocimiento: Vasos comunicantes en el cambio de milenio, 1960-2010", Tesis de doctorado, Universidad Complutense de Madrid, 2010.

- [4] R. Pressman, Ingeniería del Software: Un enfoque práctico, 7.<sup>a</sup> ed., McGraw Hill, 2010.
- [5] The Joint Task Force for Computing Curricula 2005, Computing Curricula 2005: The Overview Report, A volume of the Computing Curricula Series, USA: ACM IEEE Computer Society, 2006. Disponible en: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf>
- [6] P. Bourque y R. Fairley, Eds., Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0, IEEE Computer Society Press, 2014. Disponible en: <https://www.computer.org/web/swebok/v3>
- [7] International Organization for Estandardization (ISO), Estándar ISO/IEC 2382:2015. Disponible en: <https://www.iso.org/standard/63598.html>
- [8] M. Shaw, "Progress toward an Engineering Discipline of Software", en 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), Austin, TX: IEEE Explore, 2016. Disponible en: <http://ieeexplore.ieee.org/document/7883280/>
- [9] American Society for Quality, Quality Glossary, 2017. Disponible en: <https://asq.org/quality-resources/quality-glossary/>
- [10] P. Crosby, La calidad no cuesta: el arte de cerciorarse de la calidad, Compañía Editorial Continental, 1990.
- [11] J. Juran y A. Godfrey, Juran's Quality Handbook, McGraw-Hill, 1998.
- [12] International Organization for Estandardization (ISO), Estándar ISO 9000:2015. Fundamentos y vocabulario. Disponible en: <https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:es>
- [13] International Organization for Estandardization (ISO), Estándar ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, 2017. Disponible en: <https://www.iso.org/standard/35733.html>
- [14] IEEE Standard Association, Estándar IEEE 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. Disponible en: [https://standards.ieee.org/standard/610\\_12-1990.html](https://standards.ieee.org/standard/610_12-1990.html)
- [15] Joint Task Force on Computing Curricula IEEE, "Computer Society & Association for Computing Machinery", Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2015.
- [16] M. Piattini, J. Calvo-Manzano, J. Cervera, y L. Fernández, Análisis y diseño de aplicaciones informáticas de gestión: Una perspectiva de Ingeniería del Software, España: RA-MA Editorial, 2003.
- [17] J. Cueva Lovelle, "Calidad del Software", Conferencia grupo GIDIS Universidad Nacional de la Pampa, 1999. Disponible en: [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
- [18] D. Heimann, IEEE Standard 730-2014 Software Quality Assurance Processes. Software Quality Group of New England, 2015. Disponible en: <http://sqgne.org/presentations/2014-15/Heimann-Apr-2015.pdf>
- [19] CMMI Product Team, CMMI® for Development, Version 1.3: Improving processes for developing better products and services, Software Engineering Institute, 2010.
- [20] E. Braude, Ingeniería de Software: Una perspectiva orientada a objetos, RA-MA Editorial, 2003.
- [21] F. Shull, V. Basili, B. Boehm, A. Brown, P. Costa, M. Lindvall, M. Zelkowitz, "What we Have Learned about Fighting Defects", en Software Metrics, 2002. Proceedings. Eighth IEEE Symposium on, IEEE, 2002, pp. 249-258. Disponible en: <http://ieeexplore.ieee.org/abstract/document/1011343/>
- [22] International Software Testing Qualifications Board (ISTQB), Standard Glossary of Terms used in Software Testing Version 3.1, 2015.
- [23] D. Garvin, Competing on the Eight Dimensions of Quality, Harvard Business Review, 1987, pp. 101-109.
- [24] J. McCall, P. Richards y G. Walters, Factors in Software Quality, vol. I. Concepts and Definitions of Software Quality, General Electric Co., 1977. Disponible en: <http://www.dtic.mil/docs/citations/ADA049014>
- [25] B. Boehm, J. Brown, H. Kaspar y L. Myron, Characteristics of software quality, TRW Softw. Technol, 1978.
- [26] R. Park, W. Goethert y W. Florac, Goal-Driven Software Measurement. A Guidebook. No. CMU/SEI-96-HB-002, Software Engineering Institute, 1996.
- [27] V. Basili, G. Caldiera y H. Rombach, "The Goal Question Metric (GQM) Approach", en Encyclopedia of Software Engineering, 2002, pp. 528-532.
- [28] D. Henderson, S. Sheetz y L. Wallace, "Understanding Software Metric Use", ISACA Journal, 2015. Disponible en: <https://www.isaca.org>

- org/journal/archives/2015/volume-1/pages/understanding-software-metric-use.aspx
- [29] A. Fuggetta, y E. Di Nitto, “Software Process”, en Proceedings of the on Future of Software Engineering, ACM, 2014, pp. 1-12. Disponible en <https://dl.acm.org/citation.cfm?id=2593883>
- [30] CMMI Architecture Team, Introduction to the Architecture of the CMMI® Framework, Carnegie Mellon, 2007. Disponible en: <ftp://ftp.sei.cmu.edu/pub/documents/07.reports/07tn009.pdf>
- [31] C. Rigoni Brualla, CMMI®: Mejora del proceso en fábricas de software. Oportunidades para la Industria Española del Software en la Economía Global, 2006, pp. 9-44.
- [32] J. Garzás, C. Fernández y M. Piattini, “Una aplicación de ISO/IEC 15504 para la evaluación por niveles de madurez de PYMES y pequeños equipos de desarrollo”, Revista Española de Innovación, Calidad e Ingeniería del Software, vol. 5, N.º 2, pp. 88-98, 2009. Disponible en: <http://listas.ati.es/IMG/pdf/Num2Vol5Sep09.pdf#page=88>
- [33] R. O’Connor y C. Laporte, “The Evolution of the ISO/IEC 29110 Set of Standards and Guides. International”, Journal of Information Technologies and Systems Approach, vol. 10, N.º 1, ene.-jun. de 2017.
- [34] NYCE, “MoProSoft”, en NYCE - Verificación, [En línea], 2016. Disponible en: <https://www.nyce.org.mx/moprosoft-nyce/>
- [35] SOFTEX, “Melhoria de Processo do Software Brasileiro (MPS.BR)”, 2016. Disponible en: <https://www.softex.br/mpsbr/guias/#toggle-id-3>
- [36] Carnegie Mellon Institute, Software Engineering Institute, The IDEAL model, Software Engineering Institute, 2001. Disponible en: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=20208>
- [37] A. Graves, “Six Sigma Fundamentals: DMAIC vs. DMAV”, Six Sigma Daily, [En línea], 2014. Disponible en: <http://www.sixsigmadaily.com/six-sigma-basics-dmaic-vs-dmadv/>
- [38] International Standar ISO/IEC/IEEE 29148-2011, Systems and software engineering – Life cycle processes – Requirements engineering, 2011. Disponible en: <http://mmf.nsu.ru/sites/default/files/iso-iec-ieee-29148-2011.pdf>
- [39] I. Sommerville, Ingeniería de Software, 6.ª ed., Addison Wesley. 2000.
- [40] K. Wiegers y J. Beatty, Software Requirements, 3.ª ed., Washington: Microsoft Press, 2013.
- [41] B. Boehm, y V. Basili, “Software Defect Reduction Top-10 List”, en Foundations of empirical software engineering: The legacy of Victor R. Basili, Springer, 2005.
- [42] C. Denger y T. Olsson, “Quality Assurance in Requirements Engineering”, en Engineering and Managing Software Requirements, Springer, 2005.
- [43] K. Palomino, Estudio del comportamiento de la Industria del Software en Colombia, Medellín: Universidad Nacional de Colombia, Escuela de Minas, 2011.
- [44] L. Peláez, L. Cardona y A. Toro, Estado del arte que soporta el proceso de desarrollo de software en las pymes colombianas, Entre Ciencia e Ingeniería, N.º 10, pp. 93-107, 2011.
- [45] F. García Ramírez P. Puello Marrugo, “Gestión de Requisitos en la Ingeniería del Software”, Ingeniator | Revista Virtual de los Programas de Ingeniería. Universidad de San Buenaventura, vol. 1, N.º 1, pp. 57-65, 2010.
- [46] H. Salazar, H. Rengifo, L. Machuca y J. Aranda, “Una guía general para la especificación y verificación formal de requerimientos usando EVENT-BTM Y RODINTM”, Educación en Ingeniería, pp. 82-91, 2012.
- [47] The Standish Group, CHAOS Report, 2014.
- [48] A. Serrano, Metamodelo para la mejora del proceso de gestión de requisitos, Madrid: Universidad Politécnica de Madrid, 2013.
- [49] C. De La Cruz y G. Castro, “La Ingeniería de requerimientos en las pequeñas empresa del Departamento de Risaralda”, Lámpsakos, pp. 110-119, 2014.
- [50] P. Basto y F. Silveira, Desafíos y oportunidades de la industria del software en América Latina, CEPAL, 2009.
- [51] T. C. Kinnear y J. R. Taylor, Investigación de mercados, 5.ª ed., Bogotá: McGraw-Hill. 1999.
- [52] A. Toro y L. Peláez, “Ingeniería de Requisitos: de la especificación de requisitos de software al aseguramiento de la calidad. Cómo lo hacen las Mipymes desarrolladoras de software de la ciudad de Pereira”, Entre Ciencia e Ingeniería, pp. 117-123, 2016.
- [53] C. López, Detección de defectos de diseño mediante métricas de código, Valladolid: Universidad de Valladolid, 2012.
- [54] Ministerio de las Tecnologías de la Información y las Comunicaciones (Mintic) y Federación Colombiana de la Industria de Software y TI (Fedesoft), Informe de Caracterización del Sector Software y Tecnologías de la Información en Colombia, Bogotá D.C.: Fedesoft, 2015.



# Capítulo III

## METODOLOGÍAS CERCANAS AL DESARROLLO DE SOFTWARE LIBRE Y DE CÓDIGO ABIERTO (FOSS). EVALUANDO EL PROCESO DE DESARROLLO DESDE LA COOPERACIÓN

*Byron Cuesta Quintero - byroncuesta@ufpso.edu.co*

Docente e investigador, Departamento de Sistemas e Informática,  
Universidad Francisco de Paula Santander, Ocaña

*Jorge Andrick Parra Valencia - andrick.parra@gmail.com*

Docente e investigador, Universidad Autónoma de Bucaramanga, Colombia

### I. INTRODUCCIÓN

El software libre es un fenómeno que es posible debido a la cooperación de comunidades distribuidas en red, que se reúnen voluntariamente para desarrollar un software compartido que resuelve una necesidad común de varias personas [1]. El contenido del documento muestra elementos que permiten interpretar la realidad de las prácticas de cooperación en el desarrollo de código abierto iniciando con entender su organización bajo un modelo para administrar, contribuir y mantener un código fuente libre, fundamentado en elementos que generan acciones que favorecen las actividades grupales sobre bienes comunes como la confianza y la reciprocidad.

En los proyectos de software libre y de código abierto, el proceso de desarrollo no está asociado con una metodología demasiado estricta para la construcción del software. Generalmente utiliza prácticas más informales que se pueden asociar con elementos propios de las metodologías de desarrollo ágil, como el trabajo en equipo, la entrega temprana, el modelo iterativo y adaptativo ante los cambios de las condiciones iniciales del proyecto y el valor de negocio para los clientes debido a las entregas continuas. Para el caso de estudio se escogieron metodologías ágiles como XP y Scrum, por ser cercanas al desarrollo de proyectos de FOSS [2].

El resultado de la investigación permitió identificar y asociar un conjunto de elementos o prácticas similares en las fases por las que pasa un proyecto software, indagando metodologías del Software Libre, el mundo ágil, y su asociación con la aplicación de acciones cooperativas que permiten sumar y resolver conflictos en el desarrollo de proyectos de código abierto.

### II. ANTECEDENTES

#### A) El Ambiente Cooperativo

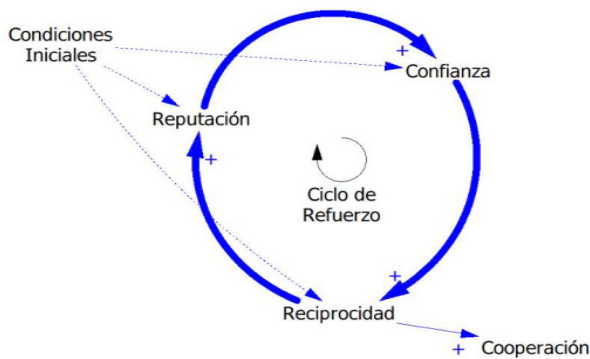
Las acciones colectivas generan interacciones en las que los participantes se enfrentan a momentos en los cuales las personas pueden tomar decisiones independientes en situaciones que afectan al colectivo por ser dependientes [3]. Estas situaciones, en las que la acción individual afecta a los demás, se conocen como dilemas sociales y se asocian con los conflictos de racionalidad individual y colectiva [4], [5]. Una acción individual asociada con negarse a contribuir (*free-rider*<sup>1</sup>) puede afectar la cooperación y los intereses colectivos [3]. Si todos en el colectivo contribuyen, todos obtendrán un beneficio positivo, pero también existe la posibilidad de que los participantes se enfrenten a situaciones que hagan

---

<sup>1</sup> Persona que quiere beneficiarse de un bien común sin contribuir a su mantenimiento.

cambiar su decisión inicial para beneficiarse de manera individual, así exista una situación que dé mayor valor a todos [3], [6].

En una colectividad, un ambiente de cooperación se establece mediante la relación de variables estructurales como la confianza, generada por las expectativas sobre el comportamiento de los individuos, la reciprocidad, expresada en las normas producto de la socialización entre los individuos y la reputación, a través de crear identidad individual [7]. Como se muestra en la Figura 1, estas variables al ser cambiadas pueden modificar el nivel de cooperación de los individuos en comunidades con bienes comunes, para alcanzar acciones colectivas positivas, lo que permite resolver conflictos y crear identidades grupales a través del uso de elementos como la comunicación cara a cara, el compromiso mutuo y la creación y el refuerzo de las normas [8], [9], [10].



**Fig. 1.** Variables estructurales para el mecanismo de cooperación [11].

Una situación representada por una acción individual puede cambiar si todos deciden contribuir y generar un beneficio positivo para todos. En los dilemas sociales, los individuos se pueden enfrentar a tentaciones que los pueden hacer cambiar y tomar decisiones aisladas, y nadie se encuentra motivado a cambiar su preferencia dadas las elecciones de los demás participantes, así exista un resultado que dé una mayor ventaja a todos [7-9].

Cuando se involucran varios actores, se distinguen dos grandes tipos de dilemas: el de los bienes públicos, que se refiere a la existencia de la tentación de disfrutar de lo bueno sin contribuir a su creación, es

decir, obtener el mejor resultado posible sin cooperar, y el de los bienes comunes, en el cual el individuo se siente tentado con un beneficio inmediato que produce un costo compartido por todos. Si todos sucumben a la tentación, el resultado es un desastre colectivo [4, 9].

Las personas pueden ser egoístas en sus beneficios inmediatos y no siempre son generosas para el beneficio en grupo. De esta manera, cuando el contexto lo hace posible, las personas se ganan la confianza de otros individuos, y a partir de ahí se pueden resolver dilemas sociales y mantener la cooperación dentro de los grupos, pero también los cambios en ese contexto pueden destruir la cooperación que ha surgido. En la teoría de los juegos se utiliza la estrategia tit-for-tat para resolver de manera óptima el dilema del prisionero, y evidenciar que cuando muchos individuos utilizan reciprocidad hay un incentivo para adquirir una reputación basada en la confianza [10].

## B. Cooperación en las Comunidades FOSS

Las personas que hacen parte de las comunidades de desarrolladores de software libre se caracterizan por estar geográficamente dispersos en el mundo, formando equipos virtuales autoorganizados alrededor de un recurso común, el código fuente, al cual se accede de forma libre y se gestiona sin necesidad de una estructura formal o autoridad jerárquica [12]. El acceso a los recursos compartidos a través de canales de comunicación mediada por computador, da un valor que está representado en la motivación que encuentran los desarrolladores por querer participar y contribuir en el desarrollo, formando redes profesionales virtuales con creencias y valores en común, que permiten crear relaciones de confianza que ayudan a aprender de otros, a partir de la disposición en la red del propio software y de artefactos como las discusiones de listas de correo (*kernel cousins*<sup>2</sup>) [13], [47], [53].

La construcción del Software Libre se realiza a través de la organización de un grupo de participantes que actúan de forma voluntaria, con el ánimo de lograr un beneficio mutuo. De esa interacción de los desarrolladores alrededor de un recurso compartido, surge la necesidad de cooperar para alcanzar los objetivos [2]. Compartir código fuente permite en cada desarrollador una interdependencia en la red sobre el uso del recurso, con la cual se consigue la

<sup>2</sup> Repositorios de artefactos de software.

creación de un modelo de buenas prácticas asociadas en un ecosistema de desarrollo de software, que genera para cada proyecto un crecimiento de forma exponencial en lo referente a su código fuente [13]. Los aportes individuales al código fuente y el hecho de ser un recurso de acceso libre generan externalidades positivas para los demás, que repercuten en acciones colectivas para la conservación del recurso común. El resultado de cada contribución representa lo que se conoce como “externalidad de red”, que puede ser positiva o negativa dependiendo de cómo las acciones individuales crean la externalidad al resto del grupo [9], [14], [15].

Algunos casos de éxito asociados con proyectos de Software Libre y de código abierto (FOSS), muestran cómo la cooperación basada en acciones de confianza es necesaria para lograr los objetivos en el desarrollo del software. Veamos:

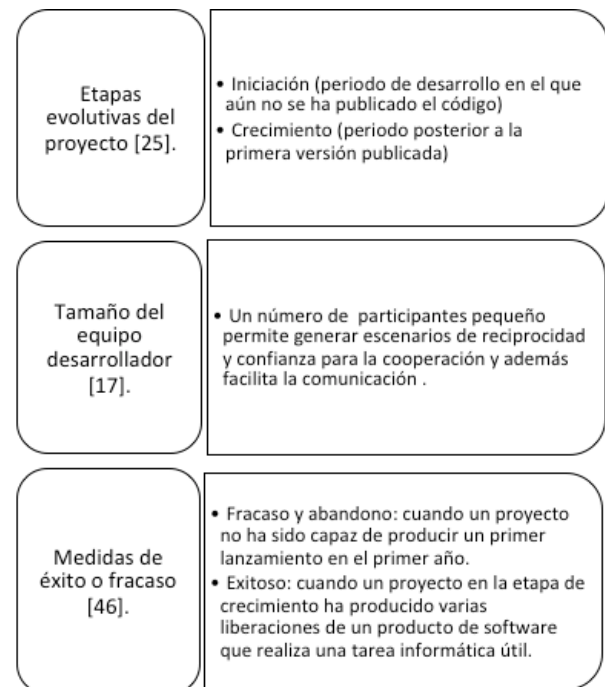
**GNOME:** Se realiza un análisis de las asociaciones entre los programadores y los archivos, utilizando los datos de los aportes de los participantes en el repositorio CVS. El estudio muestra como resultado que en los archivos grandes se evidencia el trabajo cooperativo de un círculo reducido de desarrolladores [16], [17].

**APACHE:** Se realiza un análisis de la participación y motivación de los desarrolladores del proyecto Apache. El estudio muestra como resultado que la mayoría de los módulos fueron escritos por un grupo central, que tuvo un nivel suficiente de confianza mutua para realizar contribuciones al código [18], [19].

**GNU Enterprise (GNUe):** Investigación sobre una comunidad en línea mediante una cultura organizacional, donde los participantes no conviven cara a cara y los conflictos se resuelven a través de herramientas telemáticas. Resultado: La comunicación mediada por ordenador (Internet Relay Chat –IRC–, Sistema de control de versiones –CVS– y listas de correo) facilita el trabajo en equipo y la cooperación para construir comunidad [20], [21].

Otros resultados generados en los estudios anteriores están relacionados con las comunidades virtuales

alrededor de proyectos de código abierto, es decir, el trabajo independiente de desarrolladores dispersos globalmente con un propósito compartido, que utilizan la tecnología para lograr resultados exitosos a nivel grupal [16], [17], [18], [21]. Generalmente los desarrolladores buscan reconocimiento impulsados por la necesidad de aportar para obtener reputación y ser catalogados como colaboradores de confianza, para que les sean asignadas nuevas responsabilidades [2], [22], [23], [24]. La Figura 2 muestra un esquema de los atributos con los cuales es posible hacer seguimiento a proyectos de código abierto, para verificar su éxito o fracaso. Esta información puede ser ubicada en sitios de alojamiento como Sourceforge.net, con datos como: cuándo fue lanzada la primera versión, los releases del software o si el proyecto fue abandonado.



**Fig. 2.** Esquema de atributos para el seguimiento a proyectos de código abierto.

### C. Metodologías de Software Cercanas a Proyectos FOSS

**Software Libre:** Las actividades y resultados en un proceso de Software Libre, se asocian con un conjunto de tareas sin una especificación estricta de un modelo clásico de desarrollo de software, soportado en la existencia de una cultura abierta, las creencias en las

raíces del software libre y el trabajo distribuido en un ambiente de cooperación [12], [26]. Un ejemplo de este modelo de desarrollo es el *kernel* del sistema operativo Linux, el cual fue desarrollado bajo un modelo tipo bazar, cuya característica está representada en la ausencia de autoridades que controlen la planificación de las tareas, como sucede en los modelos propietarios [2]. En el modelo del software libre, generalmente los usuarios son los mismos desarrolladores y los proyectos surgen a raíz de una necesidad personal [27]. La idea o necesidad personal empuja a los desarrolladores a construir el software y cuando tienen una solución funcional la liberan de forma temprana, compartiendo el código fuente en una comunidad virtual, lo cual permite buscar otros desarrolladores con igual necesidad para que en un ambiente de cooperación contribuyan a la solución y se beneficien mutuamente. Los aportes mutuos generan que el código fuente tenga integración continua y un nuevo valor agregado en cada entrega [28], [29].

La especificación de requerimientos en los proyectos de software libre no constituye una tarea necesaria plasmada en un documento [2]. Al liberar el software de forma temprana, los requisitos surgen implícitamente como afirmaciones en repositorios e hilos de discusión [13].

En los proyectos de código abierto hay una carencia de diseño detallado. Esto dificulta la participación de nuevos desarrolladores y que, en algunos casos, la reutilización de código sea escasa [2].

El desarrollo de software libre surge como una necesidad de algún desarrollador, quien lo diseña e implementa, y a medida que logra funcionalidad en el software lo libera y comparte con desarrolladores que tengan las mismas necesidades. De esta manera, durante la implementación los mismos desarrolladores siguen una programación a prueba y error hasta que se consiguen los resultados deseados desde el punto de vista subjetivo del programador [28], [29], [30].

Para el mantenimiento se utilizan herramientas de control de versiones (CVS). Estas herramientas permiten tener un mecanismo centralizado para coordinar el desarrollo y controlar las mejoras que se incorporan a los archivos del software. Cada equipo

del proyecto o administrador del repositorio CVS debe decidir y resolver conflictos para saber si la nueva versión aplica para liberarla como una versión estable [17], [18], [21].

**Programación Extrema:** La Programación Extrema (XP) es una metodología ágil que enfatiza la necesidad de adaptarse a los cambios de requisitos durante todo el proyecto, en vez de definirlos al comienzo, lo que sugiere menos esfuerzo. Se caracteriza por crear ambientes de trabajo que facilitan las relaciones interpersonales del equipo de desarrollo y la mejora en la comunicación con el cliente. XP se basa en cuatro valores: la simplicidad en el proceso y la codificación, la comunicación fluida para evitar conflictos, la retroalimentación continua entre los actores (cliente, equipo, usuarios) y el coraje para enfrentar los cambios [31], [32].

En la fase de Planificación se redactan las historias de los usuarios que escriben los propios clientes, tal y como ven ellos las necesidades del sistema. Se crea un plan de entregas que incluye iteraciones individuales, que corresponden a un periodo de tiempo de desarrollo del proyecto de entre una y tres semanas. Se utilizan las rotaciones de personal para que todo el mundo conozca cómo funciona el sistema en general. El reparto de tareas se asigna por parejas, lo cual facilita la comunicación, y se realizan reuniones diarias, de poca duración y, de ser posible, delante de la pantalla del ordenador [33], [34].

En la fase de Diseño se trata siempre de realizar las tareas de la manera sencilla. Se elige una metáfora para el sistema, que es una historia acerca de cómo el sistema funciona. Se llevan a cabo reuniones entre todos los desarrolladores implicados, donde se toman las decisiones mediante el uso de tarjetas CRC (class, responsibilities and collaboration). Cada tarjeta representa un objeto del sistema, en la que aparece escrito en la parte superior el nombre del desarrollador, sus responsabilidades en la parte izquierda y los objetos con los que colabora en la parte derecha [35].

En la fase de Desarrollo, el cliente está siempre disponible. Durante la reunión del plan de entregas, el usuario propondrá qué historia de usuario se incluye en cada plan. Es posible refactorizar los módulos

existentes, dejarlos igual de simples pero con nuevas funcionalidades añadidas. Debido a las rotaciones de personal y a que el código es común a todos los programadores, es importante que se definan prácticas propias de los lenguajes de programación, por ejemplo, la indentación y los nombres de variables. Todo el código es desarrollado por dos personas que trabajarán de forma conjunta en un ordenador. Sólo una pareja se encargará de integrar el código para evitar problemas a causa de la integración de los módulos que se han desarrollado y no han sido testeados todavía. Lo ideal es trabajar 40 horas semanales, y es muy importante definir claramente el plan de entregas [36], [37].

En la fase de Pruebas todo el código debe ir acompañado de su unidad de pruebas. Descubrir todos los errores que pueden aparecer lleva tiempo, y más si se deja la depuración total para el final. Una historia de usuario no se considera completa mientras no supere sus pruebas de aceptación [38], [39].

**Scrum:** Scrum es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. Una fortaleza clave de Scrum radica en el uso de equipos multifuncionales, autoorganizados y con poder, que dividen su trabajo en ciclos de trabajo cortos y concentrados llamados *sprints* [40], [41], [42].

Scrum define una estructura de tres elementos: principios, aspectos y procesos.

Los principios son pautas básicas de aplicación, son seis y son de uso obligatorio en todos los proyectos [41].

1. Control del proceso empírico (Transparencia, inspección y adaptación).
2. Autoorganización (Equipos con un gran compromiso y responsabilidad).
3. Colaboración (Valor compartido: conciencia, articulación y apropiación).
4. Priorización basada en valor (Ofrecer máximo valor de negocio).
5. Tiempo asignado (Planificación: Sprints, Daily Standup Meetings, Sprint Planning Meetings, y Sprint Review Meetings)

6. Desarrollo iterativo (Manejo de cambios y crear productos que satisfagan las necesidades del cliente)

Los aspectos son considerados importantes para todos los proyectos Scrum [41].

1. Organización (Roles y responsabilidades del proyecto. Define responsabilidad compartida e innovación continua).
2. Justificación del negocio (Entrega temprana de resultados. Priorizar requisitos, gestionar riesgos y cambios en el sprint y obtener valor en cada incremento del producto).
3. Calidad (Capacidad del producto entregable de cumplir con los criterios de aceptación y de alcanzar el valor de negocio que espera el cliente).
4. Cambio (Los procesos de desarrollo están diseñados para aceptar el cambio mediante el uso de los sprints cortos y repetitivos que incorporan la retroalimentación con el cliente en cada entrega del sprint).
5. Riesgo (Evento incierto que puede afectar los objetivos de un proyecto. Los riesgos gestionados deben incluirse en el Product Backlog para que el Scrum Team pueda mitigarlos)

Scrum está compuesto por cinco fases que incluyen diecinueve procesos. Los procesos abordan las actividades y el flujo específico de un proyecto [41].

La fase de inicio surge a partir de un caso de negocio, el cual justifica la necesidad de implementar un proyecto, con este insumo se crea la visión del proyecto, que es una referencia del equipo Scrum para su implementación, se definen los roles de los participantes, se crean las historias épicas, que son el insumo para el *product backlog*<sup>3</sup> priorizado, y se realiza la planeación de las fechas de liberación correspondiente a entregas parciales del proyecto.

En la fase de planeación y estimación se crean las historias de usuario con su respectiva aprobación y estimación, las tareas estimadas y las tareas del *sprint backlog*<sup>4</sup>.

En la fase de implementación el equipo de desarrollo crea los entregables a partir de iteraciones denominadas sprints, incluye realizar diariamente la validación del

<sup>3</sup> Representa una lista de necesidades que tienen que ser resueltas para lograr el éxito del proyecto.

<sup>4</sup> Representa una serie de tareas que el equipo de desarrollo tiene que implementar en un tiempo determinado.



estado de los entregables con respecto a las fechas estimadas, por medio de reuniones denominadas *daily stand up* o *daily Scrum* que pueden repercutir en el perfeccionamiento del *product backlog*.

En la fase de Revisión y retrospectiva se hacen reuniones que van acompañadas con demostraciones y validación del *sprint*. Finalmente, la fase de lanzamiento ocurre al finalizar el *sprint*, con la cual se hace una liberación del producto que incluye la entrega al cliente de los entregables o puesta en producción, y la reunión de Retrospectiva del proyecto.

#### D) Cooperación en el Mundo Ágil

Los valores expresados en el manifiesto ágil<sup>5</sup> y su asociación con sus principios muestran la importancia del trabajo en equipo y cómo, en las interacciones cotidianas de los desarrolladores y los responsables del negocio, es necesario crear ambientes de cooperación para alcanzar los objetivos. El escenario muestra desarrolladores motivados alrededor del código fuente compartido, donde prevalece la comunicación cara a cara para resolver las necesidades de desarrollo en un ambiente de ritmo constante y cambios inesperados, asociados con modificaciones o nuevos requerimientos software no definidos inicialmente [54].

En XP los valores expresan las acciones del equipo de desarrollo, los cuales se pueden asociar para identificar elementos de cooperación, como la comunicación verbal entre clientes y desarrolladores, lo cual permite una colaboración eficaz, la simplicidad a través de soluciones inmediatas que garanticen maximizar la cantidad de entregas, la retroalimentación para mejorar y ajustar el software a las necesidades del cliente y la valentía y respeto de los desarrolladores para afrontar los cambios, entendiendo que los aportes están expuestos al debate para lograr un trabajo recíproco que beneficie a todos [33].

En Scrum los principios indican la necesidad de tener equipos de desarrollo autoorganizados, que utilizan prácticas cooperativas basadas en el compromiso mutuo de los participantes para enfrentar los cambios

y una constante comunicación para solucionar las ambigüedades, dándole valor compartido al conjunto de tareas priorizadas, que permiten minimizar los tiempos de desarrollo para cumplir con los requerimientos del cliente y buscar mayor valor de negocio en cada entrega [41, 42].

Los equipos de desarrollo en el mundo ágil se caracterizan por tener una clara visión del producto que se va a desarrollar, y de esta manera garantizar que el software por construir cumpla con los objetivos del negocio. Esta característica que define una guía global del proceso de desarrollo, se denomina metáfora en la metodología XP [31]. En las metodologías ágiles, durante el proceso de desarrollo, los individuos y sus interacciones generan ambientes propios que favorecen la comunicación y permiten mitigar problemas que son visibles durante los ciclos de inspección y adaptación, los cuales son aplicados de forma frecuente y se evidencian en las prácticas de desarrollo como la programación en parejas, la integración continua y las reuniones diarias de revisión y retrospectiva. Los problemas de comunicación en el mundo ágil se mejoran cuando los desarrolladores tienen un entorno para trabajar juntos, con normas representadas en elementos como el respeto por los participantes y sus contribuciones, la confianza en la comunicación y el compromiso con el equipo y sus objetivos para entregar valor [55].

### III. MÉTODO

Este estudio tuvo como objetivo identificar y comprender las prácticas de desarrollo de software comunes en los proyectos de software libre y las metodologías ágiles, haciendo énfasis en cómo la cooperación es el elemento integrador que permite la adecuada gestión de recursos comunes, y en la importancia del establecimiento de normas para el desarrollo y mantenimiento del software por las comunidades de desarrolladores. Se tuvo como fuente principal la revisión de literatura dentro de un marco conceptual que incluyó las teorías de la cooperación en general, la cooperación en las comunidades FOSS, las metodologías cercanas a proyectos FOSS y la cooperación en el mundo ágil.

<sup>5</sup> Documento que resume en cuatro valores y doce principios las mejores prácticas para el desarrollo de software.

## IV. RESULTADOS

Este artículo presenta un conjunto de prácticas de procesos software, la influencia de la cooperación propia de los proyectos de software libre y su asociación con las metodologías ágiles cercanas al desarrollo de código abierto: XP y Scrum.

### A. El Ambiente Cooperativo y la Cooperación en las Comunidades FOSS

La Tabla 1 muestra elementos que promueven la cooperación a partir de la revisión de las teorías de la cooperación, la cooperación en las comunidades FOSS y la cooperación en el mundo ágil.

**Tabla 1.** Elementos de Cooperación en el Software Libre y las Metodologías Ágiles

Teorías de la Cooperación	Software Libre	Mundo Ágil
<b>Contribuir y mantener bienes comunes</b>		
Acciones para contribuir y mantener un bien compartido [4].	Acceder libremente al código fuente para estudiar, adaptar y redistribuir [30].	Desarrolladores motivados alrededor de código fuente compartido [54].
<b>Crear grupos pequeños con identidad</b>		
Facilita la comunicación a través del uso de normas sobre el uso de recursos comunes [3].	Comunicación mediada por computador (IRC, CVS y listas de correo), facilita el trabajo en equipo y la construcción de comunidad [21, 46, 47, 52].	Equipos pequeños de desarrollo, autoorganizados [41, 42].
<b>Promover la confianza</b>		
Compromiso mutuo, expectativas sobre el comportamiento de los demás [7].	Cambiar los roles de los participantes de forma continua. Un participante puede trabajar en un módulo y luego pasar a trabajar en otro [18].	Confianza en la comunicación y el compromiso con el equipo para entregar valor [41, 55].

Teorías de la Cooperación	Software Libre	Mundo Ágil
<b>Promover la reciprocidad</b>		
Correspondencia mutua, creación y refuerzo de las normas que los individuos aprenden de la socialización y las experiencias de la vida [7].	Participar para aprender y desarrollar nuevas habilidades [2, 56].  Integrar el código fuente frecuentemente [30].	Afrontar cambios, entendiendo que los aportes están expuestos al debate para lograr un trabajo recíproco que beneficie a todos [33, 34].
<b>Obtener reputación</b>		
Los individuos crean identidades que proyectan sus intenciones [7].	Publicar versiones de forma temprana y frecuente [30, 49].	Ciclos frecuentes de inspección y adaptación que facilitan las entregas tempranas [55].

### B. Metodologías Ágiles y el Software Libre

La Tabla 2 muestra una comparación entre las etapas de desarrollo del Software Libre y del mundo ágil: XP y Scrum.

**Tabla 2.** Etapas de Software Libre y Ágiles

Software libre	XP	SCRUM
<b>Análisis y especificación de requerimientos</b>		
Los requisitos aparecen como afirmaciones en los hilos de discusión sobre lo que el software debe o no debe hacer [13].	Los clientes redactan las historias de los usuarios. Se crea un plan de entregas que incluye iteraciones [32], [33], [34].	En la fase de inicio se crea la visión del producto y el <i>product backlog</i> priorizado, con las fechas de liberación correspondiente a entregas parciales del proyecto [41].
<b>Diseño</b>		
El software libre surge como una necesidad de algún desarrollador, quien lo desarrolla y lo libera compartiéndolo en la red [28], [29], [48], [50].	Realizar diseños simples. Si alguna parte de la implementación resulta especialmente compleja, se debe replantear [31], [32].	En la Fase de Planeación y Estimación, se crean las historias de usuario que sirven de insumo para el <i>sprint backlog</i> [41].

Software libre	XP	SCRUM
<b>Implementación y pruebas</b>		
Publicar versiones de prueba de forma temprana y frecuente [2], [30], [51].	Publicar versiones de prueba de forma temprana y frecuente [33].  Refactorizar, permite modificar o ampliar partes de código ya escritas anteriormente [35], [37], [38].	Crear versiones denominadas entregables, a partir de iteraciones llamadas <i>sprints</i> [41].  En la fase de Revisión y retrospectiva, hacer demostraciones y validación del <i>sprint</i> .
<b>Mantenimiento</b>		
Se utilizan herramientas de control de versiones (CVS) para coordinar el desarrollo y controlar las mejoras que se incorporan a los archivos de software [2], [17], [18], [21], [16], [45].	Con el sistema en producción, la velocidad de desarrollo suele disminuir y es posible requerir nuevo personal para cumplir con tareas de soporte para el cliente [35].	Después de finalizada la fase de lanzamiento o puesta en producción, es posible definir nuevas iteraciones para el desarrollo de productos para soporte y arreglos necesarios [41].

Este trabajo sigue la línea y aporta a los trabajos realizados por: 1) Corbucci y Goldman [57], quienes presentan un conjunto de características comunes a los métodos ágiles y las comunidades de software de código abierto que permiten mejorar el desarrollo de software, donde se identifican factores y técnicas para mejorar la comunicación de las partes interesadas, el establecimiento de un entorno que brinde al equipo de desarrollo información sobre su trabajo, el uso de historias de usuario para un mejor seguimiento a los requisitos y la adopción de la reunión de retrospectiva en el mismo espacio de trabajo con una interacción de manera colectiva, usando canales de IRC y asociando el contenido al sitio web del proyecto. 2) Fellhofer, Harzl y Slany [58] enseñan un documento que muestra la necesidad de adaptar las técnicas de desarrollo ágil a la colaboración remota, a través del uso de herramientas de comunicación. Además muestran la adopción de métodos de desarrollo ágil, como la programación en parejas (XP), para la transferencia de conocimiento entre desarrolladores y el desarrollo dirigido por pruebas (TDD), con el fin

de garantizar que el código siga siendo funcional y comprobable, la usabilidad y experiencia de usuario (UX) y la importancia de una documentación disponible en línea. 3) Valerii, Do Cao y Oke [59] exploran el uso de SCRUM, el cual se basa en una estrecha cooperación entre los desarrolladores con reuniones cara a cara, organizadas de forma regular, y el desarrollo por parte de las comunidades FOSS, que se realiza principalmente por personas distribuidas geográficamente. Los autores combinan prácticas de los dos aspectos de desarrollo y lo denominan “Scrum distribuido”, dan importancia a las personas y las relaciones para aumentar el nivel de comunicación de los desarrolladores y la comprensión de las funciones de los compañeros de trabajo.

## V. DISCUSIÓN

Los resultados obtenidos a partir del marco conceptual permitieron establecer el siguiente análisis:

El trabajo voluntario de los desarrolladores de Software Libre y código abierto está representado en el costo y el aporte que deben asumir los participantes de forma individual, para alcanzar un beneficio colectivo alrededor de las contribuciones y el mantenimiento sobre un bien compartido de acceso libre [4]. Las diferentes contribuciones generan dependencias entre los participantes regidas por el establecimiento de normas de confianza apoyadas en el compromiso mutuo y basadas en la reciprocidad, que permiten fortalecer el grupo y facilitan la comunicación para formar un ecosistema de software alrededor de comunidades virtuales [7], [21].

En las metodologías ágiles se encuentran elementos y prácticas comunes al proceso de desarrollo de Software Libre, de las cuales se puede destacar: su énfasis orientado a las personas, la importancia del cliente en el equipo de desarrollo y su aplicación en proyectos innovadores, donde se necesita coraje para enfrentar los cambios que surjan en todo el proyecto. La fase de desarrollo se caracteriza por una planeación baja con una documentación mínima, gestionada por equipos pequeños autoorganizados que trabajan en ambientes de colaboración y cooperación con responsabilidades compartidas, donde los procesos tienen un estilo adaptativo e iterativo constante, el cual permite hacer entregas tempranas y dar valor de negocio para el cliente en cada entrega del producto durante la vida del proyecto [40], [41], [42], [43].



Generalmente la resolución de conflictos en estos ambientes de desarrollo fomenta el diálogo abierto de los desarrolladores, ya sea cara a cara o a través de una comunicación mediada por computador en un entorno saludable, que genera interdependencia alrededor de un recurso común, entendiendo que una acción positiva sobre el código fuente compartido repercute en el beneficio mutuo del equipo de desarrollo [13], [44], [53].

## VI. CONCLUSIONES

La cooperación es el engranaje que permite a las comunidades que se benefician de un bien común encontrar la motivación de sus participantes para contribuir individualmente con su mantenimiento, y así obtener réditos para todos los participantes. Estos aportes positivos se apoyan en la articulación de acciones recíprocas suficientes para establecer reglas de comportamiento basadas en la confianza, que permiten alcanzar identidades de grupo que facilitan el trabajo en equipo y la resolución de conflictos.

Un modelo de desarrollo de software representa un conjunto de actividades y resultados de un proceso que se afianza de acuerdo con las condiciones para trabajar en equipo voluntariamente, entendiendo la reciprocidad necesaria de los desarrolladores para construir y mantener software de código abierto a partir de acciones que generen confianza, para lograr escenarios de cooperación que faciliten la comunicación y permitan obtener beneficios positivos para todos los participantes que forman la comunidad.

Las prácticas de desarrollo de software en el mundo de código abierto se acercan por su naturaleza con las metodologías ágiles, por su enfoque basado en compartir el código fuente, la ausencia de jerarquía, el cliente en sitio, su estilo iterativo y adaptativo para enfrentar los cambios surgidos durante un proyecto, la refactorización del código para añadir nuevas funcionalidades, las pruebas unitarias de forma continua y el valor de retorno propio de entregas tempranas y frecuentes, que repercuten en integración continua del producto desarrollado.

La comunicación es un valor esencial que promueve la cooperación en los procesos de desarrollo de las metodologías ágiles, y se encuentra sustentada en los ciclos frecuentes de inspección y adaptación, donde el cliente y los desarrolladores trabajan juntos de forma continua en cada iteración del producto, formando equipos autoorganizados con normas que permiten

resolver conflictos a partir de la utilización de prácticas de desarrollo que facilitan la comunicación, como la programación en parejas, la integración continua y las reuniones propias del mundo ágil.

## REFERENCIAS

- [1] R. Krishna, Analysis of new paradigms for the development and application in free and open source software Engineering, 2009. Disponible en: <http://krishnarajpm.com/pdf/foss.pdf>
- [2] J. González Barahona, J. Seoane y G. Robles, “El desarrollador y sus motivaciones”, en *Introducción al software libre*, Barcelona: Universitat Oberta de Catalunya, 2003, pp. 91-245.
- [3] E. Ostrom, Social dilemmas and human behavior, en *Economics in Nature: Social Dilemmas, Mate Choice and Biological Markets*, R. Noe, J.A.R.A.M. van Hooff y P. Hammerstein, eds., Cambridge: Cambridge University Press, 2001, pp. 23-41.
- [4] P. Kollock, “Social dilemmas: The anatomy of cooperation”, *Annual Review of Sociology*, vol. 24, N.º 1, pp. 183-214, 1998.
- [5] A. Biel, C. Von Borgstede y U. Dahlstrand, “Norm Perception and Cooperation in Large Scale Social Dilemmas”, *Resolving Social Dilemmas: Dynamic, Structural, and Intergroup Aspects*, 2013, p. 245.
- [6] M. Foddy, M. Smithson, S. Schneider y M. A. Hogg, *Resolving social dilemmas: Dynamic, structural, and intergroup aspects*, Psychology Press, 2013.
- [7] E. Ostrom, “A Behavioral Approach to the Rational Choice Theory of Collective Action Presidential Address, American Political Science Association, 1997”, *Public administration issues, Higher School of Economics*, Issue 1, 2010, pp. 5-52.
- [8] G. Hardin, “The tragedy of the commons”, *Journal of Natural Resources Policy Research*, vol. 1, N.º 3, 2009, pp. 243-253.
- [9] P. Kevenhórster, “Elinor Ostrom, Governing the Commons. The Evolution of Institutions for Collective Action, Cambridge 1990”, en *Schlüsselwerke der Politikwissenschaft*, VS Verlag für Sozialwissenschaften, 2007, pp. 349-352.
- [10] J. Walker y E. Ostrom, “Trust and reciprocity as foundations for cooperation: Individuals, institutions, and context”, en *Capstone Meeting of the RSF Trust Initiative at the Russell Sage Foundation*, 2007.

- [11] J. Parra, “Constructo para la evaluación de la cooperación en dilemas sociales de gran escala”, Tesis de doctorado, Universidad Nacional de Colombia, 2010.
- [12] G. Giaglis y D. Spinellis, “Division of effort, productivity, quality, and relationships in FLOSS virtual teams: Evidence from the FreeBSD project”, *Journal of Universal Computer Science*, vol. 18, N.º 19, pp. 2625-2645, 2012. doi: 10.3217/jucs-018-19-2625.
- [13] I. Steinmacher, T. Conte, M.A. Gerosa y D. Redmiles, “Social barriers faced by newcomers placing their first contribution in open source software projects”, en *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*, ACM, 2015, pp. 1379-1392.
- [14] E. Ostrom, “Collective action and the evolution of social norms”, *Journal of Natural Resources Policy Research*, vol. 6, N.º 4, pp. 235-252, 2014.
- [15] W. Kathan, K. Hutter, J. Füller y J. Hautz, “Reciprocity vs. free-riding in innovation contest communities”, *Creativity and Innovation Management*, vol. 24, N.º 3, pp. 537-549, 2015.
- [16] G. Poo-Caamaño, L. Singer, E. Knauss y D. M. German, “Herding cats: a case study of release management in an open collaboration ecosystem”, en *IFIP International Conference on Open Source Systems*, Springer, Cham, 2016, pp. 147-162.
- [17] G. Robles, J. M. González-Barahona, C. Cervigón, A. Capiluppi y D. Izquierdo-Cortázar, “Estimating development effort in free/open source software projects by mining software repositories: A case study of openstack”, en *Proceedings of the 11th Working Conference on Mining Software Repositories*, ACM, 2014, pp. 222-231.
- [18] T. Chełkowski, P. Gloor y D. Jemielniak, “Inequalities in open source software development: Analysis of contributor’s commits in apache software foundation projects”, *PloS One*, vol. 11, N.º 4, 2016. Disponible en: <https://doi.org/10.1371/journal.pone.0152976>
- [19] M. Gharehyazie, D. Posnett, B. Vasilescu y V. Filkov, “Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation”, *Empirical Software Engineering*, vol. 20, N.º 5, pp. 1318-1353, 2015.
- [20] M. MM. Syeed y I. Hammouda, “Who contributes to what? Exploring hidden relationships between FLOSS projects”, en *IFIP International Conference on Open Source Systems*, Springer, Berlin: Heidelberg, 2014, pp. 21-30.
- [21] A. Filippova y H. Cho, “Mudslinging and manners: Unpacking conflict in free and open source software”, en *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 1393-1403. ACM, 2015.
- [22] Z. Iskoujina y J. Roberts, “Knowledge sharing in open source software communities: motivations and management”, *Journal of Knowledge Management*, vol. 19, N.º 4, pp. 791-813, 2015.
- [23] J. Howison y K. Crowston, “Collaboration through open superposition”, *Mis Quarterly*, vol. 38, N.º 1, pp. 29-50, 2014.
- [24] N. Choi y I. Chengalur-Smith, “Characteristics of Open Source Software Projects for the General Population: Reciprocity and Network Effects”, *Journal of Computer Information Systems*, vol. 56, N.º 1, pp. 22-30, 2016.
- [25] C. Schweik y R. English, *Tragedy of the FOSS Commons? Investigating the Institutional Designs of Free/Libre and Open Source Software Projects*, Washington, DC: IEEE Computer Society, 2007.
- [26] K. Crowston y J. Howison, “FLOSS Project Effectiveness Measures 1”, *Successful OSS Project Design and Implementation: Requirements, Tools, Social Designs and Reward Structures*, 2016, pp. 149.
- [27] M. Marco, *Free software contributions to improve traditional software management projects*, Barcelona: Universitat Oberta de Catalunya, 2005. Disponible en: [http://jpl.cpl.upc.edu/iii-jornades/ponencies/free-software-contributions-to-improve-traditional-software-management-projects/at\\_download/file](http://jpl.cpl.upc.edu/iii-jornades/ponencies/free-software-contributions-to-improve-traditional-software-management-projects/at_download/file)
- [28] A. Narduzzo y A. Rossi, *Modularity in action: GNU/Linux and free/open source software development model unleashed*, N.º 020. Department of Computer and Management Sciences, University of Trento, Italy, 2008.
- [29] L. Bilke, N. Watanabe, D. Naumov y O. Kolditz, “An integrated development workflow for community-driven FOSS-projects using continuous integration tools”, en *EGU General Assembly Conference Abstracts*, vol. 18, 2016, p. 9058.
- [30] E. Raymond, “The Cathedral and the Bazaar, Knowledge, Technology & Policy”, Fall99, vol. 12, Issue 3, pp. 23-49, 1999.
- [31] C. Lassenius, T. Dingsøyr y M. Paasivaara, eds., *Proceedings of Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland*, vol. 212. Springer, May 25-29, 2015.

- [32] D. Aycart et al., *Ingeniería del software en entornos de Software Libre*, Barcelona: Universitat Oberta de Catalunya, 2007, pp. 28-37.
- [33] K. Beck, *Embracing change with extreme programming*, Computer, Vol. 32, N.º 10, pp. 70-77, 1999.
- [34] K. Beck y M. Fowler, *Planning Extreme Programming*, Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 2000.
- [35] K. Beck y C. Andres, *Extreme Programming Explained: Embrace Change*, XP Series, Boston, MA: Addison -Wesley Longman Publishing Co., MA 2005.
- [36] A. Hameed, "Software Development Lifecycle for Extreme Programming", *International Journal of Information Technology and Electrical Engineering*, vol. 5, N.º 1, 2016.
- [37] M. Fowler, K. Beck, J. Brant, W. Opdyke y D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison Wesley Longman, Inc., 1999.
- [38] W. Lewis, *Software Testing and Continuous Quality Improvement*, CRC Press LLC, 2016.
- [39] G. Myers, C. Sandler y T. Badgett, *The art of software testing*, John Wiley & Sons, 2011.
- [40] R. Pressman, *Ingeniería de Software: un enfoque práctico*, 5.ª ed., Madrid: Editorial McGraw-Hill, 2004, pp. 55-66.
- [41] *Guía SBOK. Una guía para el conocimiento de Scrum*, SCRUMstudy, Edición, 2016.
- [42] G. Pantaleo y L. Rinaudo, *Ingeniería del Software*, Buenos Aires: Editorial Alfaomega, 2015, pp. 54-102.
- [43] I. Sommerville, *Ingeniería de software*, 7.ª ed., Addison Wesley, 2005.
- [44] G. Hertel y B. Orlikowski, "Project management in distributed virtual teams", en *Applied psychology for project managers*, Springer, Berlin: Heidelberg, 2015, pp. 305-321.
- [45] F. Bolici, J. Howison y K. Crowston, "Stigmergic coordination in FLOSS development teams: Integrating explicit and implicit mechanisms", *Cognitive Systems Research*, N.º 38, pp. 14-22, 2016.
- [46] K. Crowston et al., "Defining Open Source Project Success", en *Proceedings of the 24th International Conference on Information Systems (ICIS 2003)*, Seattle, Wash.
- [47] M. Martínez-Torres y M. D. Díaz-Fernandez, "Current issues and research trends on open-source software communities", *Technology Analysis & Strategic Management*, vol. 26, N.º 1, pp. 55-68, 2014.
- [48] K. Wei, K. Crowston, U.Y. Eseryel y R. Heckman, "Roles and politeness behavior in community-based free/libre open source software development", *Information & Management*, vol. 54, N.º 5, pp. 573-582, 2016.
- [49] Y. Cai y D. Zhu, "Reputation in an open source software community: Antecedents and impacts", *Decision Support Systems*, N.º 91, pp. 103-112, 2016.
- [50] N. Choi y J. A. Pruett, "The characteristics and motivations of library open source software developers: An empirical study", *Library & Information Science Research*, vol. 37, N.º 2, pp. 109-117, 2015.
- [51] M. Hilton, T. Tunnell, K. Huang, D. Marinov y D. Dig, "Usage, costs, and benefits of continuous integration in open-source projects", en *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ACM, 2016 pp. 426-437.
- [52] S. Sowe, A. Cerone y D. Settas, "An empirical study of FOSS developers patterns of contribution: Challenges for data linkage and analysis", *Science of Computer Programming*, N.º 91, pp. 249-265, 2014.
- [53] Y. Benkler, "Peer production and cooperation", *Handbook on the Economics of the Internet*, N.º 91, 2016.
- [54] K. Beck et al., *Manifiesto for Agile Software Development*, 2001. Disponible en: <http://agilemanifesto.org/>
- [55] J. Sutherland, *Agile Principles and Values*, [En línea], Microsoft, Docs, 2013. Disponible en: <https://msdn.microsoft.com/en-us/library/dd997578.aspx>
- [56] Reingart, M., "Desarrollo rápido de software libre de alta calidad", en *II Simposio Argentino sobre Tecnología y Sociedad (STS)-JAIIO 44*, Rosario, 2015.
- [57] H. Corbucci y A. Goldman, "Open Source and Agile: Two worlds that should have a closer interaction", *Instituto de Matemática e Estatística (IME)*, 2009.
- [58] S. Fellhofer, A. Harzl y W. Slany, "Scaling and Internationalizing an Agile FOSS Project: Lessons Learned", en *IFIP International Conference on Open Source Systems*, Springer, Cham, 2015, pp. 13-22.
- [59] K. Valerii, T. Do Cao y F. Oke, "Advantages of SCRUM in Open Source Software Development", *Open Source Software Development*, 2016.



# Capítulo IV

## REVISIÓN SISTEMÁTICA A LOS CONTENIDOS DIGITALES INTERACTIVOS ELABORADOS CON PRÁCTICAS ÁGILES

*Ricardo de J. Botero Tabares - rbotero@tdea.edu.co*

Profesor titular, Tecnológico de Antioquia

*Roberto Guevara Calume - rcgcalume@gmail.com*

Docente e investigador, Corporación Universitaria Uniremington

*Carlos Arturo Castro Castro carlos.castro@usbmed.edu.co*

Docente e investigador, Universidad de San Buenaventura

### I. INTRODUCCIÓN

Toda entidad o artefacto creado por los humanos conlleva contenido: las publicaciones impresas, los sitios web, los blogs, los cursos virtuales, el cine y la tv, el marketing, las noticias emitidas por diversos medios de comunicación y el arte; incluso la naturaleza misma tiene un contenido inmanente polifacético en sí mismo.

¿Qué es un contenido? ¿Cómo se clasifican los contenidos? ¿Qué hace interactivo un contenido? ¿Cuáles son las buenas prácticas que se publican en la literatura para estructurar, diseñar y evaluar contenidos interactivos? ¿Cuáles son las buenas prácticas que se publican en la literatura en relación con la interactividad y el agilismo? ¿En qué áreas o disciplinas aplican los diferentes autores la interactividad y/o el agilismo? ¿Cuáles son las buenas prácticas para el desarrollo de software? Estos interrogantes se plantean y responden en el presente artículo, a partir de una revisión de la literatura en la que se indagó por las buenas prácticas que se proponen para estructurar, diseñar y evaluar contenidos interactivos con prácticas ágiles.

Para responder los interrogantes planteados de una manera formal y rigurosa, en el año 2014 se inició la ejecución del proyecto de investigación “Estructuración, diseño y evaluación de contenidos interactivos a partir de prácticas ágiles – EDECIPA” [1], propuesto por investigadores de tres instituciones de educación superior de la ciudad de Medellín,

Colombia: el Tecnológico de Antioquia - Institución Universitaria, la Universidad de San Buenaventura y la Corporación Universitaria Remington. En este proyecto se profundizaron tres aspectos esenciales relacionados con los contenidos digitales: 1) Su clasificación o categorización, 2) Sus niveles de interactividad y 3) Las prácticas ágiles para producirlos.

La ejecución de EDECIPA conllevó actividades para identificar y reconocer el estado del arte alrededor de los contenidos digitales, la interactividad y las prácticas ágiles en diversos contextos, mediante una revisión sistemática de la literatura relacionada con los términos “contenido”, “contenido digital”, “interactividad y sus niveles”, “contenido interactivo” y “prácticas ágiles”. El proyecto permitió también especificar e integrar las buenas prácticas reconocidas en la estructuración y el diseño de contenidos interactivos, evaluar los resultados de uso de los contenidos diseñados desde la perspectiva de sus objetivos mediante una experimentación focalizada, y verificar y validar los contenidos diseñados.

El capítulo se divide en cinco secciones: en la primera se presenta la metodología empleada para la revisión sistemática; en la segunda se reseñan los trabajos previos relacionados con los términos de estudio; en la tercera se exponen los resultados a partir de las respuestas a las preguntas de investigación; en la cuarta se exponen las conclusiones, y en la quinta sección se presenta el trabajo futuro.



## II. METODOLOGÍA

La metodología de trabajo para determinar las publicaciones, estudios e informes relacionados con los contenidos digitales interactivos elaborados con prácticas ágiles, conllevó una revisión sistemática de la literatura dividida en tres etapas: 1) Planificación, 2) Realización y 3) Documentación [2]. Estas fases se combinan en una serie de procedimientos simples que establecen una revisión fiable de la literatura mediante un protocolo inicial en el que se estructura la metodología de la investigación [3], que consta de cinco etapas:

- A) Definir el área temática: Las buenas prácticas para el desarrollo de contenidos digitales interactivos, desde el punto de vista de la interactividad y el agilismo.
- B) Precisar las preguntas de investigación: P1. ¿Qué se entiende por “contenido” en general?, P2. ¿Se pueden identificar diferentes tipos de contenido?, P3. ¿Qué se entiende por interactividad?, P4. ¿Existen grados, niveles o categorías de interactividad?, P5. ¿Qué se entiende por “contenido interactivo”?, P6. ¿Cuáles son los contenidos interactivos que se publican en la literatura?, P7. ¿Cuáles son las buenas prácticas que se publican en la literatura en relación con la interactividad y el agilismo?, P8. ¿En qué áreas o disciplinas aplican los autores la interactividad y/o el agilismo? y P9. ¿Cuáles son las buenas prácticas que se proponen en la literatura para estructurar, diseñar y evaluar contenidos interactivos?
- C) Concretar el proceso de búsqueda: La búsqueda se realizó en castellano e inglés con los términos Interactividad (interactivity), contenidos interactivos (interactive content), escalas de interactividad (scales interactivity), recursos multimedia (multimedia resources), buenas prácticas (good practices), mejores prácticas (best practices), prácticas ágiles (agile practices), agilismo (agile), diseño interactivo (interactive design) e implementación interactiva (interactive implementation). Las bases de datos consultadas fueron IEEEExplore, ACM Digital Library, ScienceDirect, Elsevier, SpringerLink, Scopus, Web of Science, DOAJ y CiteSeerX.
- D) Definir los criterios de inclusión y exclusión: Para incluir un artículo en el análisis debe haber sido revisado por pares, estar disponible en línea, escrito en inglés o castellano, e informar sobre la definición de interactividad o los niveles de interactividad. Los documentos se clasificaron por tres criterios: 1) mediante la lectura del título y el resumen; 2) se excluyeron todos los documentos que no coincidían claramente con los criterios expuestos; 3) los otros fueron analizados con mayor cuidado involucrando la valoración de la calidad.
- E) Precisar la valoración de la calidad: Se realizó una búsqueda de artículos, libros, reportes técnicos, experiencias y trabajos empíricos, teóricos y experimentales que hubieran pasado por una revisión de pares. Otros criterios para validar la calidad fueron: Calidad de la fuente (factor de impacto), Resultados verificables, Nivel de aceptación, Trayectoria del autor, Verificación de la aplicación y Relevancia del trabajo de acuerdo con las citaciones.

## III. TRABAJOS RELACIONADOS

La revisión sistemática conllevó búsqueda de literatura en bases de datos científicas para los focos de interés del proyecto: interactividad y sus niveles, contenidos interactivos y prácticas ágiles.

### A. Interactividad

Todo cuanto aparece acompañado del adjetivo “interactivo” adquiere un valor añadido por sí mismo y lo hace máspreciado e innovador para quien lo utiliza [4]; la cualidad de “interactivo” aporta un valor especial a un objeto, ya que este adquiere cierta capacidad de diálogo con el usuario.

La interactividad se usa en muchos contextos, es un vocablo de moda, un adjetivo que se escucha y se usa indiscriminadamente; sin embargo, mucha gente no sabe realmente cuál es su esencia, llegando a interpretaciones erróneas o inexactas.

Si bien existe una definición en el Diccionario lengua española [5] para el término interactividad, como “Cualidad de interactivo”, que a su vez define



interactivo, como “1. Que procede por interacción” y “2. Dicho de un programa: Que permite una interacción, a modo de diálogo, entre el ordenador y el usuario”, ambas son definiciones ambiguas para nuestro propósito, máxime si se tiene la intención de evaluar qué tan interactivo puede ser un artefacto como un programa de computador, un video, una plataforma computacional, un curso de educación virtual o, en forma general, un contenido digital.

La revisión de la literatura demuestra que existen diferentes definiciones de interactividad dependiendo del campo relacionado: ciencias de la computación, interacción humano-computador, (HCI), comunicaciones, arte, entretenimiento (juegos), publicidad o *marketing* [6]. Por ejemplo, se han llegado a definir hasta 186 técnicas interactivas de evaluación de aula agrupadas en 19 tipos, que incluyen acciones del instructor y de los estudiantes, redes sociales y juegos e interacción en línea, entre otros [7]. Se ha planteado que el e-learning es un componente vital para expandir los límites educativos mediante el desarrollo de contenidos realistas e inmersivos [8]. Se ha explorado la interacción con el contenido en el contexto de los cines 3D mediante un teléfono móvil [9]. Se ha descrito un sistema de edición para música interactiva usando el estándar MPEG-A IM AF (Formato de Solicitud de Música Interactiva), que proporciona una interactividad al usuario con los contenidos musicales, en la que el usuario puede recomponer el contenido de la música según su preferencia [10]. Otros trabajos proponen un marco de dispositivo de interfaz humana para aplicaciones interactivas, que incluyen contenidos educativos en entornos ubicuos mediante el uso combinado del marco WebIB (sistema de desarrollo de software para los contenidos Web 3D que se ejecutan en Internet y TV Digital) [11], y se presenta un decálogo de aspectos a los cuales denomina “desiderata”, para la comunicación interactiva humano – robot, cubriendo aspectos verbales y no verbales [12].

Existen varios trabajos que atienden problemas y necesidades del *marketing* y especialmente del periodismo, y que definen desde estas áreas la interactividad [13], [14], [15], [16].

Además de los trabajos mencionados, que hacen un acercamiento a la definición de los niveles de interacción, existen otros que permiten hacer un recorrido por temas relacionados con la comunicación personal [17], que plantean una división de cinco grados en la comunicación.

Una forma de facilitar la interacción puede ser el hipertexto, que tal vez constituye la forma más simple de interactividad. El hipertexto obliga a la lectura en red y posibilita la lectura no secuencial, y para esto es imprescindible la existencia de tres elementos inherentes: nodos, enlaces y anclajes [18].

## B. Niveles de Interactividad

En cuanto a los niveles de interactividad, se han publicado varios trabajos para catalogarlos, pero sigue existiendo poca orientación práctica que permita clasificar los contenidos digitales según su grado de interactividad. Entre estos trabajos se encuentran los experimentos para estudiar la interactividad y la vivacidad de los sitios web comerciales [19], que han hallado conceptualizaciones desde una perspectiva de la comunicación interpersonal. En este mismo trabajo se define la interactividad como la forma en la cual los mensajes de una secuencia se relacionan entre sí, y sobre todo, la forma como los mensajes posteriores son influenciados por mensajes anteriores.

Por otra parte, se definen elementos básicos para determinar el efecto de los diferentes niveles de interactividad en el aprendizaje de los estudiantes de tecnología [20]. A su vez, se encuentran aportes que buscan diseñar materiales de apoyo, en tal sentido existe el estándar IEEE 1484.12.3 para Learning Technology [21], el cual propone una categorización, pero se limita a mencionarlas sin dar criterios claros que permitan evaluar un contenido interactivo.

Existen varias aproximaciones sobre los niveles de interactividad empleados para evaluar, desde una guía para desarrolladores y diseñadores de cursos e-learning de calidad [22], hasta blogs especializados como el e-learningWeeky [23]. Otro aporte viene del Departamento de Defensa de Estados Unidos [24], donde han desarrollado una guía que proporciona información básica para la planificación, diseño,

desarrollo, implementación, evaluación y gestión de productos Interactivos Multimedia para la Instrucción (IMI), como texto, tecnologías de audio, terapia de voz, elementos visuales, videos, gráficos y animación. También se ha definido la evaluación de IMI: Interactive Multimedia Instruction Evaluation (IMIE). Para la estructuración, diseño y evaluación de contenidos interactivos con prácticas ágiles, se presenta una revisión de los niveles de interactividad de los contenidos digitales [25].

### C. Contenidos Interactivos

Un contenido interactivo ofrece información personalizable en función de las necesidades y exigencias de los usuarios [26].

Son muchos los trabajos relacionados con la temática de contenidos que aportan de una u otra manera a la categorización de los contenidos interactivos. En el contexto de la cartografía geográfica, se propone una arquitectura de software para la creación dinámica de objetos visuales en aplicaciones para internet con contenidos interactivos [27]. Para la difusión de multimedia digital terrestre se presenta un software reproductor de contenido interactivo bidireccional, que contiene un sistema de edición equipado con una potente interfaz gráfica que permite crear y editar contenido interactivo [28]; para la codificación automática de contenido web sobre Televisión Digital Interactiva (IDTV), se propone un sistema que incluye una aplicación cliente para navegar por el contenido codificado en un televisor mediante el control remoto [29], además se propone un conjunto de nuevas formas o categorizaciones de ITV [30]. Se plantean nuevos microformatos y esquemas de microdatos que permiten una descripción básica de objetos multimedia para crear contenidos 3D interactivos enfocados a la web [31]. Se describen las herramientas XMT que pueden generar, manipular y traducir un documento XML para la radiodifusión interactiva, e introducen un sistema de edición basado en las herramientas XMT proporcionadas [32]. Un contenido interactivo se desarrolla para el curso “Fundamentos de la tecnología de la información” de una escuela de secundaria de Estambul, Turquía [33]. Para producir programas de difusión interactiva, se presenta un sistema de edición basado en el contenido, mediante el uso de

las tecnologías MPEG-4 [34]. Para la estructuración, diseño y evaluación de contenidos interactivos con prácticas ágiles, se presenta una categorización de los contenidos interactivos [35].

### D. Prácticas Ágiles

Algunos autores han realizado diferentes investigaciones para descubrir cómo se complementan las buenas prácticas del agilismo y las de la interactividad en el diseño de contenidos interactivos, aunque por tradición estas dos metodologías utilizan diferentes enfoques en un proyecto con respecto a la asignación de recursos, tiempos, atención a los requisitos y prototipos [36]. El agilismo se esfuerza por ofrecerles a los clientes pequeños conjuntos de funciones lo más veloces posible y en iteraciones cortas, mientras que la interactividad invierte un esfuerzo considerable en la investigación y el análisis antes de comenzar el desarrollo. Hay una creciente literatura en la que se trata de combinar las buenas prácticas desde el agilismo y la interactividad en el diseño de contenidos interactivos [37], [38], sin embargo, esta integración no se aborda adecuadamente.

Se ha descrito una integración de eXtreme Programming (XP) [39] y User-Centered Design (UCD) relacionada en una aplicación de multimedia para teléfonos móviles [40]. También se ha presentado un marco que se puede utilizar para integrar las prácticas del diseño interactivo y el desarrollo ágil, donde se sugieren cinco principios para una integración exitosa: 1) participación del usuario, 2) colaboración y cultura, 3) desarrollo de prototipos, 4) proyecto de ciclo de vida, y 5) gestión de proyectos [41].

Un estudio cualitativo de proyectos ágiles reales puede implicar el diseño interactivo [42]. Algunos resultados conllevan que la naturaleza del desarrollo iterativo facilita la realización de las pruebas de usabilidad, lo que les permite a los desarrolladores incorporar los resultados de estas pruebas en iteraciones posteriores. Además, esto también puede mejorar significativamente la comunicación y la relación entre los equipos de trabajo. En este sentido, el problema de lograr la integración de las buenas prácticas del agilismo y de la interactividad es que ambos equipos tienen que contribuir con sus habilidades al desarrollo

del proyecto, lo que normalmente se ha caracterizado como una fusión entre un método ágil con un método de diseño User eXperience (UX) que, de acuerdo con los mismos autores, no es fácil de alcanzar [43].

Se ha encontrado que las prácticas de los métodos ágiles de diseño centrados en el usuario, generan productos mejor diseñados que aquellos producidos al aplicar las versiones en cascada de las mismas técnicas [44]. Además, se ha sugerido que para integrar las buenas prácticas de ambos modelos es necesario que los diseñadores de interactividad comprendan mejor los principios del agilismo, y se presentan algunas prácticas para la integración de estos dos campos [45].

#### IV. RESULTADOS

Las respuestas a las diferentes preguntas de investigación fueron las siguientes:

##### A. ¿Qué se Entiende por “Contenido” en General?

El término “contenido” tuvo origen en el marketing y se expandió en la sociedad como algo omnipresente, porque se encuentra en todos los medios de comunicación, y se podría definir como la información que contiene una obra (como un libro o una escultura), una publicación (noticia en un diario, cartel de invitación a un concierto) o un mensaje (telefónico, de correo electrónico). En la producción y publicación de medios, el contenido es la información y las experiencias que proporcionan valor para un usuario final o un público objetivo [46]. El contenido es algo que se va a difundir a través de algún medio, como el habla, la escritura, o cualquiera de las diferentes artes y tecnologías [47].

##### B. ¿Se Pueden Identificar Diferentes Tipos de Contenido?

Existen varios enfoques para la clasificación de los tipos de contenido: 1) Contenidos enfocados a internet [48]; 2) Contenidos de acuerdo con su propósito específico [49]; 3) Contenidos inteligentes [50]; 4) Contenidos por el sentimiento que genera en las personas [51]; y 5) Contenidos educativos [52], [53].

##### C. ¿Qué se Entiende por Interactividad?

Interactividad es una palabra empleada en forma transversal en muchas áreas del conocimiento, como el periodismo, la aeronáutica, los sistemas informáticos, el marketing, la televisión, los juegos y la museología. Este término adquiere notoriedad a partir de la década de 1980 con la divulgación y expansión de las Tecnologías de la Información y la Comunicación (TIC).

En general, la interactividad implica: a) Intervención por parte del usuario sobre el contenido. b) Transformación del espectador en actor. c) Diálogo individualizado con los servicios conectados. d) Acciones recíprocas en modo dialógico con los usuarios, o en tiempo real con los aparatos [54].

##### D. ¿Existen Grados, Niveles o Categorías de Interactividad?

Se han planteado varios niveles de interactividad: auténtica, media y simulada [55]; implícita, explícita y de enunciado independiente [46]. También se presenta otra clasificación, dada por el Departamento de Defensa de Estados Unidos [56], que establece cuatro niveles de interactividad: pasiva, limitada, compleja y en tiempo real, descritos de la siguiente manera:

- **Nivel I (Pasivo):** El usuario actúa como un simple receptor de información. Puede leer texto en la pantalla, así como ver gráficos o imágenes. Logra interactuar simplemente usando los botones de navegación para desplazarse hacia adelante o atrás en el programa, o puede navegar por los vínculos del hipertexto.
- **Nivel II (Interacción limitada):** El usuario puede dar respuestas simples según las indicaciones de la instrucción.
- **Nivel III (Interacción compleja):** El usuario realiza múltiples acciones como respuestas a las instrucciones. Se posibilita la entrada de cuadros de texto y la manipulación de objetos gráficos para probar la evaluación de la información presentada.
- **Nivel IV (Interacción en tiempo real):** El usuario participa en una simulación que refleja exactamente la situación de trabajo.

### E. ¿Qué se Entiende por Contenido Interactivo?

Un contenido interactivo ofrece información personalizable en función de las necesidades y exigencias de los usuarios [57], y conlleva las siguientes características que aumentan los niveles cognitivos o posibilidades de aprendizaje y comprensión de los usuarios:

- El flujo de información en múltiples sentidos (información en varias direcciones).
- El rol activo del usuario en la selección de la información requerida.
- Una interfaz enriquecida con elementos multimedia que mejora el nivel de comunicación.

Para que la información final sea de calidad, práctica y transparente, un departamento de contenidos interactivos debe ofrecer una gama de servicios añadidos tales como:

- Búsqueda, clasificación y digitalización de la información.
- Documentación y referencias para proyectos.
- Mantenimiento y actualización de contenidos.
- Traducción.

Estos servicios se deben conjugar con características esenciales que les aporten calidad a los contenidos interactivos: actualidad, profundidad, originalidad y fiabilidad.

### F. ¿Cuáles son los Contenidos Interactivos que se Publican en la Literatura?

Son múltiples los contenidos interactivos que se publican en la literatura según lo especifican varios autores [58], [59], [60]. Los principales incluyen: el Periodismo (prensa en línea), Educación virtual, Educación con entornos ubicuos, Marketing, Entretenimiento, Reconocimiento de patrones, Aprendizaje automático, Multimedia digital terrestre, TV digital interactiva, Cine 3D interactivo, Música interactiva, Comunicación humano-computador y Robótica.

### G. ¿Cuáles son las Buenas Prácticas que se Publican en la Literatura en Relación con la Interactividad y el Agilismo?

Las tablas 1 y 2 [61] presentan resúmenes de las buenas prácticas de uso general identificadas desde la interactividad y el agilismo, respectivamente.

**Tabla 1.** Buenas Prácticas desde la Interactividad

Nº	Descripción de la práctica
1	Integrar todas las partes interesadas al equipo de trabajo.
2	Modelar y simular.
3	Determinar objetivos claros y alcanzables de acuerdo con el área de aplicación.
4	Motivar adecuadamente la participación de los actores.
5	Contar con una adecuada infraestructura de trabajo.
6	Tener en cuenta adecuados niveles de comunicación de acuerdo con la audiencia.
7	Implementar procesos permanentes de retroalimentación.
8	Diseñar paralelamente el sistema de evaluación y simularlo.
9	Evaluar el logro objetivo.
10	Diseñar para cada audiencia.
11	No universalizar la interactividad.
12	Reconocer los actores
13	Analizar las experiencias previas.
14	Utilizar siempre tecnología de punta y más allá.
15	Visionar posibles escenarios futuros.
16	Determinar y hacer cumplir la vida útil del diseño.
17	Retirar a tiempo.

**Tabla 2.** Buenas Prácticas desde el Agilismo

Nº	Descripción de la práctica
1	Documentar toda actividad, fase y procedimiento.
2	Todos los procesos deben ser iterativos.
3	Prototipar.
4	Reconocer y aprovechar las habilidades, destrezas, capacidades, y conocimientos de cada integrante del equipo.
5	Diseñar a partir de historias de usuarios.
6	Estructurar y ejecutar pruebas paralelas al desarrollo.

Nº	Descripción de la práctica
7	Validar los requisitos, las interfaces, y los diseños mediante la observación, la medición, y el registro
8	Diseñar y desarrollar lo más integral que se pueda.
9	Construir escenarios.
10	Refactorar.
11	Simular los tiempos de entrega.
12	Entregables con ciclos de desarrollo cortos.
13	Mantener motivado al equipo de trabajo.
14	Medir continuamente el progreso del proyecto.
15	Respetar los estándares del área específica.
16	Integrar a todos los actores desde las etapas tempranas.
17	Reconocer cada error y documentarlo.
18	Mantener y fomentar la comunicación de doble vía.

#### H. ¿En qué Áreas o Disciplinas Aplican los Autores la Interactividad y/o el Agilismo?

La interactividad y el agilismo se aplican en variadas disciplinas; por ejemplo, la primera se aplica en la realidad virtual, el diseño y la animación 3D, la multimedia y el e-learning, las cuales connotan desarrollo de software; la segunda se aplica en el desarrollo y en la ingeniería de software. Un resumen de dicha aplicabilidad se presenta en la Tabla 3 [61].

**Tabla 3.** Áreas o Disciplinas de Aplicación de la Interactividad y el Agilismo

Interactividad	Agilismo
<ul style="list-style-type: none"> <li>– Realidad virtual</li> <li>– Diseño y animación 3D</li> <li>– Multimedia</li> <li>– Cine y TV</li> <li>– e-Learning</li> <li>– Educación presencial, virtual, y a distancia</li> <li>– Marketing</li> <li>– Publicidad</li> <li>– Astronomía</li> <li>– Física</li> <li>– Paleontología</li> <li>– Medicina</li> <li>– Ingeniería</li> </ul>	<ul style="list-style-type: none"> <li>– Desarrollo de software</li> <li>– Gestión y administración organizacional</li> <li>– Gestión de proyectos</li> <li>– Inteligencia de negocios</li> <li>– Planes de estudios</li> <li>– Ingeniería industrial</li> <li>– Ingeniería de software</li> <li>– Mercadeo</li> </ul>

#### I. ¿Cuáles son las Buenas Prácticas que se Proponen en la Literatura para Estructurar, Diseñar y Evaluar Contenidos Interactivos?

Aunque no se encontraron trabajos específicos que describieran buenas prácticas para estructurar, diseñar y evaluar contenidos interactivos, en las tablas 4, 5 y 6 se listan las que tienen alguna relación con este propósito [61].

**Tabla 4.** Buenas Prácticas para Estructurar Contenidos Interactivos

Nº	Descripción de la práctica de estructuración
1	A partir de prototipos.
2	Desde escenarios factibles.
3	Comenzando con historias de usuario.
4	Teniendo en cuenta el desarrollo socio-tecnológico actual.
5	Reconociendo al usuario final.
6	En equipo.
7	De acuerdo con el público objetivo.
8	No universal.
9	Documentando todo el proceso.
10	Con tecnología de punta.
11	Respetando los estándares y normas de cada área.
12	Teniendo en cuenta los costos.

**Tabla 5.** Buenas Prácticas para Diseñar Contenidos Interactivos

Nº	Descripción de la práctica de diseño
1	A partir de prototipos modelados, validados, y simulados.
2	Simulando y modelando desde escenarios factibles.
3	Teniendo en cuenta los factores de usabilidad.
4	Con experimentos.
5	Haciendo inmersión de la audiencia.
6	En equipo.
7	Centrado en el usuario.
8	Documentando todo el proceso.
9	Con tecnología de punta.



Nº	Descripción de la práctica de diseño
10	Teniendo en cuenta los estados de ánimo de la audiencia.
11	Manteniendo un ritmo constante.

**Tabla 6.** Buenas Prácticas para Evaluar Contenidos Interactivos

Nº	Descripción de la práctica de evaluación
1	De forma iterativa.
2	Los logros instruccionales.
3	Resultados vs historias de usuarios.
4	De forma individual y grupal.
5	Documentando todo el proceso.
6	Por fuera de los modelos tradicionales.
7	Centrado en el logro.
8	Con proyección futura.
9	Pensando en el mejoramiento continuo.
10	De forma ágil y objetiva.
11	Con retroalimentación.

## V. CONCLUSIONES

Llegar a un acuerdo sobre la definición de interactividad que beneficie a cualquier área del conocimiento, no parece ser algo viable, ya que cada área emplea la interactividad de una forma particular en función de sus propias necesidades, e incluso se le parametriza de formas diferentes.

El uso de la interactividad como una variable en las investigaciones ha aumentado con la aparición de nuevos canales de comunicación. Además, aunque el agilismo ha tenido mayor aplicación y difusión desde el desarrollo de software, se encontraron otras áreas y propuestas en las que los investigadores están haciendo uso de sus prácticas.

Los autores presentan trabajos relacionados en los que se describen buenas prácticas desde la interactividad y el agilismo para desarrollar planes y proyectos de mercadeo, de gestión y administración, y para estructurar, diseñar y evaluar productos desde varias áreas o disciplinas de aplicación.

Se encontraron varios tipos de clasificación de contenidos: educativos, enfocados a internet, según el

marketing digital, según el Modelo de Puntos de Contacto y por el sentimiento que genera en las personas, lo cual conlleva una clasificación extensa desde diferentes puntos de vista. También se identificaron varios niveles de interactividad, donde los más evidentes fueron cuatro: pasivo, limitado, complejo y de interacción en tiempo real; este último constituye el más profundo nivel de interactividad y puede conllevar la inmersión en mundos de realidad virtual.

Un contenido interactivo ofrece información personalizable en función de las necesidades y exigencias de los usuarios y conlleva características que aumentan sus niveles cognitivos y su comprensión.

Los contenidos interactivos se publican en áreas como el periodismo, educación, marketing, cine, TV, entretenimiento, comunicación humano-computador y robótica.

En la literatura se difunden de manera amplia las buenas prácticas desde la interactividad y el agilismo. Además, se encontró que no era fácil una definición para buenas prácticas que tuviera amplia aceptación, sin embargo, se pudo responder a cada una de las preguntas planteadas en la investigación, al tiempo de lograr una descripción de las buenas prácticas difundidas para estructurar, diseñar y evaluar contenidos interactivos.

Finalmente, la integración de las buenas prácticas desde la interactividad y el agilismo exige diferentes niveles de planificación y colaboración entre investigadores en ambas áreas del conocimiento.

## VI. TRABAJO FUTURO

El trabajo futuro e inmediato que conlleva el proyecto “Estructuración, diseño y evaluación de contenidos interactivos a partir de prácticas ágiles” se dará en las instituciones de educación superior que lo financiaron, a partir de la divulgación del concepto de la interactividad, con sus niveles y ventajas, en cursos donde se manejen interfaces humano-computador e interfaces gráficas de usuario. Este trabajo de carácter curricular implicará promover el uso de la interactividad y el agilismo con sus buenas prácticas a lo largo del pènsum académico de diferentes programas del área informática y en el desarrollo de software para los trabajos de grado.



Finalmente, se debe presentar al agilismo como un principio que puede ser utilizado en muchas áreas de producción y desarrollo, para implementar buenas prácticas generales en la estructuración, diseño y evaluación de contenidos interactivos. Estos aportes en diversas áreas de conocimiento potencializarán los proyectos que se lleven a cabo, orientados a visibilizar el diseño de contenidos interactivos combinando de manera eficaz la interactividad y el agilismo.

## REFERENCIAS

- [1] C. Castro, R. Botero, y C. Guevara, “Estructuración, diseño y evaluación de contenidos interactivos a partir de prácticas ágiles. Proyecto de investigación interinstitucional: U. de San Buenaventura Medellín, Tecnológico de Antioquia Institución Universitaria, Corporación Universitaria Remington”, 2014.
- [2] M. Turner, P. Brereton, B. Kitchenham, D. Budgen y M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain”, *Journal of Systems and Software*, vol. 4, N.º 4, pp. 571-583, 2007.
- [3] M. E. Serna, Methodology for perform reliable literature reviews, en prensa, 2014.
- [4] M. E. Minguell, “Interactividad e interacción”, *Revista Interuniversitaria de Tecnología Educativa*, N.º 0, pp. 92-97, 2000.
- [5] RAE, Versión electrónica de la 23ª edición del “Diccionario de la lengua española”, 2014. Disponible en: <http://www.rae.es/diccionario-de-la-lengua-espanola/la-23a-edicion-2014>
- [6] M. Carter y M. Donovan, Levels of Interactivity and Determining What’s Appropriate. *The E-learning Guild*, 2007.
- [7] T. A. Angelo y K. P. Cross, *Classroom Assessment Techniques*, 2nd Edition. San Francisco: Jossey-Bass, 1993.
- [8] I. Haddish, *Next Generation E-education: Bringing Fully- Interactive Content Into University Labs & Classrooms*, 2012.
- [9] M. Hakila et al., “Let me catch this! Experiencing Interactive 3D Cinema through Collecting Content with a Mobile Phone”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1011-1020, 2014.
- [10] I. Jang et al., *Interactive Music Contents Authoring System Based on IM AF*. *Consumer Electronics (ICCE)*, 2011 IEEE International Conference on, 2011, pp. 833-834.
- [11] K. Kaneko et al., “Open Device Control (OpenDC): Human Interface Device Framework for Interactive Applications Including Educational Contents in Ubiquitous Environments. *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE)*, 2012 IEEE Seventh International Conference on, 2012, pp. 122-126.
- [12] N. Mavridis, *A review of verbal and non-verbal human-robot interactive communication*, Cornell University Library, 2014.
- [13] M. Cebrain, “Comunicación interactiva en los Cibermedios”, *Revista Científica de Educación*, vol. XVII, N.º 33, pp. 16-24, 2009.
- [14] M. A. Cabrera, “La interactividad de las audiencias en entornos de convergencia digital”, *Revista ICONO* 14, N.º 15, pp. 164-177, *Revista ICONO*.
- [15] L. Navarro, “La interactividad en los géneros periodísticos de los cibermedios”, *Razón y Palabra*, vol. 18, N.º 84, 2013.
- [16] A. Rost, *La interactividad en el periódico digital*, Barcelona: Univesitat Autònoma de Barcelona, 2006.
- [17] J. M. Perceval y S. Tejedor, “Los cinco grados de la comunicación en educación”, *Revista Científica de Comunicación y Educación*, vol. 15, N.º 30, pp. 155-163, 2008.
- [18] M. A. Martí, *Tecnologías del lenguaje*, Barcelona: Editorial UOC, 2003.
- [19] J. Coyle y E. Thorson, *The Effects of Progressive Levels of interactivity and Vividness in Web Marketing Sites*, *Journal of Advertising*, vol. XXX, N.º 3, pp. 65 77, 2001.
- [20] S. K. Chand, Z. Przasnyski y L. Leon, “How Levels of Interactivity in Tutorials Affect Students’ Learning of Modeling Transportation Problems in a Spreadsheet”, *Decision Sciences Journal of Innovative Education*, vol. 8, N.º 1, 2010.
- [21] IEEE Computer Society, *IEEE Standard for Learning Technology-Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata*, New York: IEEE, 2005.
- [22] CDC, *A guide for creating quality electronic learning*. Atlanta: CDC eLearning Workgroup, 2013.
- [23] B. J. Schone, *Engaging Interactions for eLearning*. Ebook, 2007.
- [24] USDOD - U.S. Department of Defense, *Development of interactive multimedia instruction (IMI)*, (Part 3 of 4 parts), MIL-HDBK-29612-3, 1999.

- [25] R. Guevara, R. Botero y C. Castro, “Una revisión a los niveles de interactividad de los contenidos digitales”, Memorias del XX Congreso Internacional de Informática Educativa – TISE 2015, Volumen 11, Chile.
- [26] N. Lloret y F. Canet, Nuevos escenarios, nuevas formas de expresión narrativa: La Web 2.0 y el lenguaje audiovisual. Anuario Hipertext.net, número 6 [En línea], Barcelona: Universitat Pompeu Fabra, 2008. Disponible en: <http://www.upf.edu/hipertextnet/numero-6/lenguaje-audiovisual.html> [Accedido: 08-09-2015].
- [27] A. Bekkari, D. Mammass y S. Idbraim, “Software Architecture for the Dynamic Creation of Visual Objects in Applications for the Internet with Interactive Content”, International Conference on Multimedia Computing and Systems, ICMCS '09, 2009.
- [28] W. Cheong, J. Cha, M. Ki y K.A. Moon, “Bidirectional interactive contents player for terrestrial digital multimedia broadcasting”, IEEE Transactions on Consumer Electronic, Vol. 54, Issue 1, pp. 93-99, 2008.
- [29] S. Ferretti y M. Rocchetti, “MHP Meets the Web: Bringing Web Contents to Digital TV for Interactive Entertainment”, Eighth IEEE International Symposium on Multimedia, 2006. ISM'06, pp. 169-176.
- [30] J. F. Jensen, “Interactive Television: New Genres, New Format, New Content”, Proceedings of the second Australasian conference on Interactive entertainment, 2005, pp. 89-96.
- [31] J. Flotyński y K. Walczak, “Microformat and Microdata Schemas for Interactive 3D Web Content”, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, 2013, pp. 549-556.
- [32] Y. Joung, H. Kim y K. Kim, “XMT tools for interactive broadcasting contents description”, Proceedings of the Third International Workshop on Digital and Computational Video, 2002. DCV 2002.
- [33] D. Karahocaa et al., “Interactive e-content development for vocational and technical education”, Procedia - Social and Behavioral Sciences, vol. 2, Issue 2, pp. 5842-5849, 2010.
- [34] K. Kim, I. Lee y M. Ki, “Interactive Contents Authoring System based on XMT and BIFS”, Proceedings of the 10th ACM International Conference on Multimedia, Juan les Pins, France, 2002.
- [35] C. Castro, R. Botero y R. Guevara, “Categorización de los contenidos interactivos”, Memorias del XX Congreso Internacional de Informática Educativa – TISE 2015, Vol. 11, Chile.
- [36] D. Fox, J. Sillito y F. Maurer, “Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry”, Proceedings Agile conference, Washington, USA, 2008, pp. 63-72.
- [37] J. Ferreira, H. Sharp y H. Robinson, “Values and assumptions shaping agile development and user experience design in practice”, en Agile Processes in Software Engineering and Extreme Programming. XP 2010: International Conference on Agile Software development, A. Sillitti, A. Martin, X. Wang y E. Whitworth, eds., Springer, Berlin: Heidelberg, 2010, pp. 178-183.
- [38] T. Silva et al., “User-centered design and agile methods: A systematic review”, Proceedings Agile Conference, August 7-13, Salt Lake City, USA, 2011, pp. 77-86..
- [39] Z. Hussain et al., “Integration of Extreme Programming and User-Centered Design: Lessons Learned”, en Agile Processes in Software Engineering and Extreme Programming 2009, Pula, 2009, pp. 174-179.
- [40] S. Chamberlain, H. Sharp y N. Maiden, “Towards a framework for integrating agile development and user-centred design” en Extreme Programming and Agile Processes in Software Engineering, 2006, pp. 143-153.
- [41] Z. Hussain, W. Slany y A. Holzinger, “Current state of agile user-centered design: A survey” en HCI and Usability for e-Inclusion, 2009, pp. 416-427.
- [42] J. Ferreira, J. Noble y R. Biddle, “Agile development iterations and UI design”. Proceedings Agile Conference, August 13-17, Washington, USA, 2007, pp. 50-58.
- [43] J. Ferreira, H. Sharp y H. Robinson, “User experience design and agile development: Managing cooperation through articulation work”, Software-Practice & Experience, vol. 41, N.º 9, pp. 963-974, 2011.
- [44] D. Sy, “Adapting usability investigations for agile user-centered design”, Journal of Usability Studies, vol. 2, N.º 3, pp. 112-132, 2007.
- [45] J. Carroll y H. Beyer, Contextual design for agile teams (Synthesis Lectures on Human-Centered Informatics), Morgan & Claypool Publishers, 2010.
- [46] M. G. Moore, “Three types of interaction”, American Journal of Distance Education, vol. 3, N.º 2, pp. 1-6, 1989.

- [47] C. Ammer, "Content", The American Heritage Idioms Dictionary. Online [Accedido Julio 2015].
- [48] M. Quesada, ed., Internet como fuente generadora de contenidos especializados. IX Encuentro de profesores de periodismo especializado, Barcelona, 2009.
- [49] M. D. Castellano, "Aspectos pedagógicos del diseño de contenidos digitales interactivos, consideraciones sobre el proceso de mediación", Revista Virtual Universidad Católica del Norte, N.º 49, 51-71, 2016.
- [50] The Rockley Group, Inc., What is Intelligent Content? The Content Wrangler. [En línea], 2008. [Accedido Junio 2016].
- [51] S. Aughtmon, 21 Types of Content We Crave. Content Marketing Institute. [En línea]. 2012 [Accedido Abril 2015]
- [52] J. Boneu, "Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos", Revista de Universidad y Sociedad del Conocimiento, Vol. 4, N.º 1, 2007.
- [53] T. Sharon y A. J. Frank, "Digital libraries on the Internet. Jerusalem (Israel): Proceedings of the 66th IFLA Council and General Conference.
- [54] R. Aparici y M. Silva, "Pedagogía de la interactividad", Comunicar. Revista Científica de Comunicación y Educación, 2012.
- [55] R. Bretz y M. Schmidbauer, Media for Interactive communication, California: Sage Publications, 1983.
- [56] R. Schwier y F. Misanchuk, Interactive Multimedia Instruction, Englewood Cliffs, New Jersey: Educational Technology Publications Inc., 1993.
- [57] N. Lloret, "Visión general de los contenidos en Internet", Hipertext.net, [En línea], N.º 1, 2003. [Accedido Julio 2015].
- [58] P. Rojo, La empresa periodística multimedia: creación y difusión digital de contenidos interactivos, Madrid: Siranda Editorial, 2005.
- [59] J. Tramullas, "Herramientas de software libre para la gestión de contenidos", Hipertext.net, [En línea], N.º 1, 2005 [Accedido Junio 2015].
- [60] F. García, "Contenidos educativos digitales: Construyendo la Sociedad del conocimiento", Revista de Tecnologías de la Información y Comunicación Educativas, N.º 6, 2006, CNICE, España.
- [61] M. E. Serna, ¿Cuál es el nivel de divulgación de buenas prácticas desde la interactividad y el agilismo?, 2015, en prensa.



## Capítulo V

# GESTIÓN DE RIESGOS ASOCIADOS A LAS PERSONAS EN PROYECTOS CON SCRUM: UNA REVISIÓN BIBLIOGRÁFICA

*Gloria Yaned Tobón Cataño - gytobon@elpoli.edu.co*

Politécnico Colombiano Jaime Isaza Cadavid

*Mónica María Rojas Rincón - mmrojas@elpoli.edu.co*

Politécnico Colombiano Jaime Isaza Cadavid

*Adriana Xiomara Reyes Gamboa - axreyes@elpoli.edu.co*

Politécnico Colombiano Jaime Isaza Cadavid

## I. INTRODUCCIÓN

Las fábricas de software de pequeño o gran tamaño se ven ante la necesidad de definir una estrategia que les permita responder a las expectativas del cliente, sin afectar las variables de tiempo, costo y alcance. Es así como las metodologías de gestión de proyectos tienen la planeación como eje esencial de control. No obstante, algunas veces se presenta una pobre gestión de riesgos, o la identificación detallada de éstos en etapas en las cuales resulta no ser efectiva. Es necesario entonces tener presente que los proyectos son ejecutados por personas que positiva o negativamente terminan afectando el resultado final del producto en cuestión.

La relación recíproca entre las metodologías de gestión de proyectos y de productos, junto a la gestión de riesgos asociados a las personas, pueden ser la clave para anticiparse a situaciones futuras que podrían afectar el resultado final del proyecto. Para esto es necesario no solo comprender las metodologías, sino integrarlas en el momento justo.

Existen en la actualidad diversas metodologías de desarrollo de software. Según Navarro et al. [1],

... las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías.

Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos —no predictivos—. La segunda diferencia es que las metodologías ágiles son orientadas a las personas —no orientadas a los procesos.

Esta orientación a las personas más que a los procesos lleva a replantear también la manera de gestionar los riesgos, es decir, no orientados al proceso sino a las personas. Las fases del proyecto las ejecutan personas, así como el producto lo realizan también personas.

Este trabajo se organiza de la siguiente forma: en el segundo acápite se presenta la conceptualización de proyectos, de riesgos y de metodologías ágiles como Scrum; en el tercero, se presenta la metodología que se siguió para la revisión de literatura y los resultados de su aplicación, se analizan las investigaciones realizadas en gestión de proyectos, gestión de los riesgos en los proyectos de software y la gestión de proyectos con el marco Scrum, todos ellos centrados en identificar los riesgos asociados a las personas que son quienes desarrollan el software, y de esta forma identificar limitaciones que sirven como punto de partida para futuros trabajos en esta línea; en el acápite cuatro, las conclusiones de la revisión y, finalmente, las referencias.

## II. MARCO CONCEPTUAL

### A) Proyectos

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los mismos implica que se tenga un principio y un fin definidos. El final se alcanza cuando se logran los objetivos del proyecto, cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

Asimismo, se puede poner fin a un proyecto si el cliente (patrocinador o líder) desea terminar el proyecto [2].

Noriega Martínez et al. [3] indican que “un proyecto de software es el desarrollo de software incluyendo artefactos relacionados. El proyecto de construcción de software involucra diversas áreas del conocimiento utilizadas en mayor o menor grado durante las fases del proyecto, en las áreas de actividades, de gestión y desarrollo”.

### B) Riesgos

Riesgos es la combinación de la probabilidad de ocurrencia de un evento y su consecuencia [4]. Cobit [5] define los riesgos informáticos como riesgos de negocio asociados con el uso, propiedad, operación, desenvolvimiento, influencia y adopción de TI dentro de una organización.

Los riesgos en los proyectos de software se definen como “el proceso sistemático de identificación, análisis y respuesta a los riesgos que se presentan durante el ciclo de vida de un proyecto. Su objetivo principal es minimizar la probabilidad y las consecuencias de los eventos perjudiciales” [5].

### C) SCRUM

SCRUM “es un marco de referencia dentro de la metodología de desarrollo de software ágil” [7], que se define como: “Una estrategia flexible y holística de desarrollos de productos, donde un equipo de desarrollo trabaja como una unidad para alcanzar un objetivo común” [6].

“Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo” [8].

Scrum no define un proceso formal de gestión de riesgos mucho menos los riesgos del propietario. Sin embargo, cada artefacto Scrum o reunión potencialmente ayuda a identificar o mitigar los riesgos. Si el Scrum master, el product owner o el equipo desarrollador quiere gestionar los riesgos de una manera formal o más proactiva, entonces pueden simplemente registrarlo en el burn-down chart, si efectivamente hacen el registro es recomendable priorizar por el valor más alto y los requisitos del próximo Sprint que estén en un alto riesgo primero [9]

## III. REVISIÓN BIBLIOGRÁFICA

La metodología utilizada para la revisión bibliográfica fue desarrollada siguiendo la guía para ejecutar revisión de literatura sistemática en Ingeniería de Software [10]. Los pasos que se siguieron fueron: Planeación de la revisión, ejecución y reporte. Algunas de las actividades llevadas a cabo en este sentido se enuncian a continuación:

Para el estudio primario, de acuerdo con la pregunta de investigación, se realizó un bosquejo de la estructura y características pretendidas, fundamentando la revisión en artículos, tesis y libros.

Para el estudio secundario se realizaron búsquedas en diferentes bases de datos, principalmente Ebsco, IEEE, Google Scholar. La información recolectada fue en su mayoría analizada y clasificada durante el segundo semestre de 2017.

Las principales palabras claves utilizadas fueron:

- Risk + Software + People
- Risk + Software + Scrum
- Project + Risk + Scrum
- Scrum + Software + Riesgo

Los criterios de preselección, fueron los siguientes:

- Fecha de publicación reciente.
- Idioma (inglés, español).
- Lectura del título del artículo, tesis u otro.



- Análisis del resumen y conclusiones.
- Pertinencia con el tema de investigación.

Finalmente, el análisis de la información seleccionada se enfocó en el tema de investigación, para identificar con estas bases si es procedente o no con la hipótesis definida. La revisión se estructuró realizando agrupación de los trabajos en gestión de proyectos, metodologías ágiles – Scrum y gestión de riesgos.

### A) Gestión de Proyectos

Entre los artículos analizados en esta categoría se revisaron 16, 10 de los cuales corresponden a publicaciones sobre proyectos finalizados que presentan resultados y conclusiones claras sobre los aportes que ofrece al área de estudio. En los seis restantes se presentan resultados parciales y se plantean algunas aplicaciones específicas.

Saleh [11] indica que muchos proyectos fallan por una pobre gestión, y agrega que hay varios elementos primordiales en la gestión del proyecto: gestión de riesgos, gestión del recurso humano, estimación de esfuerzo y costo, monitoreo del proyecto, gestión del tiempo, etc. Sin embargo, no involucra de manera explícita las etapas de un proyecto con una metodología ágil como Scrum dentro de las etapas del proyecto. Queda entonces el interrogante de los efectos o consecuencias de dicha interacción.

Marieta Peña et al. [12] cuestionan el origen de las fallas en los desarrollos de software, y las relacionan directamente con una mala estimación de éstas, lo que genera como consecuencia el fracaso inminente en los proyectos. Por esta razón, proponen un método para mitigar esta situación por medio del modelo de la representación lingüística 2- tupla, en la cual se tienen como variables principales las opiniones de diversas personas expertas, que se analizan y cuantifican matemáticamente. El autor no se enfoca en una metodología específica de riesgos, por el contrario, indica que el resultado de la factibilidad del proyecto es finalmente el punto clave para analizar los riesgos y así tomar las decisiones que requiera el proyecto.

Wuttke et al. [13] indican algunos factores que se deben considerar en los proyectos como: los patrocinadores, la organización, el ambiente interno y externo, la gestión del tiempo, los costos, recursos humanos, riesgos, adquisiciones, interesados, etc. Es importante tener presente que, si bien el autor busca tener controlado el proyecto, no se circunscribe a una metodología de desarrollo de software específica. El PMBOK es claro en especificar en la etapa de ejecución el momento en el cual inicia efectivamente el desarrollo del producto. No se tiene en consideración, entonces, en las fases iniciales la comprensión o entendimiento de la necesidad específica por parte de las personas que harán parte del equipo desarrollador del producto, puesto que su inicio está directamente asociado a la etapa de ejecución, lo cual podría afectar de una forma u otra el resultado del proyecto.

Banica et al. [14] explican cómo los desarrollos de software ágil requieren una gestión de proyectos que se adapte a sus características, y por eso enuncian las DevOps (en inglés Development Operations), las cuales no constituyen un método nuevo, sino una guía o manera de hacer por medio del seguimiento continuo al proyecto, el compromiso del equipo de acuerdo con sus roles y la calidad del producto. Cada uno de estos aspectos hace parte de la gestión en proyectos de desarrollo de software, sin embargo, hace falta un método o modelo claramente definido que permita hacer la gestión del riesgo orientado a las personas.

Amézquita et al. [15] explican cómo relacionar la gestión de proyectos de software según las indicaciones del PMI en desarrollos de software con Scrum, y muestran las ventajas de esta metodología ágil versus las fases de un proyecto, pues si desde la fase de planeación el cliente empieza a ver resultados, va a estar más satisfecho. El hecho de recibir entregas parciales de acuerdo con los Sprints, junto con las diversas reuniones, permiten tener un mayor control sobre la ejecución del proyecto. Este artículo no profundiza en una gestión de riesgos para Scrum orientada a personas, solo afirma que éstos pueden mitigarse por medio del control continuo que se hace con Scrum y la gestión del proyecto.

Ceballos et al. [16] identifican como factores claves en los proyectos: las personas, el proceso, el proyecto y el producto. Muchas veces el proyecto se puede ver afectado por lo que ellos llaman infoxicación, que no es más que un uso inadecuado de Internet por parte de las personas, lo cual de una forma u otra termina afectando el proyecto. En cuanto al proceso, proyecto y producto, se refiere a la necesidad de seguir una metodología ya sea XP o Scrum u otra que sea adecuada al tipo de proyecto en particular, con el fin de mitigar posibles riesgos. Como desventaja se encuentra que los autores no tienen en cuenta una metodología guía o modelo completo que permita hacer gestión a los riesgos con Scrum.

Leidy Patricia Carranza [17] se refiere a los aspectos claves que van a permitir el éxito en los proyectos considerando el espectro, también llamado las 4 P: personal, porque las personas son fundamentales para el desarrollo del proyecto: proceso, porque dependiendo de las decisiones que allí se tomen, va a ser el resultado del producto; proyecto, porque la gestión al mismo va a permitir tener un mayor control para obtener el resultado deseado y, finalmente, producto, que es lo que se quiere lograr, se debe medir y verificar. La autora relaciona la importancia de evaluar los riesgos antes que se presente una situación no deseada, no obstante, no indica el cómo ni tampoco se refiere a ningún método específico ágil.

En [18] se da a conocer cómo debe ser la administración de proyectos para software ágil indicando que, en un mundo cambiante, los requisitos son también cambiantes. Con el enfoque ágil se requieren entregas iterativas y rápidas, sin olvidar la flexibilidad y el código trabajado. El autor indica que hoy en día los sistemas complejos no son lineales, por esto se requiere una APM o administración de proyectos ágil que permita completar el proyecto para lograr la satisfacción del cliente, con un producto que de verdad agregue valor al negocio. En este sentido, el autor justifica la administración de proyectos ágil para, en otras palabras, tener mayor control sobre el proyecto y de esta manera lograr el éxito. No obstante, no especifica una metodología de riesgos en la que se involucre a las personas y solo se refiere en forma general a Scrum.

Mariño et al. [19] dan a conocer investigaciones que giran en torno a la gestión del desarrollo y mantenimiento de software, indican que el desarrollo ágil constituye una alternativa a los desarrollos tradicionales caracterizados por su rigidez, y mencionan que la metodología implica una gestión de proyectos orientada al trabajo en equipo, la cual catalogan como una mejor práctica. Los autores no profundizan en las metodologías ágiles usadas en el desarrollo de software, proporcionan una propuesta metodológica para la gestión de proyectos de software ágil, basada en web, pero no presentan ninguna estrategia para la gestión de los riesgos orientados a las personas.

Joffre [20] da a conocer diversos aspectos que a nivel de seguridad deben considerarse en los proyectos web desarrollados con PHP, los cuales sirven como mitigación de posibles riesgos que puedan presentarse en las aplicaciones. Joffre no relaciona una metodología de riesgos específica, lo cual deja a consideración del desarrollador.

Vargas et al. [21] demuestran que seguir las etapas de gestión de proyectos es crucial para su resultado final, para ello utilizan CMMI orientado al desarrollo de software. Los resultados finales indican que seguir la metodología impacta positivamente el resultado final. Para la gestión del riesgo se basan en la propuesta de CMMI de una manera iterativa durante todo el proyecto. Los autores realizan su estudio sin precisar metodologías específicas de desarrollo y muestran su análisis sin describir de forma detallada posibles riesgos asociados a las personas que intervienen en el proyecto.

Rodríguez et al. [22] dan a conocer un enfoque que permite gestionar el conocimiento de los proyectos de software con Scrum, en este sentido, todas las lecciones aprendidas de las etapas del proyecto con Scrum quedan registradas y podrán servir como base para otros proyectos. El interés de los autores es básicamente aprender de los éxitos y fracasos de otros proyectos, no obstante, no tienen en cuenta la gestión de riesgos como metodología que permita aprovechar de alguna manera la gestión del conocimiento.

En [23] se presenta una metodología de gestión de proyectos basada en PMI, con énfasis en Scrum y XP por ser las metodologías con mayor respaldo y acogida a nivel mundial. El autor toma como base los valores del manifiesto ágil: individuos, software funcional, colaboración del cliente, respuesta a cambios, sumados a los principios ágiles, y establece una relación entre cada una de las etapas del proyecto descritas en el PMBOK versus Scrum y XP. En lo que respecta a la gestión de riesgos, indica que se debe realizar de forma permanente, pero no profundiza en el tema y tampoco indica el cómo realizarla.

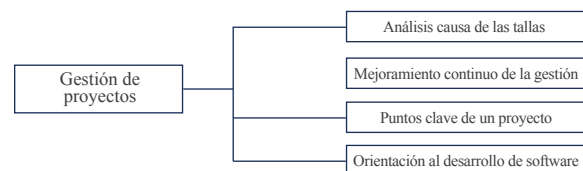
Muñoz [24] explica la manera de aplicar la ingeniería colaborativa en los proyectos de software que utilizan la guía del PMBOK. Indica además que quienes participen en los proyectos deberán contar con la experiencia y el conocimiento requeridos. También considera necesario tener las herramientas y procedimientos para la gestión del proyecto. El autor le otorga gran valor al compromiso de los actores que participan en el proyecto, al trabajo en equipo, a la responsabilidad personal y a las habilidades interpersonales. Si bien el autor tiene a las personas como punto clave para el éxito del proyecto, no especifica ni clasifica riesgos relacionados con ellas en las etapas del proyecto.

Cârstea [25] manifiesta la necesidad de entender y tener control de las operaciones del proyecto y ser flexibles al cambio. Recomienda además contar con herramientas para medir el progreso de los proyectos. El autor proporciona algunas pautas con el fin de lograr los objetivos del proyecto, pero no tiene en cuenta las etapas del proyecto ni la gestión de riesgos.

Rashid [26] asegura que el 10 % de las inversiones en un proyecto se van a la basura debido a una mala gestión del proyecto. El autor indica que hay una gran diferencia entre lo que la gente quiere y lo que obtiene, teniendo presente que dos proyectos no son iguales y que las herramientas usadas para gestionar tampoco lo son. En su análisis, el autor no tiene en cuenta técnicas para la gestión del riesgo del proyecto, aunque otorga gran relevancia a la elaboración de los objetivos del proyecto.

En los trabajos analizados en esta sección, se identifican diversas metodologías y estrategias planteadas para lograr el éxito en la gestión de proyectos, sin embargo, no se identifica una metodología en la que la gestión del riesgo se presente con el enfoque de orientación a las personas. Estos artículos permiten identificar que a pesar de aplicar metodologías y buenas prácticas, se siguen presentando fallas en los proyectos en gran medida relacionadas con las personas.

La revisión en cuanto a la gestión de proyectos, se puede agrupar en los siguientes conceptos, como se muestra en la Figura 1.



**Fig. 1.** Agrupación en Gestión de proyectos

## B) Metodologías Ágiles - Scrum

Dentro de esta categoría se presentan los 18 artículos analizados sobre Metodologías ágiles, todos corresponden a investigaciones finalizadas, en los cuales se hace referencia a los riesgos que se presentan en los proyectos que se trabajan bajo estos principios.

Las metodologías ágiles pueden proporcionar pautas que van a conducir a una visualización pronta del producto o parte de él. Aamir y Naeem [27] proponen un modelo para incorporar las actividades de control de calidad en Scrum, en relación con el concepto de pruebas del *backlog* (requisitos de usuario). El autor se refiere a Scrum como un proceso de desarrollo de software liviano que es ampliamente usado en la industria debido a su corto ciclo de vida de desarrollo y rápida respuesta a los cambios en los requisitos. El proceso de calidad de software es un punto importante a considerar con el fin de entregar productos con calidad, no obstante, debería tenerse presente como un factor relevante en la gestión del riesgo del proyecto, pero más orientado a los riesgos generados por las personas que intervienen en él.

Henriksen y Pedersen [28] relacionan el éxito de los proyectos y las metodologías ágiles, y dan a conocer cómo cada uno de los aspectos presentes en las metodologías pueden llegar a afectar de una forma u otra el resultado del proyecto. Dependiendo del tipo de proyecto, una actividad puede ser o no conveniente. En el estudio que realizaron concluyeron que las variables de mayor relevancia son: las iteraciones, la revisión del sprint, el diseño incremental, la retrospectiva del sprint y el mejoramiento colaborativo. Si bien el enfoque corresponde a analizar las metodologías, solo se referencia los riesgos como uno de los puntos críticos en el resultado de un proyecto, sin profundizar al respecto [28].

Yishuai Lin et al. [29] proponen una herramienta que sirva como asistente en el proyecto de software ágil, el cual se basa en el paradigma del Sistema Multi-Agente (Multi-Agent System –MAS–). Lo que buscan básicamente con esta herramienta es el mejoramiento en el monitoreo, la estimación de costos del proyecto y la eficiencia de los trabajadores. Hace falta explicitar cómo gestionar los riesgos relacionados con las personas que intervienen tanto en el proyecto como en el producto.

Una comparación entre Scrum, XP y Crystal se presenta en [30], los autores concluyen finalmente que las metodologías ágiles, además de ser adaptativas y no requerir tanta documentación como las tradicionales, tienen una orientación más a las personas que a los procesos. También indican que no son apropiadas para todos los tipos de contexto de software, pues, por ejemplo, requieren una gran participación del cliente, y si este no tiene la disposición constituye un aspecto que puede llevar al fracaso del proyecto. En este artículo los autores no especifican una metodología de riesgos, solo especifican algunas pautas a tener en consideración para que los proyectos puedan lograr el éxito.

Walczak y Kuchta [31] realizaron un estudio en una compañía, para lo cual seleccionaron tres proyectos con la idea de utilizar tres tipos de metodologías ágiles, entre las cuales se encontraban Scrum y XP. Concluye el estudio que la administración de riesgos en las metodologías ágiles no es suficiente, pues existen riesgos en el proyecto que no son mitigados.

Si bien los autores detallan cada uno de los puntos en los cuales es necesario gestionar los riesgos, no tienen en cuenta la gestión del proyecto como tal.

H. G. Gemünden [32] indica que normalmente cuando se habla de metodologías ágiles se refiere a productividad y eficiencia. Manifiesta que es importante tener un equilibrio entre los individuos, el desarrollo, el trabajo colaborativo con el cliente, los procesos, herramientas, documentación, negociación y planes. Por este motivo, señala que la metodología no es ágil sino frágil, y se cuestiona si las metodologías ágiles consideran suficientemente los riesgos de una forma proactiva o solo responden por entregas rápidas. El autor anima a que se investigue al respecto de modo que se logre el equilibrio deseado, pero no indica una manera de gestionar riesgos orientados a las personas.

Gamboa [33] asegura que seleccionar las metodologías ágiles que mejor se adapten a la necesidad específica no garantiza el éxito del proyecto, pero sí puede ayudar. Se refiere a los principios de manifiesto ágil e indica también que se debe evitar el desperdicio de tiempo en los proyectos. Hace un comparativo de varias metodologías ágiles entre las cuales se encuentra XP Scrum. Con relación a Scrum, menciona las herramientas para que los proyectos estén organizados. El autor lleva a cabo un estudio en una fábrica de software con problemas de retrasos en las entregas, donde pasan de trabajar con metodologías tradicionales a ágiles, y obtienen resultados satisfactorios, pues se logra entregar incluso antes del tiempo estipulado. Se logró además un mejor trabajo en equipo fomentando la colaboración en él y mayor satisfacción entre las personas involucradas en el proyecto. El autor no estipula ninguna metodología de riesgos y asume que por el hecho de tener metodologías ágiles se acota la probabilidad de riesgo en el proyecto.

Mariño y Alfonzo [34] indican cómo Scrum puede aplicarse en distintos tipos de proyectos no exclusivos al software, y muestran cómo utilizarlo en un trabajo final de aplicación en una carrera universitaria en Argentina. Toman en consideración para la gestión del proyecto, además de su planificación, la gestión de los riesgos, la cual infieren como necesidad

continua de ajuste por medio de reuniones de revisión y retrospectiva. Si bien los autores involucran la gestión de los riesgos, no definen una manera explícita de cómo hacer efectiva esta gestión orientada a las personas, pues no indican un método o modelo que se pueda seguir.

Flórez y Grisales [35] presentan un conjunto de criterios que pueden servir para seleccionar una metodología de desarrollo de acuerdo con las características del proyecto. En este sentido, se refieren a RUP para proyectos grandes, mientras que las metodologías ágiles son orientadas a proyectos pequeños, en los que las personas son el principal factor de éxito, lo que permite entregas tempranas en un desarrollo progresivo. En cuanto a Scrum, dicen que el control de calidad es difícil de implementar dadas sus características. Los autores no presentan una manera explícita de gestionar los riesgos orientados a personas, por el contrario, dejan todo el trabajo a la metodología o marco de trabajo.

Ozierańska et al. [36] indican que los factores por los que fallan los proyectos con metodología ágil están relacionados con la estructura organizacional, el proceso del desarrollo del producto, la gente involucrada en el proyecto, la tecnología y el diseño. En cuanto a las personas, el análisis se basa en la parte psicológica y emocional de los miembros del equipo. Los autores muestran situaciones que se pueden presentar en proyectos con Scrum por medio de cuadros donde incluyen un conjunto de puntos para considerar por cada factor, no obstante, no detallan cómo sería la gestión de los riesgos relacionada con cada factor ni la relación directa con la gestión del proyecto desde todas sus fases.

Becerra y Sanjuán [37] proponen una manera de desarrollar sistemas seguros con la metodología Scrum, acorde con todas las etapas del desarrollo. Para ello recomiendan identificar los activos involucrados en todo el proceso y los riesgos que puedan presentarse. Los autores no seleccionan una metodología de riesgos específica, sino que dan a conocer las que ellos consideran más reconocidas: Cramm, Magerit y Octave. Los autores indican qué se debe hacer para tener sistemas seguros, pero no manifiestan el cómo, dejan que sea el lector quien

finalmente determine de qué manera podría gestionar los riesgos con Scrum y tener control de la seguridad en el Sistema de Información. La orientación de los autores es exclusivamente al producto.

Godoy et al. [38] presentan una herramienta que permite simular desarrollos con Scrum con el fin de que los administradores del proyecto conozcan el impacto de las decisiones tomadas y así tengan un mayor control sobre el proyecto y el desarrollo Scrum. Si bien este tipo de herramientas proporcionan una manera de visualización del producto a futuro, no contienen un modelo de gestión del riesgo, además la gestión del proyecto no es tomada en cuenta en todas sus etapas.

Alan Moran [39] propone una metodología para la gestión de riesgos ágil y Scrum, en ella explica la necesidad de tener presente, para la gestión de los riesgos: los objetivos del proyecto, su contexto y el ambiente del riesgo que pueda afectar el desarrollo del producto. También da a conocer de qué forma se pueden gestionar los riesgos de acuerdo con Scrum. Sin embargo, el autor no explica riesgos comunes en los proyectos de software en todas sus etapas, ni su relación explícita con las personas.

Merhout y Kovach [40] indican que el gobierno de TI involucra diferentes aspectos como: gestión y optimización de riesgos, seguridad, prevención de fraudes, documentación de sistemas, estrategias, etc. Los autores afirman que las metodologías ágiles como Scrum son tan compatibles con el gobierno de TI como las metodologías tradicionales. Si bien los autores hacen referencia a la gestión de riesgos como algo inmerso en Scrum y desde el gobierno de TI, no explican cómo podría gestionarse.

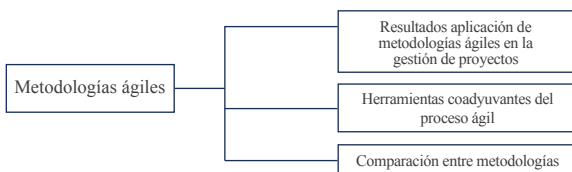
Tanner y Mackinnon [41] demuestran las causas por las cuales se generan interrupciones en la mitad de un Sprint, y concluyen, a partir del estudio, que algunas de estas son: falta de entendimiento de lo que el cliente quiere, no conocer en detalle cómo trabaja Scrum, altas cargas de trabajo para los desarrolladores y fallas en la comunicación entre departamentos. Si bien los autores indican las causas y proporcionan la forma de mitigar cada una de estas situaciones, hace falta tener en cuenta los riesgos asociados a las personas que se generan en las diferentes etapas del proyecto.



Rodríguez y Dorado [42] se refieren a las fortalezas de Scrum, aunque manifiestan también algunas debilidades como: la dependencia hacia el cliente, la poca documentación, el tiempo en el que el cliente no interviene en el proyecto cuando se requiere que lo haga, la estimación del tiempo y el costo del proyecto cuando se afecta por situaciones adversas. Si bien los autores indican las limitantes de Scrum, no establecen mecanismos que permitan gestionar los riesgos asociados a ellas.

Villegas, Ruiz y López [43] realizan un análisis relacionado con los inconvenientes que pueden presentarse con las personas y Scrum. Indican que el conflicto es algo natural en el ser humano y que se debe tener presente que el software es desarrollado por personas para personas. Los autores enfatizan en la importancia de la comunicación en los equipos Scrum. En su análisis concluyen que los problemas asociados a personas en Scrum se dividen en: Conflictos por tareas (ideas u opiniones), proceso (forma de hacer las cosas en el equipo) y relación (diferencia en actitudes, personalidad, valores). Si bien el autor tiene en cuenta a las personas como factores de riesgo en el proyecto con Scrum, no identifica los riesgos posibles ni el posible tratamiento que pueda llevarse a cabo para su respectiva gestión.

Los trabajos analizados en esta sección se pueden agrupar de acuerdo con los elementos que se muestran en la Figura 2.



**Fig. 2.** Agrupación de trabajos con metodologías ágiles

Para esta sección, se tuvo en cuenta analizar la forma en que se identifican y enfrentan los riesgos desde las metodologías ágiles, al respecto se identifican trabajos en los cuales listan posibles riesgos, pero sobre todo asociados a los proyectos, y no se plantea una metodología o estrategia clara de manejo de los riesgos.

### C) Gestión de Riesgos de Software

Los artículos analizados en esta sección corresponden a 10 proyectos finalizados con conclusiones y aportes concretos, que se enfocan en los riesgos que se identifican en diferentes proyectos de software.

Sam Ghosh et al. [44] afirman que tanto el PMBOK como Scrum proporcionan métodos para manejar los riesgos. Indican que Scrum estratégicamente ataca y bloquea los riesgos en cada Sprint de manera tal que el product owner puede tomar decisiones acerca de qué hacer con los riesgos, manejarlos o gestionarlos, en otras palabras, el product owner gestiona los riesgos y ejecuta el backlog. Entonces establece una relación entre la manera de gestionar los riesgos del PMBOK y aprovechar las características de Scrum para decidir qué hacer con las situaciones de riesgo que se presenten. No obstante, el nivel que presenta es general y no indica una relación directa con los proyectos de software.

Rabechini y Monteiro [45] demuestran que existe una relación directa entre la gestión de riesgos y el éxito en los proyectos. Afirma que el impacto de la gestión de riesgos en los proyectos se puede ver desde un punto de vista positivo, como una oportunidad, o desde un punto de vista negativo. Relacionan el cargo de administrador de riesgos como un profesional que identifica posibles situaciones que se pueden presentar y que además establece la respuesta a ellas. Si bien los autores concluyen que la gestión del riesgo tiene total relevancia para lograr el éxito en los proyectos, no tiene un enfoque en proyectos de desarrollo con metodologías ágiles.

Una investigación de estándares relacionados con el proceso de desarrollo de software se expone en [46]. En ella se concluyen 12 factores y 52 subfactores que deberían considerarse con el fin de minimizar los riesgos en el proyecto. Según los autores, los proyectos pueden fallar por: cambios en los requisitos, cronogramas optimistas, presión por entregar a tiempo, complejidad del software, etc. Cada uno de estos factores tiene que ver necesariamente con las personas involucradas. Los autores no tienen en consideración las metodologías ágiles, por tal motivo se podrían tener en cuenta los factores como aspectos



relevantes para la gestión del riesgo a personas, pero ajustados a las metodologías ágiles.

Letier, Stefan y Barr [47] relacionan los riesgos con la toma de decisiones que afectan la arquitectura de software y los requisitos del sistema, lo cual genera pérdidas. Para ello proponen un método que pretende modelar la incertidumbre tanto para requisitos como para arquitectura, por medio de un análisis matemático que infiera las consecuencias y permita de esta manera una mayor probabilidad de éxito en los proyectos. Aunque los autores proporcionan herramientas matemáticas para lidiar con decisiones en requisitos y arquitectura en consecuencia con las características del individuo, no tienen en cuenta las características de las metodologías ágiles ni tampoco la gestión de proyectos con PMBOK.

Ebenezer, Path y Binfield [48] dan a conocer la situación particular de las aplicaciones web relacionadas con los bloqueos a éstas. Un aspecto que se debe considerar cuando los programas las identifican como peligrosas, así no lo sean. El punto de vista del autor cobra relevancia cuando se tiene en cuenta como un factor para considerar en la planeación inicial del proyecto, como parte de la gestión de riesgos de acuerdo con la metodología ágil, dado que no tiene una relación directa con el desarrollo de software.

Cleland-Huang [49] se refiere a la necesidad de desarrollar sistemas seguros con Scrum, para este fin indica cómo, antes de hacer las historias o lista de requisitos para desarrollar, se elabore la lista que incluye un desarrollo seguro. Se busca involucrar al cliente, los interesados, usuarios y el equipo, para plasmar en un documento su percepción acerca de cómo debe trabajar el sistema para que sea seguro. Partiendo de este punto, el product owner prioriza las historias de usuario y las historias de seguridad, finalmente el equipo selecciona las historias para el Sprint y se obtiene un producto. Es así como el autor se enfoca en considerar desde un principio la seguridad del producto, no tiene en cuenta aquellos riesgos o amenazas que pueden presentarse asociados a las personas desde el proyecto e incluso en el resultado final del producto, ya que su foco de atención es exclusivamente la parte técnica.

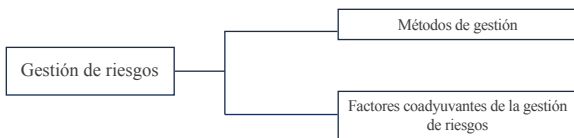
R. Chapman et al. [50] dan a conocer su punto de vista acerca de la agilidad versus los sistemas desarrollados, en los que prevalece la integridad como calidad en el desarrollo de software. Muestran la dependencia de lo “ágil” hacia las pruebas. Indican algunas pautas en cuanto a las historias de usuario y requisitos no funcionales, de modo que se tengan en cuenta posibles situaciones que puedan afectar negativamente el desenvolvimiento del producto final, y recomiendan para sistemas críticos una documentación más formalizada. Si bien los autores dan a conocer algunas pautas que pueden ayudar a complementar la metodología ágil con el fin de minimizar posibles situaciones no deseadas, no profundizan en una metodología que permita controlar mayormente el proceso ágil para mitigar los riesgos, y tampoco tienen en cuenta las fases del proyecto de software.

Pérez Moya y Zulueta [51] definieron un proceso que permite gestionar los riesgos en los proyectos de software. Para ello tomaron los artefactos que arroja el proyecto, y definieron actividades que involucran: planificar la gestión de riesgos, identificación, análisis, actividades para la resolución, comunicación, control y evaluación. Este proceso lo aplican y obtienen resultados positivos. Si bien la gestión propuesta presenta sus ventajas de acuerdo con lo especificado, no se refieren en ningún momento a algún tipo específico de metodología de desarrollo de software ni tampoco a riesgos asociados a las personas.

Cagliano et al. [52] proponen un marco de gestión de riesgo con base en diversos métodos existentes en la literatura, relacionando las etapas de gestión del riesgo con las del proyecto. También se refieren al término de madurez en la gestión del riesgo, como un conjunto de fases por las cuales las empresas podrían pasar. El nivel de madurez permitirá conocer la efectividad de la gestión del riesgo. Las propuestas de los autores proporcionan bases para la gestión, no obstante, no profundizan en todas las etapas del proyecto, y aunque mencionan la necesidad de mejorar la comunicación de los riesgos como un punto importante, su enfoque no es hacia los riesgos asociados a las personas. Otro punto para considerar es que no se refieren a un tipo específico de proyecto con Scrum.

Martell-Fernández y Zulueta [53] crearon un modelo para análisis de riesgo en líneas de producto de software orientado a los activos de software. Con este fin se refieren a contar con un analista de riesgos, como también con un equipo de riesgos que se encarga de identificarlos, priorizarlos, medir el impacto y darles priorización. En este proceso el equipo analiza el resultado y hace seguimiento por medio de indicadores para tener un mayor control. Los autores no tipifican riesgos y no tienen en cuenta a las personas como entes generadores de riesgo, su énfasis son los activos del proyecto de software.

Los trabajos analizados en esta sección particularmente muestran los métodos y factores que se han utilizado para gestionar los riesgos en proyectos de desarrollo de software, y se resume en la Figura 3.



**Fig. 3.** Trabajos en gestión de riesgos.

#### D. Limitaciones

Es importante recordar que los proyectos de software son ejecutados por personas que, al tener limitaciones en cuanto a conocimiento, su nivel técnico e incluso en sus relaciones sociales, pueden inclinar la balanza hacia el éxito o el fracaso. Con respecto a la gestión de riesgos, Viteri Cárdenas [54] afirma que Scrum

... no tiene una gestión de riesgos explícita, por tal razón y tomando en cuenta que en el desarrollo de un proyecto suelen presentarse riesgos que afectan directa e indirectamente produciendo pérdidas se debe considerar la aplicación de alternativas que permitan la gestión oportuna de los riesgos.

Al respecto también se realiza un análisis con el fin de determinar por qué, a pesar de que los riesgos son gestionados en los proyectos, siguen ocurriendo proyectos que fracasan o son afectados negativamente y se concluye que la razón es la incompetencia de la gente, pues las personas no tienen el conocimiento o las habilidades necesarias para gestionar los riesgos [55].

De acuerdo con la literatura revisada, algunos autores si bien tratan la gestión de proyectos como tal no se enfocan en una metodología ágil como Scrum, otros sí tienen este enfoque pero no profundizan en la gestión de riesgos. Algunos autores tienen el enfoque hacia proyectos, riesgos y Scrum, pero no detallan una metodología o no tienen una orientación específica hacia las personas. Entonces se puede concluir que no existe como tal un modelo para proyectos de software con Scrum orientado a gestionar los riesgos relacionados con las personas.

#### IV. CONCLUSIONES

Los autores referenciados comparten la necesidad de gestionar los riesgos en los proyectos. Algunos proponen estrategias, mientras otros dejan el camino para que se profundice en el tema.

Scrum, como marco ágil de desarrollo, ofrece grandes ventajas según varios autores, no obstante, dejan las puertas abiertas a la metodología de gestión de riesgos que se va a utilizar.

Aunque algunos autores coinciden en que en gran parte del riesgo en los proyectos son las personas, no proporcionan una metodología clara de gestión de los riesgos asociados a las personas, por lo que la necesidad al respecto está latente.

A lo largo de la documentación analizada se observa, por un lado, el análisis de riesgos del proyecto y, por otro, el del producto, dejando la necesidad de la unión de ambos como manera de control.

Una parte de los riesgos en los proyectos de software se encuentra asociada a las personas, quienes, de acuerdo con su experticia, conocimiento y trabajo colaborativo, planean y ejecutan un proyecto. Scrum minimiza la documentación que se genera en otro tipo de metodologías. Tiene una orientación a las personas más que a los procesos, su objetivo son las entregas rápidas. Los riesgos asociados a las personas son aquellos que de una forma u otra son generados por miembros del equipo y llegan a afectar el resultado del proyecto, van desde la toma de decisiones en el proceso de planeación hasta la entrega de un producto que se espera sea de calidad y genere satisfacción al usuario.

## REFERENCIAS

- [1] A. Navarro Cadavid, Andrés, J. D. Fernández y J. Morales, “Revisión de metodologías ágiles para el desarrollo de software”, *Prospectiva*, vol. 11, N.º 2, pp. 30-39, 2013.
- [2] Project Management Institute, *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)*, Pensilvania: Project Management Institute, Inc, 2013.
- [3] R. Noriega M. et al., *Curso de Ingeniería de Software*. IT Campus Academy, 2015.
- [4] ISACA, *COBIT 5 for Risk*, 2017. Disponible en: <http://www.isaca.org/cobit/pages/risk-product-page.aspx> [Accedido: 30 de octubre de 2017].
- [5] L. F. Sanz y P. Bernad Silva, “Gestión de riesgos en proyectos de desarrollo de software en España: estudio de la situación”, *Revista Facultad de Ingeniería Universidad de Antioquia*, 70, pp. 233-243, 2014.
- [6] T. Dimes, *Conceptos Básicos de Scrum: Desarrollo de Software Agile y Manejo de Proyectos Agile*, Babelcube Inc., 2015.
- [7] T. Blokehead, *Scrum - ¡Guía definitiva de prácticas ágiles esenciales de Scrum!*, Babelcube Inc., 2016.
- [8] L. Straccia, P. Pytel y M. F. Pollo Cattaneo, “Metodología para el desarrollo de software en proyectos de I+ D en el nivel universitario basada en Scrum”, *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, 2016.
- [9] M. Tomanek y J. Juricek, “Project risk management model based on PRINCE2 and SCRUM frameworks”, 2015. Disponible en: <https://arXiv> preprint arXiv:1502.03595
- [10] B. Kitchenham et al., “Guidelines for performing Systematic Literature Reviews in Software Engineering”, 2007. Disponible en: [https://www.researchgate.net/publication/302924724\\_Guidelines\\_for\\_performing\\_Systematic\\_Literature\\_Reviews\\_in\\_Software\\_Engineering](https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering). 3 de Octubre de 2018.
- [11] K. A. Saleh, *Software Engineering*, U.S.A.: J. Ross Pub., 2007.
- [12] M. Peña Abreu, C. R. Rodríguez y P. Y. Piñero, “Computación con Palabras para el Análisis de Factibilidad de Proyectos de Software”, *Tecnura*, vol. 20, N.º 50, pp. 69-84. oct.-dic. 2016. doi: 10.14483/udistrital.jour.tecnura.2016.4.a05.
- [13] T. Wuttke, P. Snijders y A. Zandhuis, *El Compañero de Bolsillo de la Guía del PMBOK®*. Van Haren, 2014.
- [14] L. Banica, M. Radulescu, D. Rosca y A. Hagiu, “Is Devops Another Project Management Methodology?”, *Informatica Economică*, vol. 21, pp. 39-51, N.º 3, Jul. 2017. doi: 10.12948/issn14531305/21.3.2017.04.
- [15] R. A. Amézquita, “Análisis de la aplicación de la metodología SCRUM como complemento de las metodologías del PMI para el control de proyectos de desarrollo de software”, *Tesis de pregrado*, Universidad Militar Nueva Granada, 2015.
- [16] C. Arteaga, F. A. Barrera y J. J. Chaparro, “Factores claves en la gestión de proyectos”, *Tecnología Investigación y Academia*, vol. 1, N.º 2, 2013.
- [17] L. P. Carranza, “Gestión en Proyectos de Software”, *Tecnología Investigación y Academia*, vol. 4, N.º 2, pp. 12-19, 2016.
- [18] S. Augustine et al., “Agile Project Management: STEERING from the EDGES”, *Communications of the ACM*, vol. 48, N.º 12, pp. 85-89, 2005. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=bth&AN=21115061&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=21115061&lang=es&site=ehost-live).
- [19] S. I. Mariño, M. V. Godoy y P. L. Alfonzo, “Propuestas y revisión de metodologías de la Ingeniería del Software”, *XVI Workshop de Investigadores en Ciencias de la Computación*. 2014.
- [20] J. S. Monar, “Propuesta de un método utilizando técnicas de programación seguras para el desarrollo de aplicaciones web en entorno PHP para mitigar riesgos potenciales de seguridad”, *Tesis de maestría*, Escuela Superior Politécnica de Chimborazo, 2017.
- [21] W. Yépez, C. Primera y M. Torres, “Mejoras al proceso de planificación de proyectos de software usando el Modelo de Madurez de Capacidad Integrado (CMMI)”, *Compendium*, vol. 16, N.º 30, 2013.
- [22] G. H. Rodríguez et al., “Un enfoque inteligente para asistir en la planificación de proyectos ágiles”, *XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO)-XV*, Buenos Aires, 2014.

- [23] E. Londoño Hoyos, “Formulación de una metodología para la gestión de proyectos de desarrollo de software, a través de principios y prácticas ágiles con enfoque PMI”, Tesis de pregrado, Universidad Militar Nueva Granada, 2016.
- [24] J. L. Jurado y C. J. Pardo, “La Gestión de Proyectos de Software. Una Prospectiva en la Aplicación de Estrategias en la Ingeniería Colaborativa”, Lámpsakos, vol. 9, N.º 1, pp. 26-33, ene.-jun. 2013. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=fua&AN=95420203&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=95420203&lang=es&site=ehost-live).
- [25] C.G. Cârstea, “Know-How in Software Project Management”, *Economy Transdisciplinarity Cognition*, vol. 17, N.º 2, pp. 102-107, sept. 2014. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=bth&AN=100116514&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=100116514&lang=es&site=ehost-live)
- [26] B. Rashid, “What People Want from Their Project Management Apps in 2017”, *Forbes.Com*, 02 June 2017, p. 1. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=bth&AN=123441096&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=123441096&lang=es&site=ehost-live).
- [27] M. Aamir, y M. Naeem Ahmed Khan, “Incorporating quality control activities in scrum in relation to the concept of test backlog”, *Sādhanā*, vol. 42, N.º 7, pp. 1051-1061, 2017.
- [28] A. Henriksen y S. A. R. Pedersen. “A Qualitative Case Study on AGILE PRACTICES and PROJECT SUCCESS in Agile Software Projects.” *Journal of Modern Project Management*, May-Aug 2017, pp. 62-73. EBSCOhost, doi: 10.19255/JMPM01306.
- [29] L. Yishuai et al., “Multi-Agent System for Intelligent Scrum Project Management”, *Integrated Computer-Aided Engineering*, vol. 22, N.º 3, pp. 281-296, Sept. 2015. doi: 10.3233/ICA-150491.
- [30] A. Navarro, J. D. Fernández y J. Morales, “Revisión de metodologías ágiles para el desarrollo de software”, *Prospectiva*, vol. 11, N.º 2, pp. 30-39, 2013.
- [31] W. Walczak y D. Kuchta, “Risks Characteristic of Agile Project Management Methodologies and Responses to Them”, *Operations Research & Decisions*, vol. 23, N.º 4, pp. 75-95, oct. 2013. doi: 10.5277/ord130406.
- [32] H. G. Gemünden, “Foundations of Project Management Research: Stakeholders and Agile”, *Project Management Journal*, vol. 46, N.º 6, pp. 3-5, Dic. 2015/Ene. 2016. doi: 10.1002/pmj.21557.
- [33] J. C. Gamboa, “Aumento de la productividad en la gestión de proyectos, utilizando una metodología ágil aplicada en una fábrica de software en la ciudad de Guayaquil”, *Revista Tecnológica-ESPOL*, vol. 27, N.º 2, 2014.
- [34] S. I. Mariño y P. L. Alfonso, “Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación”, *Scientia et technica*, vol. 19, N.º 4, pp. 413-418, 2014.
- [35] L. Flórez y F. Grisales, “Formulación de criterios para la selección de metodologías de desarrollo de software”, Tesis, Universidad Tecnológica de Pereira, 2014.
- [36] A. Ozierańska et al., “The Critical Factors of Scrum Implementation in IT Project – the Case Study”, *Journal of Economics & Management*, vol. 25, N.º 3, pp. 79-96, Jul. 2016. doi:10.22367/jem.2016.25.06.
- [37] P. Becerra y M. Sanjuán, “Revisión de estado del arte del ciclo de vida de desarrollo de software seguro con la metodología SCRUM”, *Revista Investigación y Desarrollo en TIC*, vol. 5, N.º 2, 2017.
- [38] D. A. Godoy et al., “Simulando proyectos de desarrollo de software administrado con Scrum”, *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014.
- [39] A. Moran, *Agile Risk Management and Scrum*, Switzerland: Institute for Agile Risk Management, 2014.
- [40] J. Merhout y M. Kovach, “Governance Practices over Agile Systems Development Projects: A Research Agenda”, 2017.
- [41] M. Tanner y A. Mackinnon, “Sources of Interruptions Experienced during a Scrum Sprint”, *Electronic Journal of Information Systems Evaluation*, vol. 18, N.º 1, pp. 3-18, jul. 2015. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=bth&AN=108315662&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=108315662&lang=es&site=ehost-live).
- [42] C. Rodríguez y R. Dorado, “¿Por qué implementar Scrum?”, *Revista Ontare*, vol. 3, N.º 1 pp. 125-144, 2015.

- [43] E. Villegas, J. M. Ruiz y F. López, “El conflicto en el desarrollo ágil: Una perspectiva desde el SCRUM.” *Revista Gestión y Región*, N.º 21, 2016, pp. 121-149.
- [44] S. Ghosh et al., “Enhance PMBOK® by Comparing it with P2M, ICB, PRINCE2, APM and Scrum Project Management Standards”, *PM World Today*, vol. 14, N.º 1, pp. 1-77, 2012.
- [45] R. Rabechini y M. Monteiro de Carvalho, “Understanding the impact of project risk management on project performance: An empirical study”, *Journal of Technology Management & Innovation*, N.º 8, pp. 6-6, 2013.
- [46] H. H. Khan y M. Noman Malik, “Software Standards and Software Failures: A Review with the Perspective of Varying Situational Contexts”, *IEEE Access* 5, pp. 17501-17513, 2017.
- [47] E. Letier, D. Stefan y E. T. Barr, “Uncertainty, risk, and information value in software requirements and architecture”, *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014.
- [48] C. Ebenezer, P. Bath y S. Pinfield, *Access denied? Managing access to the Web within the NHS in England: technology, risk, culture, policy and practice*, 2016.
- [49] J. Cleland-Huang, “Safety Stories in Agile Development”, *IEEE Software*, vol. 34, N.º 4, pp. 16-19, 2017.
- [50] R. Chapman et al., “What Can Agile Methods Bring to High-Integrity Software Development? Considering the Issues and Opportunities Raised by Agile Practices in the Development of High-Integrity Software”, *Communications of the ACM*, vol. 60, N.º 10, pp. 38-41, oct. 2017. doi: 10.1145/3133233.
- [51] O. Pérez Moya y Y. Zulueta, “Proceso para gestionar riesgos en proyectos de desarrollo de software”, *Revista Cubana de Ciencias Informáticas*, vol. 7, N.º 2, 2013.
- [52] A. C. Cagliano et al., “Choosing Project Risk Management Techniques. A Theoretical Framework”, *Journal of Risk Research*, vol. 18, N.º 2, pp. 232-248, Feb. 2015. doi: 10.1080/13669877.2014.896398.
- [53] V. Martell-Fernández y Y. Zulueta, “Modelo para el Análisis de Riesgos en Líneas de Productos de Software”, *Revista Cubana de Ciencias Informáticas*, vol. 8, N.º 1, pp. 82-98, ene.-mar. 2014. Disponible en: [search.ebscohost.com/login.aspx?direct=true&db=fua&AN=94967093&lang=es&site=ehost-live](http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=94967093&lang=es&site=ehost-live).
- [54] J. A. Viteri, “Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE.” *Propuesta de gestión de riesgos de proyectos software, desarrollados con la metodología SCRUM*, 2015.
- [55] Klakegg, Ole Jonny. “Project Risk Management: Challenge Established Practice.” *Administrative Sciences (2076-3387)*, vol. 6, no. 4, Dec. 2016, pp. 1-3. EBSCOhost, doi:10.3390/admsci6040021.





## Capítulo VI

# REPRESENTACIÓN DE EVENTOS A PARTIR DE ESTRUCTURAS LINGÜÍSTICAS BASADAS EN ROLES SEMÁNTICOS: UNA EXTENSIÓN AL ESQUEMA PRECONCEPTUAL

*Paola Andrea Noreña Cardona - panc524@gmail.com*

Tecnológico de Antioquia - Institución Universitaria, Medellín, Colombia.

*Carlos Mario Zapata Jaramillo - cmzapata@unal.edu.co*

Universidad Nacional de Colombia, Medellín, Colombia.

*Aixa Eileen Villamizar Jaimes - aixa.villamizar@tdea.edu.co*

Tecnológico de Antioquia - Institución Universitaria, Medellín, Colombia.

## I. INTRODUCCIÓN

Los eventos son sucesos que pasan en un momento específico durante el flujo de procesos y causan cambios de estado en los procesos, que son producto de las condiciones y restricciones que incluyen los eventos [1–2]. La información que contienen los eventos es útil para identificar el comportamiento de un sistema de software. Esta información se puede referir a tiempo, declaración, mensaje, condición y señal [3]. Además, con los eventos se puede direccionar la ejecución del orden de los procesos en el sistema [4]. Los eventos pueden ser: (i) disparadores, aquellos que generan el inicio de servicios y procesos, y (ii) de resultado, aquellos que los finalizan. El evento disparador none o declaración es aquel que, a partir del cumplimiento de una instrucción, produce el disparo de los procesos. Este evento, generalmente, influencia a los demás, ya que puede disparar otros eventos. Por ejemplo, el evento mensaje surge como una alerta de la ocurrencia del evento declaración [5].

El Esquema Preconceptual (EP) es un modelo de ingeniería de requisitos para representar y facilitar la comprensión de los elementos de un dominio a los analistas y los interesados. Este esquema incluye características dinámicas y estructurales que permiten dar una vista completa del dominio a partir de

estructuras lingüísticas como: conceptos (sustantivos), relaciones dinámicas (que se pueden relacionar con procesos) que requieren un agente (quien realiza la acción) y relaciones estructurales que relacionan la dependencia entre conceptos y su sintaxis presenta los verbos tiene o es [6].

Los eventos de declaración basados en el EP se usan para disparar relaciones dinámicas y se representan a partir de estructuras lingüísticas denominadas relaciones eventuales. Estas relaciones incluyen en su sintaxis un concepto con un verbo intransitivo que contiene cero o un actante (elementos que acompañan al verbo); esta cantidad indica que en la relación eventual no se requiere un agente para realizar la acción [6].

Algunos enfoques del modelado computacional representan la vista comportamental de sistemas de software organizacional e incluyen eventos en su representación [3], [5], [7–14]. De ellos, algunos incluyen elementos del dominio científico [13–14]. Otros enfoques presentan la vista estructural del sistema incluyendo eventos [15–17]. De estas propuestas, sólo Liu et al. [17] incluyen elementos del dominio científico. Algunos de estos enfoques presentan errores lingüísticos en la representación de eventos disparadores de tipo declaración (se suelen

confundir con procesos y objetos; los verbos se conjugan en diferentes tiempos y algunos no presentan estructuras que deben acompañar al verbo). El EP integra las dos vistas (comportamental y estructural), a diferencia de los demás enfoques. Sin embargo, las estructuras lingüísticas se enfocan en sistemas de software organizacional para representar este tipo de eventos, por lo que faltan estructuras lingüísticas en el EP para representarlos en dominios de software científico. Estos dominios comprenden sistemas de software con contenido de información en áreas de ingeniería, medicina, ciencias y matemáticas [18].

En este capítulo, el EP se extiende a partir de estructuras lingüísticas basadas en roles semánticos, los cuales son elementos que acompañan al verbo para la interpretación semántica de una oración y determinan la estructura sintáctica y semántica de un evento [19]. Estos roles se pueden clasificar como actantes y circunstanciales. Los actantes son valencias (argumentos) de un verbo, de los cuales sólo se consideran en esta extensión el paciente y el experimentador, quienes reciben la acción o el evento. Pacientes y experimentadores se pueden usar para describir relaciones eventuales. Los circunstanciales amplían el significado de la oración (por ejemplo, fuerza y causa) y también se pueden usar en la descripción de un evento [20–21].

En dominios de software científico, los eventos presentan, además del verbo, entre cero y dos actantes y circunstanciales de fuerza y causa. Por lo tanto, en estos dominios, en la extensión al EP se incluyen estos roles semánticos para identificar funciones de los actantes y circunstanciales en la representación de eventos disparadores tipo declaración. Esta extensión incrementa las estructuras lingüísticas a partir de estos roles, para representar eventos disparadores tipo declaración en procesos de desarrollo de sistemas de software en dominios científicos. Finalmente, se presentan estructuras gráficas para incluir la representación.

Este capítulo se divide de la siguiente forma: en la Sección II se presenta el marco teórico; en la Sección III se expone el trabajo previo; en la Sección IV se plantea el problema; en la Sección V se propone la solución; en la Sección VI se incluyen los resultados,

y, finalmente, se presentan las conclusiones y el trabajo futuro.

## II. MARCO TEÓRICO

### A. Eventos

Los eventos surgen en un momento determinado y designan la ocurrencia de algo significativo que pasa en el flujo de procesos, como el comienzo o el fin de una operación. Los eventos son responsables de cambiar el estado de los procesos, y afectan la secuencia de los flujos a partir de condiciones o restricciones [1–2]. Por lo tanto, la identificación y el análisis de los eventos se pueden considerar procesos importantes para la ingeniería de requisitos. La información que se obtiene a partir de los eventos permite analizar el comportamiento de los sistemas de software que se especifica en productos de trabajo como: diagrama de procesos, tarjetas de educación de requisitos, diagrama de máquina de estados, diagrama de actividad y esquemas conceptuales, entre otros [22–23].

Los eventos disparadores se incluyen en los sistemas de software por su capacidad de generar el inicio de servicios o procesos (por ejemplo, sistemas de microservicios y sistemas con sensores). Además, pueden iniciar otros eventos disparadores. Un tipo de evento disparador es el None o declaración, el cual incluye información de las instrucciones que causan un disparo para generar procesos o servicios. Este evento puede indicar la ocurrencia de un fenómeno natural, mecánico o humano. Los eventos de resultado de los sistemas de software sirven para finalizar servicios o procesos [3].

### B. Esquema Preconceptual (EP)

El EP es un modelo de ingeniería de requisitos para representar de manera gráfica y computacional el dominio de un problema, y les permite a los analistas e interesados entender y validar fácilmente la información que se describe desde un sistema de software [24]. En el EP se utiliza un conjunto de estructuras lingüísticas y gráficas que se presentan en la Figura 1: conceptos (sustantivos, por ejemplo, factura, secretario, informe); relaciones estructurales (verbos que generan conexiones permanentes entre

conceptos, por ejemplo, secretaria es empleado); relaciones dinámicas (acciones y procesos; por ejemplo, secretaria registra factura); implicaciones (relación de causa efecto, por ejemplo, edita implica crea); condicionales (requisitos previos para realizar una acción, por ejemplo, la cuenta solo se puede pagar si la fecha es inferior a la fecha de vencimiento); las conexiones (enlaces entre conceptos, por ejemplo, secretario imprime informe); el joint (enlace entre otros enlaces), y el marco (informe) [6]. Además, el EP incluye relaciones de logro (objetivos determinados) y relaciones eventuales (verbos para representar eventos, las cuales incluyen un verbo intransitivo que se acompaña de un concepto [22]). Los EP incluyen eventos disparadores para iniciar relaciones dinámicas y otros eventos, y eventos de resultado para finalizar relaciones dinámicas en un dominio [6].

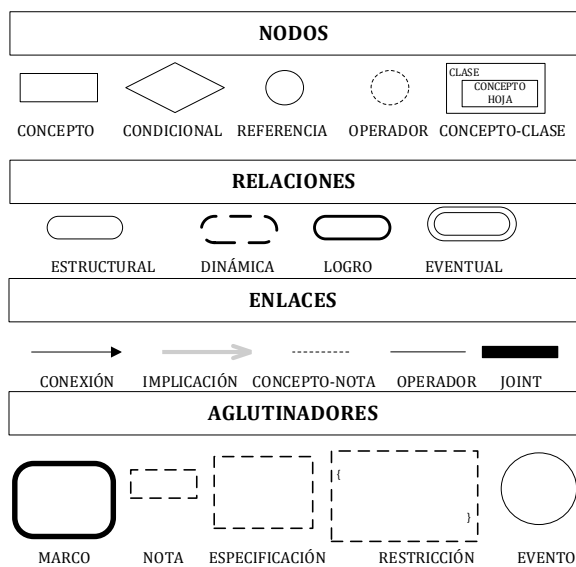


Fig. 1. Estructuras del EP [6].

### C. Roles Semánticos

Un rol semántico es la relación subyacente entre un participante y el verbo principal en la interpretación semántica de una oración. Los roles semánticos también se conocen como roles teta, roles temáticos o relaciones temáticas, papeles temáticos, funciones semánticas o casos profundos (casos de gramática) [25]. Estos roles permiten la identificación y representación de estructuras sintácticas y semánticas de un evento [19] y se clasifican en actantes y circunstanciales.

**1) Actantes:** Los actantes se conocen como argumentos o valencias de un verbo. Se consideran como sustantivos que realizan o están sujetos a una acción. La cantidad de actantes que aparecen en una oración se limita por las características de las valencias del verbo principal que tiene una oración. Tesnière [20] clasifica los verbos así: avalente (sin actante; por ejemplo, llovió), monovalente (un actante; por ejemplo, Sam durmió), divalente (dos actantes; por ejemplo, Susan conoce a Sam) y trivalente (tres actantes; por ejemplo, Sam obsequió aretes a Susan). Gruber [21] y Fillmore [26] definieron algunos roles semánticos actantes como el agente (quien realiza una acción; por ejemplo, el niño corre por la calle); experimentador (el cual experimenta una acción o evento; por ejemplo, el bebé cae); paciente (el cual sufre el efecto de un evento; por ejemplo, persona sufre mareos); beneficiario (quien se favorece de una acción; por ejemplo, profesor instruye al alumno, en esta oración el profesor es el agente y alumno es el beneficiario).

**2) Circunstanciales:** Los circunstanciales se conocen como complementos circunstanciales o funciones adverbiales que expanden el significado de la oración. El número de circunstanciales que pueden aparecer en una oración es ilimitado, ya que los circunstanciales no se restringen con la valencia verbal [20]. Gruber [21] definió algunos roles semánticos de circunstanciales que permiten representar eventos como: fuerza (la cual es de origen desconocido por causa de un evento; por ejemplo, el aumento del viento cierra la puerta); causa (situación que se genera por un evento; por ejemplo, la roca se derrite).

### D. Dominios de software científico

Los dominios de software científico representan el conocimiento de sistemas de software que implican componentes científicos. Estos dominios, generalmente, se utilizan para la resolución de problemas y descripción de fenómenos e implican contenido de información en áreas de ingeniería, medicina, ciencias y matemáticas [18]. Para la representación de eventos en dominios de software científico se utilizan, además del verbo, roles semánticos que lo acompañan. Entre estos roles se incluyen de cero a dos actantes, los cuales pueden ser de tipo experimentador o paciente, y circunstanciales de fuerza o causa.

### III. TRABAJO PREVIO

Algunos enfoques de representaciones en el modelado computacional permiten la representación de la vista comportamental de un sistema de software: diagramas de actividades [3] y diagramas de máquinas de estado [5] del lenguaje unificado de modelado (UML), modelo de información común (CIM) para representar eventos [7], diagramas de máquinas de estado SysML [8], Grafo de cadena de procesos basado en eventos (Light EPC Graph) [9], plataforma de procesamiento de eventos para la gestión de procesos [10], notación de modelo de procesos de negocio (BPMN) [11], cadena de procesos basado en eventos (EPC) [12], patrones de tiempo para procesos [13] y modelos de Markov [14]. Los anteriores enfoques incluyen eventos en su representación. Sin embargo, les hace falta la vista estructural del dominio. Además, se utilizan en dominios organizacionales, a excepción de los patrones de tiempo para procesos [13] que presenta elementos del dominio científico. También se presentan errores lingüísticos en la notación de eventos: los eventos se suelen confundir con procesos o con objetos [7], [9-10] y [12]; los verbos eventuales se conjugan en diferentes tiempos en la misma representación [7-11], [13]; y algunas de las vistas no presentan estructuras que deben acompañar al verbo eventual [8], [10], [13], [15].

Otros enfoques permiten la representación de la vista estructural de un sistema de software: la ontología de eventos basada en el conocimiento simple de sistemas organizacionales (SKOS) [15], el modelo básico de eventos (BEM) [16] y la ontología de patrones para el modelado de eventos [17]. Estos enfoques también incluyen eventos en su representación, pero les hace falta la vista comportamental del dominio y se utilizan en dominios organizacionales a excepción de la propuesta de Liu et al. [17].

El EP permite integrar unidad y consistencia entre las características estructurales y dinámicas de un dominio, a diferencia de las demás representaciones. Adicionalmente, presenta estructuras lingüísticas y gráficas para representar tres tipos de eventos (ver la Figura 2): Evento disparador de tipo condicional (a) y declaración (b) y evento de resultado (c).

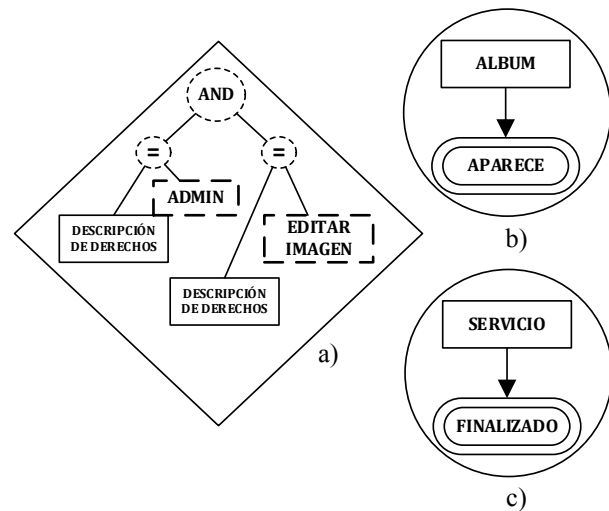


Fig. 2. Eventos en el EP [6].

El EP incluye relaciones eventuales para representar el evento disparador de tipo declaración en dominios organizacionales (por ejemplo, llegar, venir, emerger, tintinear, volar, pasar, mentir, sentar, aparecer, dormir, nevar, morir y hervir) y un concepto que acompaña la relación eventual; también tiene estructuras gráficas, como un concepto y el doble óvalo para la relación eventual, que se conectan mediante el enlace “conexión” y se integran en un círculo, al igual que el evento de resultado. La relación eventual para el evento de resultado es el verbo de la relación dinámica que implica al evento en participio pasado. El evento disparador tipo condicional contiene operadores lógicos y relacionales con conceptos y valores que se asignan al concepto y que se integran en un rombo.

### IV. PLANTEAMIENTO DEL PROBLEMA

El esquema preconceptual incluye estructuras lingüísticas para representar eventos disparadores tipo declaración en dominios organizacionales: una relación eventual y un concepto. Estas estructuras lingüísticas tienen estructuras gráficas que permiten su representación [6]. A pesar de ello, el EP no contiene estructuras lingüísticas para representar eventos disparadores tipo declaración en dominios de software científico, ya que le faltan relaciones eventuales que se incluyen en este dominio. Además de las relaciones eventuales, en los dominios científicos se utilizan entre cero y dos actantes para los objetos y actantes en los que recae el evento, pero con las estructuras

existentes del EP sólo se pueden representar eventos disparadores tipo declaración con un actante. Por ello se requieren estructuras lingüísticas y gráficas que relacionen los objetos y actores que están presentes en los eventos y las relaciones eventuales que se utilizan en dominios de software científico.

## V. PROPUESTA DE SOLUCIÓN

En este capítulo se propone una extensión al esquema preconceptual a partir de estructuras lingüísticas basadas en roles semánticos, para representar eventos en dominios de software científico. La extensión se desarrolla en cuatro fases:

### A. Identificación de eventos en dominios de software científico

En la Tabla 1 se caracterizan las relaciones eventuales para representar eventos en dominios de software científico y se presentan ejemplos de su uso; por ejemplo, la relación eventual “erupcionar” se extrae del dominio de vulcanología [27] y la relación eventual “sangrar” del dominio de medicina [28]. En el análisis sintáctico se encuentra que las relaciones eventuales “tronar”, “granizar” y “nevar” son del mismo tipo de actante del verbo “llover” que formula Tesnière [20]; las relaciones eventuales “convulsionar” y “estornudar” son del mismo tipo de actante de la relación “sangrar”; la relación eventual “crecer” es del mismo tipo de actante de la relación “incrementar”, y las relaciones eventuales “perder” y “ganar” son del mismo tipo de la relación “aumentar”.

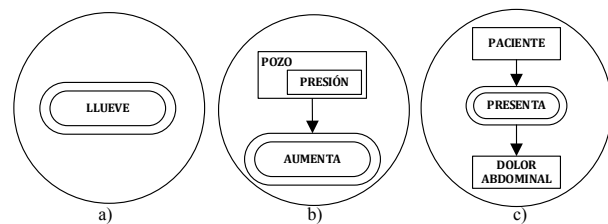
### B. Integración de estructuras lingüísticas para relaciones eventuales

A partir de roles semánticos (entre cero y dos actantes de tipo experimentador o paciente y circunstantes de tipo causa o fuerza), se integran relaciones eventuales al EP para representar eventos en dominios de software científico que se relacionan en la Tabla 1; la relación eventual “nevar” no tiene ningún actante, ya que el mismo verbo indica el evento sin la compañía de un rol semántico. Por ello, cuando se conjuga como evento en el dominio de meteorología, su forma es “nieva”. La relación eventual “erupcionar” tiene un actante y cuando se conjuga como evento su forma es “volcán

erupciona”. Esta oración indica un evento que sucede en un dominio de software científico, el cual puede ser vulcanología. El concepto “volcán” es un actante de tipo experimentador, ya que el volcán experimenta la erupción, y tiene dos circunstantes (uno de fuerza y otro de causa), porque la erupción se produce por el incremento de la presión y es también la causa de la combinación de la presión con los gases de la tierra. La relación eventual “perder” tiene dos actantes, ya que ambos le dan un sentido completo a la oración, y cuando se conjuga como evento desde el dominio de software científico de medicina su forma es “paciente pierde peso”. En este caso el primer actante es de tipo paciente, porque el paciente es quien sufre el evento de perder peso y tiene como circunstante una causa, ya que la pérdida de peso puede ser resultado de una enfermedad.

### C. Representación de eventos

En esta fase se agregan al EP estructuras gráficas que se relacionan con las estructuras lingüísticas propuestas, para representar eventos disparadores de tipo declaración en dominios de software científico. En la Figura 3 se presentan estas estructuras. En la Figura 3a se representa el evento disparador tipo declaración “llueve”, que sólo contiene la relación eventual “llueve” sin ningún actante. En la Figura 3b se representa el evento disparador tipo declaración “presión del pozo aumenta”, que contiene la relación eventual “aumenta” junto con un actante “presión del pozo”, donde “presión” es un concepto hoja (atributo) del concepto clase “pozo”. En la Figura 3c se representa el evento disparador tipo declaración “paciente presenta dolor abdominal”, que contiene la relación eventual “presenta” con dos actantes: “paciente” y “dolor abdominal”. El actante es de tipo paciente, ya que experimenta un dolor abdominal.



**Fig. 3.** Estructuras lingüísticas para representar eventos disparadores tipo declaración en dominios de software científico (Elaboración propia).



**Tabla 1.** Estructuras lingüísticas para representar eventos disparadores de tipo declaración en dominios de software científico

Relación eventual	Evento ejemplo	Dominio de software científico	Roles semánticos		
			Actante		Circunstante
			Cantidad	Tipo	
Llover [20]	Llueve	Meteorología, Climatología	0	No requiere rol semántico	Causa
Tronar	Truena	Meteorología, Climatología	0	No requiere rol semántico	Causa
Granizar	Graniza	Meteorología, Climatología	0	No requiere rol semántico	Causa
Nevar	Nieva	Meteorología, Climatología	0	No requiere rol semántico	Causa
Subir [29]	Voltaje sube	Eléctrica y electrónica	1	Experimentador	Fuerza, causa
Erupcionar [26]	Volcán erupciona	Vulcanología y geología	1	Experimentador	Fuerza, causa
Sangrar [27]	Paciente sangra	Medicina	1	Paciente	Causa
Convulsionar	Paciente convulsiona	Medicina	1	Paciente	Causa
Disminuir [30]	Precio del dólar disminuye	Administración, finanzas y economía	1	Experimentador	Causa
Incrementar [30]	Precio del dólar incrementa	Administración, finanzas y economía	1	Experimentador	Causa
Crecer	Tasa de población crece	Física y Estadística	1	Experimentador	Causa
Iniciar [31]	Servicio inicia	Administración	1	Experimentador	Causa
Sonar [11]	Alarma suena	Automatización	1	Experimentador	Causa
Caer [21]	Rayo cae	Meteorología, Climatología	1	Experimentador	Fuerza, Causa
Expirar [11]	Producto expira	Economía	1	Experimentador	Causa
Llegar [6]	Colesterol llega	Química	1	Experimentador	Causa
Emerger [6]	Bacteria emerge	Bacteriología	1	Experimentador	Causa
Venir [6]	Señal viene	Ingeniería de control	1	Experimentador	Causa
Tintinear [6]	El cascabel de la serpiente tintinea	Biología	1	Experimentador	Causa
Volar [6]	Abeja africana vuela	Zootecnia	1	Experimentador	Causa
Pasar [6]	Tiempo pasa	Meteorología	1	Experimentador	Causa



Relación eventual	Evento ejemplo	Dominio de software científico	Roles semánticos		
			Actante		Circunstante
			Cantidad	Tipo	
Hervir [6]	Magma hierve	Geología	1	Experimentador	Causa
Aparecer[6]	Onda aparece	Electrónica	1	Paciente	Causa
Dormir [6, 21]	Paciente duerme	Medicina	1	Paciente	Causa
Estornudar [6]	Paciente estornuda	Medicina	1	Paciente	Causa
Morir [6]	Animal muere	Zootecnia	1	Paciente	Causa
Sufrir [21]	Paciente sufre hemorragia	Biomédica	2	Paciente	Causa
Presentar [29]	Paciente presenta dolor abdominal	Biomédica	2	Paciente	Causa
Obstruir [29]	Lípido obstruye vena	Medicina	2	Experimentador	Causa
Aumentar [30]	Temperatura aumenta presión del agua	Hidráulica	2	Experimentador	Causa
Perder	Paciente pierde peso	Biomédica	2	Paciente	Causa
Ganar	Paciente gana peso	Biomédica	2	Paciente	Causa

(Elaboración propia).

La extensión al EP de estructuras lingüísticas y gráficas permite la representación de un evento en varias formas. Por ejemplo, el evento disparador tipo declaración “paciente sufre hemorragia” en la Figura 4a, con la relación eventual “sufre” y dos actantes “paciente” y “hemorragia”, se puede representar en la forma “paciente sangra” de la Figura 4b con la relación eventual “sangra” y el actante “paciente”, o en la forma “hemorragia aparece” de la Figura 4c con la relación eventual “aparece” y el actante “hemorragia”. Con estas representaciones no se pierde el significado de la frase que indica el evento en el dominio de software científico de medicina. También el evento “dólar gana precio” de la Figura 4d se puede representar en la forma “precio del dólar incrementa” de la Figura 4e sin alterar su significado.

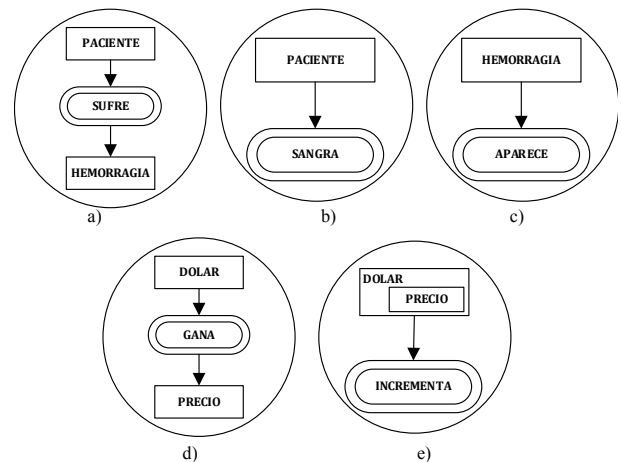


Fig. 4. Representación de eventos disparadores tipo declaración (Elaboración propia).

#### D. Extensión al EP para representar eventos en dominios de software científico

Se integran las estructuras lingüísticas y gráficas al EP para representar eventos en dominios de software científico, y se representan los eventos en un EP a partir del dominio de software científico vulcanología como caso práctico en la erupción de un volcán.

#### IV. RESULTADOS

En la Figura 5 se representa el EP de la erupción de un volcán. En la parte estructural del EP, que se identifica con la relación estructural “tiene”, se evidencian las relaciones entre los conceptos. La “tierra” es un concepto clase que tiene “punto caliente” (zonas con un alto porcentaje de ascenso magmático). El concepto clase “punto caliente” tiene conceptos hoja como: “id”, “ubicación” y “temperatura”. También la “tierra” tiene el concepto hoja “roca” y el concepto clase “volcán”. Este concepto clase tiene los siguientes conceptos hoja: “id”, “nombre”, “ubicación”, “gas”, “presión” y “magma” (rocas que se funden a muy elevada temperatura y presión, y que pueden fluir al

exterior a través de un volcán). Este concepto clase tiene conceptos hoja “id” y “viscosidad”, que puede ser “alta” o “baja”.

La parte dinámica del EP inicia con el flujo de los eventos a partir de la generación de tres eventos disparadores: dos eventos disparadores tipo condicional, “temperatura del punto caliente >700 °C” y “ubicación del punto caliente = ubicación del volcán”, junto con el evento disparador tipo declaración “temperatura del punto caliente aumenta” que se unen con un “joint”. Estos tres eventos disparan el evento declaración “magma aparece”. Si “la viscosidad del magma = baja” se dispara el evento declaración “presión de la tierra incrementa”. Si estos dos eventos, junto con el evento declaración “gas sube” inician, pueden producir el evento declaración “volcán erupciona”.

En el EP de la Figura 5 se pueden observar las estructuras lingüísticas y gráficas que se integran a la extensión del EP como relaciones eventuales: “aumenta”, “erupciona”, “incrementa”, “sube” y “aparece”.

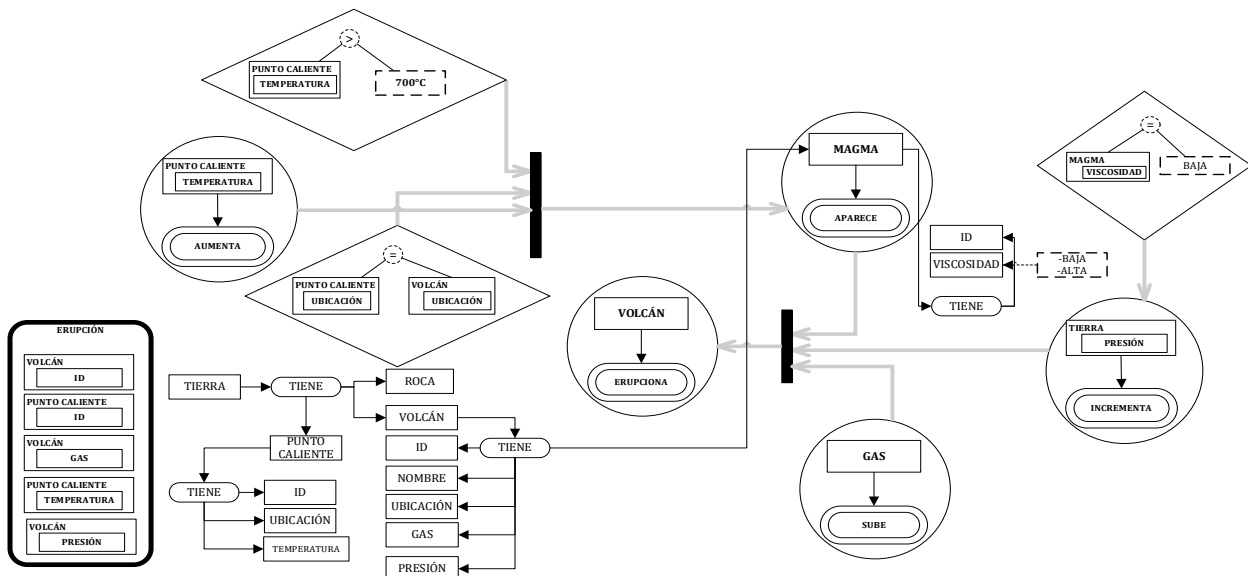


Fig. 5. Representación de eventos en EP en dominios de software científico (Elaboración propia).

La estructura “marco” del EP (véase la Figura 1) sirve para representar los informes en un sistema de software. En el EP de la Figura 5 se presenta un marco para informar acerca de una “erupción” con información del “id” del “volcán”, “id” del “punto caliente”, el “gas” del “volcán”, la “temperatura del volcán” y la “presión” del “volcán”, lo que permite representar un sistema de software en un dominio científico.

A partir de la información del marco y de la parte estructural del EP, se puede validar el funcionamiento de la representación de eventos en dominios de software científico utilizando tablas de datos. En la Tabla 2 se presenta el concepto clase “punto caliente”, que guarda el nombre de los puntos calientes de la “tierra” y su “ubicación”, como el punto “Yellowstone”, cuya ubicación es “44°38'00"N 110°10'00"O”.

**Tabla 2.** Tabla de datos del “punto caliente”

Punto Caliente		
ID	Nombre	Ubicación
101	Yellowstone	44°38'00"N 110°10'00"O
102	Canarias	28°28'00"N 16°15'00"O
103	Jan Mayen	70°59'00"N 8°32'00"O

Fuente: (Elaboración propia).

En la Tabla 3 se incluye el concepto clase “volcán” que guarda el nombre y la ubicación del volcán, como el volcán “Timanfaya” cuya ubicación es “28°28'00"N 16°15'00"O”.

**Tabla 3** Tabla de datos del “volcán”

Volcán		
ID	Nombre	Ubicación
V1	Caldera de Yellowstone	44°38'00"N 110°10'00"O
V2	Timanfaya	28°28'00"N 16°15'00"O
V3	Jan Mayen	70°59'00"N 8°32'00"O

Fuente: (Elaboración propia).

En la Tabla 4 se presenta el concepto clase “magma” que guarda el “id” del “volcán” y la “viscosidad” del “magma”, como el “V1”, que es el “id” del “volcán” “Caldera de Yellowstone” cuya “viscosidad” es “baja”.

**Tabla 4.** Tabla de datos del “magma”

Magma		
ID	ID Volcán	Viscosidad
M1	V1	Baja
M2	V2	Baja
M3	V3	Baja

Fuente: (Elaboración propia).

Finalmente, en la Tabla 5 se integra el informe “erupción”, que incluye el “id” del “volcán”, el “id” del “punto caliente”, el “gas”, la “temperatura” y la “presión” del “volcán”. El “id” del “volcán” “V2” identifica al volcán “Timanfaya” que se encuentra ubicado en el “punto caliente” “Canarias” con “id” “102”, esto quiere decir que tienen la misma “ubicación” y se cumple la condición “ubicación del punto caliente = ubicación del volcán” con un “40%” de “gas” que incrementa a “780 °C” que hacen cumplir la condición “temperatura del punto caliente >700 °C”; como la “viscosidad” del “magma” del “volcán” es “baja”, induce a que se incremente la “presión” a un valor de “28.000 Pa (Pascales)”.

**Tabla 5.** Tabla de datos del informe “erupción”

Erupción				
ID Volcán	ID Punto caliente	Gas	Temperatura	Presión
V1	101	60 %	1.200 °C	39.800 Pa
V2	102	40 %	780 °C	28.000 Pa
V3	103	30 %	705 °C	19.000 Pa

Fuente: (Elaboración propia).

## VII. CONCLUSIONES Y TRABAJO FUTURO

El uso de la extensión al EP para representar eventos a partir de estructuras lingüísticas basadas en roles semánticos permite la comunicación e interacción entre el modelado computacional y científico, ya que posibilita la identificación de funciones de actantes y circunstancias que deben acompañar la relación eventual en dominios de software científico. Además, facilita la representación de eventos en procesos de desarrollo de sistemas de software en estos dominios.

Mediante el conjunto de estructuras que se presenta se pueden caracterizar eventos en dominios de software científico, considerando que el EP permite integrar unidad y consistencia entre las características estructurales y dinámicas de un dominio, a diferencia de las demás representaciones. Analistas científicos y de negocios pueden utilizar el EP para representar dominios de software científico mediante el uso de las nuevas estructuras, debido a que el EP es una herramienta que facilita la comprensión de los elementos de un dominio a los analistas e interesados y permite representarlos.

Como trabajo futuro se pueden relacionar con estructuras matemáticas para representar eventos en dominios de software científico.

## AGRADECIMIENTOS

Este capítulo es producto del proyecto de investigación Representación de eventos en esquemas preconceptuales a partir de roles semánticos y ecuaciones matemáticas, con código Hermes 39365, entre el Tecnológico de Antioquia - Institución Universitaria y la Universidad Nacional de Colombia. Este proyecto surge a partir de la tesis doctoral “Una extensión a esquemas preconceptuales para el refinamiento en la representación de eventos y la notación matemática”, con código Hermes 39886, que financia Colciencias en la convocatoria 727 de becas para estudiantes de doctorado en Colombia.

## REFERENCIAS

- [1] P. A. Noreña y C. M. Zapata, “A Game for Learning Event-Driven Architecture: Pre-conceptual-Schema-based Pedagogical Strategy”, *Development in Business Simulation and Experiential Learning*, vol. 45, pp. 312-319, 2018.
- [2] P. A. Noreña y C. M. Zapata, “A Pre-conceptual-Schema-based Representation of Time Events Coming from Scientific Software Domain”, en *Proceedings of the 22nd World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2018, Orlando, USA, 2018*, pp. 53-58.
- [3] Object Management Group (OMG). “Superstructure 2.5”, 2015. Disponible en: <http://www.omg.org/spec/UML/2.5/>
- [4] P. A. Noreña, F. Vargas y D. Soto, “Tipificación de eventos a partir del modelo BPMN en artefactos de ingeniería de software”, *Cuaderno Activa*, vol. 6, pp. 49-61, 2014.
- [5] OMG. Object Management Group, “Business Process Model and Notation BPMN”, Standard Document. [Online]. Disponible en: [http://www.omg.org/spec/BPMN/1.2.](http://www.omg.org/spec/BPMN/1.2/) [Accedido: 18 de abril de 2016].
- [6] C. M. Zapata, *The UNC-Method revisited: Elements of the new approach*. Saarbrücken: Lambert, 2012.
- [7] G. Ravikumar, S. A. Khaparde y R. K. Joshi, “Integration of Process Model and CIM to Represent Events and Chronology in Power System Processes”, *IEEE Systems Journal*, vol. 12, n.º 1, pp. 149-160, 2018.
- [8] A. Baouya, D. Bennouar, O. A. Mohamed y S. Ouchani, “A probabilistic and timed verification approach of SysML state machine diagram”, en *Proceedings of 12th International Symposium on Programming and Systems (ISPS)*, Algiers, Algeria, 2015, pp. 1-9.
- [9] S. Xue, B. Wu y J. Chen, “LightEPC: A Formal Approach for Modeling Personalized Lightweight Event-Driven Business Process”, en *Proceedings of IEEE International Conference on Services Computing (SCC)*, Santa Clara, CA, USA, 2013, pp. 1-8.
- [10] N. Herzberg, A. Meyer y M. Weske, “An event processing platform for business process management”, *Proceedings of 17th IEEE International of Enterprise Distributed Object*

- Computing Conference (EDOC), Vancouver, Canada, 2013, pp. 107-116.
- [11] C. Haisjackl, P. Soffer, S. Lim y B. Weber, “How do humans inspect BPMN models: an exploratory study?”, *Software and Systems Modeling*, vol. 17, n.º 2, pp. 655-673, 2018.
- [12] W. Xia y M. Junpeng, “Research on Flexible Business Process of Bank Modeling Based on EPC”, en *Proceedings of International Conference Management of e-Commerce and e-Government (ICMeCG)*, Shanghai, China, 2014, pp. 54-60.
- [13] A. Lanz, B. Weber y M. Reichert, “Time patterns for process-aware information systems”, *Requirements Engineering*, vol. 19, n.º 2, pp. 113-141, 2014.
- [14] W. Chen, J. Wang, D. Shi y L. Shi, “Event-Based State Estimation of Hidden Markov Models Through a Gilbert–Elliott Channel”, *IEEE Transactions on Automatic Control*, vol. 62, n.º 7, pp. 3626-3633, 2017.
- [15] Y. Zhang, W. Liu, N. Ding, X. Wang y Y. Tan, “An Event Ontology Description Framework Based on SKOS”, en *Proceedings of 12th Intl Conf on Ubiquitous Intelligence and 15th Intl Conf on Scalable Computing and Communications and its Associated Workshops (UIC-ATC-ScalCom)*, Beijing, China, 2015, pp. 1774-1779.
- [16] Y. Wang, X. Zhou, L. Shan y K. Miao, “Study on Complex Event Processing for CPS: An Event Model Perspective”, en *Proceedings of 11th Intl Conf on Ubiquitous Intelligence and Computing and 14th Intl Conf on Scalable Computing and Communications and its Associated Workshops*, Bali, Indonesia, 2014, pp. 305-310.
- [17] W. Liu, D. Wang, Y. Wu, Y. Liu y Y. Zheng, “A software engineering practice platform based on event ontology”, en *7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, Australia, 2012, pp. 1474-1479.
- [18] D. Kelly, “Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software”, *Journal of Systems and Software*, vol. 109, pp. 50-61, 2015.
- [19] P. Moreda, “Los roles semánticos en la Tecnología del lenguaje Humano: Anotación y aplicación”, tesis doctoral, Universidad de Alicante, Alicante, España, 2008.
- [20] L. Tesnière, *Éléments de syntaxe structural*. Paris: Klincksieck, 1966.
- [21] J. Gruber, “Studies in lexical relations”, tesis doctoral, Massachusetts Institute of Technology, Boston, USA, 1965.
- [22] P. A. Noreña, “Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method”, tesis de maestría, Universidad Nacional de Colombia, Medellín, Colombia, 2013.
- [23] P. A. Noreña y C. M. Zapata, “Una representación basada en esquemas preconceptuales de eventos determinísticos y aleatorios tipo señal desde dominios de software científico”, en *Memorias del 10.º Congreso Mexicano de Inteligencia Artificial, COMIA 2018*. Mérida, México, pp. 1-14.
- [24] P. A. Noreña, D. M. Torres y C. M. Zapata, “Interoperabilidad dinámica entre sistemas basados en internet de las cosas: una representación a partir de esquemas preconceptuales”, en *Industria 4.0 Escenarios e impactos*, Medellín: Sello Editorial Universidad de Medellín, 2017, pp. 159-173.
- [25] T. Payne, *Describing morphosyntax: A guide for field linguists*, Cambridge, New York: Cambridge University Press, 1997.
- [26] C. Fillmore, “The case for case reopened”, *Syntax and Semantics*, 1977.
- [27] R. Mora y M. Pirullina, “Deslizamiento Las Torres, volcán Irazú, Costa Rica: definición de su volumen y su dirección preferencial de movimiento”, *Revista Geológica de América Central*, vol. 56, pp. 51-67, 2017.
- [28] T. H. Wu, G. K. H. Pang y E. W. Y. Kwong, “Predicting Systolic Blood Pressure Using Machine Learning”, en *Proceedings of 7th International Conference on Information and Automation for Sustainability*, Colombo, Sri Lanka, 2014, pp. 1-6.
- [29] V. Barrera, R. Velandia, F. Hernández, J. Meléndez y H. Vargas, “Atributos Relevantes para el Diagnóstico Automático de Eventos”, *Revista Iberoamericana de Automática e Informática industrial*, vol. 10, n.º 1, pp. 73-84, 2013.
- [30] P. A. Beltrán, “Precio del petróleo y el ajuste de las tasas de interés en las economías emergentes”, *Borradores de Economía*, vol. 901, pp. 1-37, 2015.
- [31] K. Zhang y L. Zhang, “Observability of Boolean Control Networks: A Unified Approach Based on Finite Automata”, *IEEE Transactions on Automatic Control*, vol. 61, n.º 9, pp. 2733-2738, 2016.





## Capítulo VII

# REPRESENTACIÓN EN EL NÚCLEO DE SEMAT DEL MODELO MADCE-TVD

*Adriana Xiomara Reyes Gamboa - axreyes@elpoli.edu.co*

Politécnico Colombiano Jaime Isaza Cadavid

*Jovani Alberto Jiménez Builes - jajimen1@unal.edu.co*

Universidad Nacional de Colombia

*Darío Enrique Soto Durán - dsoto@tdea.edu.co*

Tecnológico de Antioquia

### I. INTRODUCCIÓN

La ingeniería de software [1] busca la estandarización y normalización de los procesos y una base común de elementos, que se constituyan en un núcleo que facilite a los analistas utilizarlo en diferentes fases del ciclo de vida, como un conjunto de métodos y buenas prácticas que garanticen la calidad del producto de software.

Partiendo de esta estandarización y búsqueda de un núcleo común, es que iniciativas como SEMAT [2] buscan apoyar un proceso para redefinir la ingeniería de software, basado en una teoría sólida, principios probados y mejores prácticas que incluyan un núcleo de elementos ampliamente aceptados y que se pueda extender a usos específicos, traten asuntos tecnológicos y humanos, sean apoyados por la industria, la academia, los investigadores y los usuarios y apoyen la extensión ante los requisitos cambiantes y la tecnología. SEMAT incluye un núcleo de elementos ampliamente aceptados por la comunidad de desarrollo de software, extensible a otros usos específicos y un lenguaje sencillo para describir métodos y prácticas, buscando la estandarización y normalización de procesos.

De igual forma el Modelo MADCE-TVD [3] busca garantizar la aplicación de buenas prácticas al

momento de desarrollar contenidos educativos para televisión digital (TVD), estos contenidos educativos para la TVD se denominan contenidos para t-learning [4]. El propósito de este modelo es garantizar la calidad de los contenidos desarrollados.

Estas iniciativas buscan impactar en la calidad del producto de software, según Tomala y Jasmin [5] la calidad es el grado en el que un conjunto de características cumple con los requisitos, lo que indica la satisfacción del interesado en el producto de software.

El Modelo MADCE-TVD aplica prácticas ágiles para el desarrollo de los contenidos educativos para TVD organizados en tres fases: pre-producción, producción y post-producción. En el artículo se propone una representación en el núcleo de SEMAT del modelo MADCE-TVD.

El artículo se estructura de la siguiente forma: en la sección 2 se presentan los conceptos más relevantes de SEMAT y del modelo MADCE-TVD; en la sección 3 se describe la representación gráfica de MADCE-TVD con SEMAT, y en la sección 4 se presentan las conclusiones y trabajos futuros.

## II. MARCO TEÓRICO

### A. SEMAT (Software Engineering Method and Theory)

SEMAT apoya un proceso para redefinir la ingeniería de software basado en una teoría sólida, principios probados y mejores prácticas que [2]:

- Incluyan un núcleo de elementos ampliamente aceptados y que se pueda extender a usos específicos.
- Traten asuntos tecnológicos y humanos.
- Los apoyen la industria, la academia, los investigadores y los usuarios.
- Apoyen la extensión ante los requisitos cambiantes y la tecnología.

SEMAT se basa en una teoría sólida, con principios aprobados y buenas prácticas para apoyar el proceso de redefinir la ingeniería de software. Incluye un núcleo de elementos y un lenguaje sencillo para describir métodos y prácticas [6].

El núcleo de SEMAT contiene un grupo pequeño de “cosas con las que siempre se trabaja” llamadas alfas: oportunidad, interesado, requisitos, sistema de software, trabajo, equipo y forma de trabajo, y “cosas que siempre se hacen” llamadas espacios de actividad como explorar posibilidad, comprender requisitos y prepararse para el trabajo, entre otros, para el desarrollo de sistemas de software [7] (ver Fig.1).

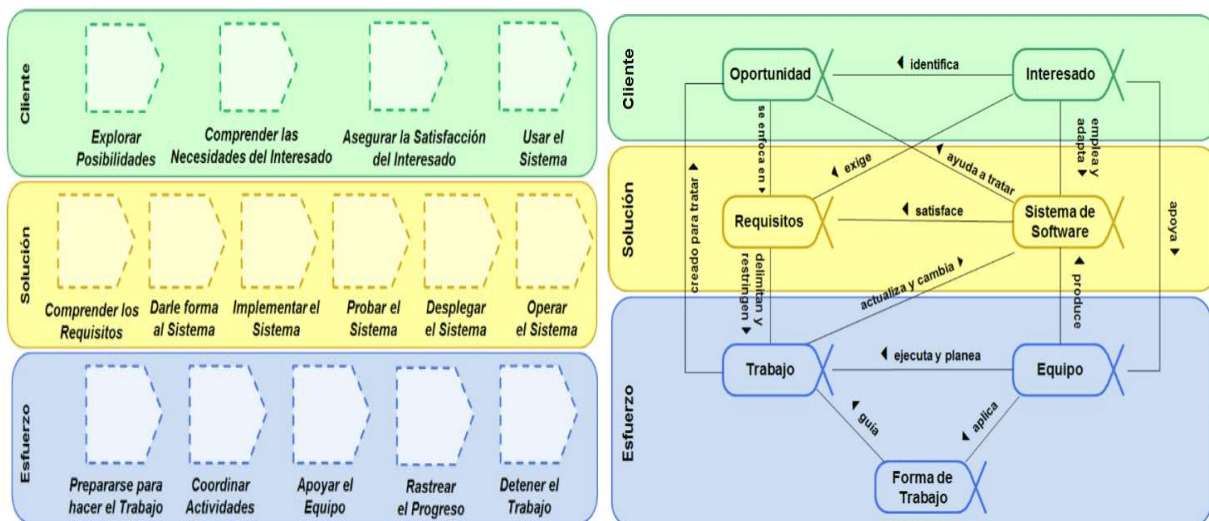

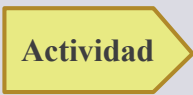


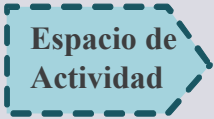



Fig. 1. Alfas y Espacios de Actividad del proceso de desarrollo de software [2]

SEMAT permite organizar los aspectos de interés en tres áreas: clientes (color verde), solución (color amarillo) y esfuerzo (color azul).

Los elementos del núcleo de SEMAT corresponden a las alfas, espacios de actividad, actividades, prácticas, métodos, patrones y productos de trabajo como se presenta en la (Tabla I) [2].

**Tabla 1.** Elementos del Núcleo de SEMAT [3].

Elemento	Descripción
 Alfa	Describe las cosas que el equipo debe administrar, producir y usar en el proceso de desarrollo, mantenimiento y soporte. SEMAT identifica siete alfas: oportunidad, interesado, requisitos, sistema de software, trabajo, equipo y forma de trabajo.
 Actividad	Define uno o más tipos de producto de trabajo y uno o más tipos de tarea, además de dar orientación sobre cómo utilizarlos en el contexto de una práctica.
 Patrón	Es una descripción de la estructura de una práctica.
 Práctica	Representa las cosas esenciales que se deben hacer para desarrollar el software. SEMAT define 15 espacios de actividad.
 Espacio de Actividad	Es un grupo de elementos necesarios del núcleo de SEMAT para expresar la guía de trabajo con un objetivo específico.
 Producto de Trabajo	Es un artefacto de valor y relevancia para el esfuerzo de la ingeniería de software. Un producto de trabajo puede ser un documento, una parte de software, una prueba de software o la preparación de un curso.

## B. MADCE-TVD: Modelo Ágil para el Desarrollo de Contenidos Educativos para t-learning

MADCE-TVD, el modelo ágil para el desarrollo de contenidos educativos para t-learning, se centra en los procesos ágiles de desarrollo de software y en los procesos técnicos, pedagógicos y de personalización. Abarca las características y funciones necesarias en el proceso t-learning a través de TVD.

Los requisitos técnicos incluyen requerimientos funcionales, de usabilidad y atributos de calidad.

Los requerimientos personales inciden en qué tipo de soluciones técnicas y pedagógicas se deben utilizar para satisfacer los requerimientos personales y el análisis de los requerimientos pedagógicos que se deben considerar para un entorno basado en la televisión, y para esto se ha considerado importante definir el modelo siguiendo tres fases: preproducción, producción y postproducción.

**1) Preproducción:** En esta fase se realiza:

- La reunión de planeación con el cliente.
- El levantamiento de las historias de usuario, que definen el alcance del contenido que se desea por parte del cliente o persona interesada.
- Se identifican las competencias para alcanzar con el contenido desarrollado.
- Se define el equipo de personas que participarán en el proceso de producción.
- Se identifican las herramientas que se van a utilizar para la construcción
- Se identifica el público objetivo y su contexto.

**2) Producción:** La fase de Producción centra su gestión siguiendo los valores y principios de SCRUM [8], y el desarrollo se realiza siguiendo las prácticas del Desarrollo Guiado por Comportamiento (BDD, por sus siglas en inglés) [9]. El proceso se lleva a cabo de forma iterativa e incremental. Para esta fase se definen unas plantillas que permiten el control del proceso que se realiza durante las iteraciones y revisiones.

**Iteraciones:** Las iteraciones siguen la definición dada por SCRUM para los sprints y cada una tiene una duración fija entre 2 y 4 semanas máximo, en cada una de ellas las entradas son las historias de usuario seleccionadas para ser desarrolladas en cada sprint. Cada día se efectúa una reunión de máximo 15 minutos para hacer una revisión del avance del trabajo, y al final de cada iteración se realiza una entrega de un producto funcional que en cada sprint se va incrementando.

El proceso de desarrollo del contenido educativo se realiza en cada iteración siguiendo la filosofía BDD, se

trata de un proceso que amplía las ideas del Desarrollo Guiado por Pruebas (TDD) [10] y las combina con otras ideas de diseño de software y análisis de negocios para proporcionar un proceso (y una serie de herramientas) a los desarrolladores, con la intención de mejorar el desarrollo del software. BDD se basa en TDD y formaliza las mejores prácticas de TDD, clarificando cuáles son y haciendo énfasis en ellas.

En BDD no se prueban solo unidades o clases, se prueban escenarios y el comportamiento de las clases a la hora de cumplir dichos escenarios, los cuales pueden estar compuestos de varias clases. Esto es fundamental en el contexto del t-learning, ya que lo que se busca de este contenido es que cumpla con un comportamiento específico, y ello se puede garantizar al aplicar este proceso.

Algunos de los hábitos que fomenta BDD, añadiéndose a los que aporta TDD, son:

- Ayuda a centrar el desarrollo en lo que es verdaderamente importante para el “negocio”.
- Si se generan las pruebas con un lenguaje concreto, pueden servir a la hora de hacer los test de Aceptación, lo cual da garantías de que lo desarrollado satisface las necesidades del cliente o interesado.

A la hora de llevar a la práctica BDD, es muy recomendable usar un lenguaje común para hacer los test. El lenguaje propuesto para el método es:

**Dado:** los pasos necesarios para poner al sistema en el estado que se desea probar.

**Cuando:** la interacción del usuario que acciona la funcionalidad que deseamos probar.

**Entonces:** en este paso vamos a observar los cambios en el sistema y ver si son los deseados.

Todo esto sirve para comprobar el comportamiento interno de la aplicación. Una vez esté la funcionalidad hecha, se debe saber si es lo que buscaba realmente el usuario, y para ello existen las pruebas de aceptación.

La revisión del producto se realiza cada vez que se desarrolla, al aplicar las prácticas de BDD y al final de cada sprint.

### 3) Postproducción

**Cierre:** En esta etapa se realiza la entrega del producto, se entregan los manuales y la documentación necesaria, así como el despliegue en el entorno final.

**Lecciones aprendidas:** Esta etapa permite revisar el proceso realizado en todas las etapas del proyecto, las fortalezas y debilidades identificadas en cada una de las fases y las estrategias aplicadas y el resultado de su aplicación, para poder identificar qué se debe mejorar y qué estrategias dieron buenos resultados con el fin de buscar la mejora en posteriores proyectos. Para esto se utiliza una plantilla de lecciones aprendidas, cuyo propósito es ayudar al equipo del proyecto a compartir el conocimiento adquirido por medio de la experiencia durante su ejecución, para que toda la organización pueda beneficiarse de la información que ésta provee.

## III. REPRESENTACIÓN EN EL NÚCLEO DE SEMAT

Un aspecto importante para la representación es identificar los elementos constantes de la norma en el núcleo de SEMAT, estos elementos son:

- La práctica
- El área de interés
- El alfa
- El espacio de actividad

### A) Práctica

Se requiere identificar las buenas prácticas asociadas al modelo MADCE-TVD, estas buenas prácticas están organizadas en las tres fases que comprende el modelo: la Preproducción, la Producción y la Postproducción. Estas fases comprenden una serie de buenas prácticas que se enmarcan en la planeación.

La planeación permite organizar y definir aspectos claves relacionados con el interesado, el desarrollo se centra en las buenas prácticas de SCRUM y de BDD; cada uno aborda los principios ágiles [11], a saber:

dividir el trabajo (product backlog, priorización, historias de usuario); dividir en pequeños grupos el equipo (entorno físico, comunicación, roles); dividir el tiempo (iteraciones, reuniones, tiempos fijos), y verificar la calidad del producto (pruebas unitarias, refactorización, integración continua, control de versión, desarrollo de forma iterativa). Todo ello lleva al cierre del proyecto que centra las buenas prácticas en las lecciones aprendidas (Fig. 2).

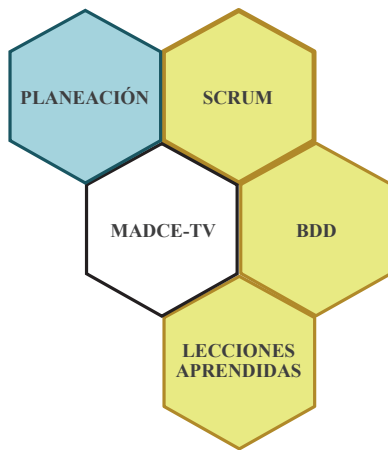


Fig. 2. Representación en el núcleo de SEMAT para MADCE-TVD

### B) Área de Interés

El Modelo MADCE-TVD se enfoca en las tres áreas de interés: Cliente, Solución y Esfuerzo. Estas áreas contienen los elementos que tienen relación con la especificación, el desarrollo del sistema de software, el trabajo y el equipo. (Fig. 3).

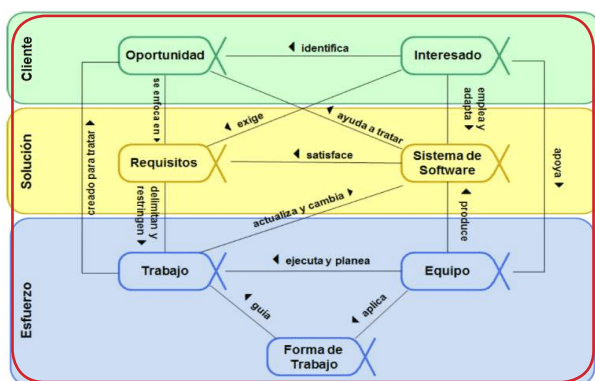


Fig. 3. Áreas de Interés solución de SEMAT

### C) Alfa

El modelo MADCE-TVD se asocia a las cinco alfas: Interesado, Requisitos, Sistema de Software, Trabajo y Equipo, ya que los procesos del modelo están asociados al contenido educativo por desarrollar (Fig. 4).

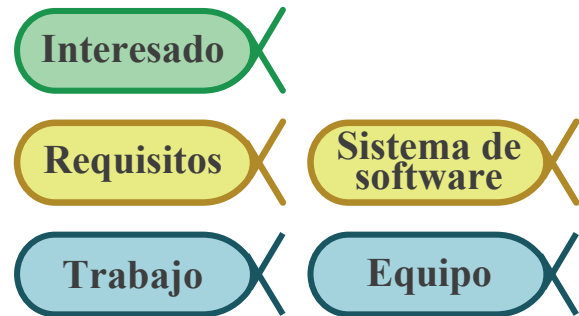


Fig. 4. Alfas asociadas a MADCE-TVD

### D) El Espacio de Actividades

El siguiente paso es identificar el espacio de actividades del modelo. Para este caso, las actividades y productos entregables de los procesos del Modelo MADCE-TVD corresponden a cada una de las fases de Preproducción, Producción y Postproducción.

En la Tabla II se presentan las actividades y productos de trabajo asociados a cada fase del modelo MADCE-TVD.

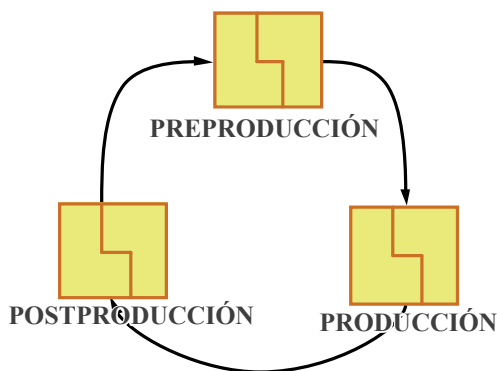
Tabla 2. Actividades y Productos de Trabajo MADCE-TVD

Actividad	Producto de Trabajo
<b>Preproducción</b>	
Definición equipo	Equipo con roles definidos
Identificación Herramientas	Herramientas seleccionadas
Reunión de Planeación	
Levantamiento HU	Backlog del producto
Caracterización público objetivo	Público objetivo definido



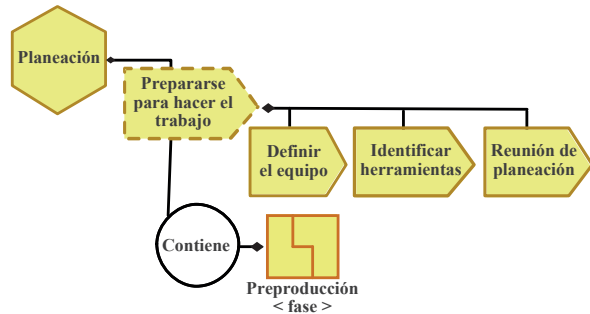
Actividad	Producto de Trabajo
Definición de competencias	Competencias definidas
Organización contenido	Mapa navegacional
<b>Producción</b>	
Definir sprints	<i>Sprint</i>
Seleccionar HU de cada <i>sprint</i>	<i>Backlog del sprint</i>
Reuniones diarias, retrospectiva, sprint	<i>Dashboard</i> Actas de reuniones
Escribir test de comportamiento	Test de comportamiento
Desarrollar código	Código desarrollado
Refactorizar	Código mejorado
Integración continua	Aplicación desarrollada
<b>Postproducción</b>	
Entrega del producto	Acta de entrega
Consolidar y analizar lecciones aprendidas	Plantilla con lecciones aprendidas

Para realizar la representación se identifican las tres fases del modelo, las cuales se representan en la Fig 5, posteriormente se representa cada una de las fases con su espacio de actividades y cada una de las actividades asociadas a ese espacio. Todos los elementos identificados en la Tabla II y todas las prácticas de Planeación, SCRUM, BDD y lecciones aprendidas se representan allí.



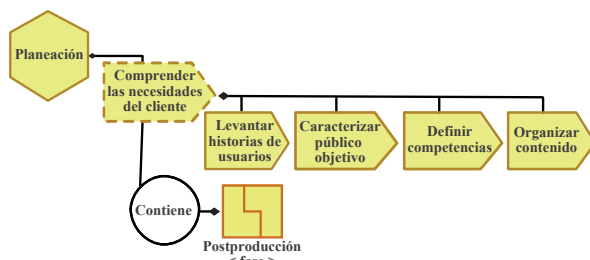
**Fig. 5.** Representación SEMAT de las fases del Modelo MADCE-TVD

La Figura 6 representa la fase Preproducción que contiene el espacio de actividad “Prepararse para hacer el trabajo” y las actividades: Definir el equipo, Identificar Herramientas y Reunión de Planeación. Estos elementos se asocian con la práctica “Planeación”.



**Fig. 6.** Representación en el núcleo de SEMAT de la fase de Preproducción y el espacio de actividad “Prepararse para hacer el trabajo”

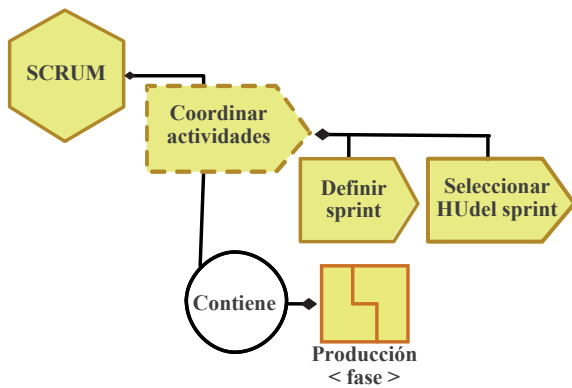
La Figura 7 representa la fase Preproducción que contiene el espacio de actividad “Comprender las necesidades del Cliente” y las actividades: Levantar Historias de Usuario (HU), Caracterizar público objetivo, Definir competencias y Organizar contenido. Estos elementos se asocian con la práctica “Planeación”.



**Fig. 7.** Representación en el núcleo de SEMAT de la fase de Preproducción y el espacio de actividad “Comprender las necesidades del cliente”.

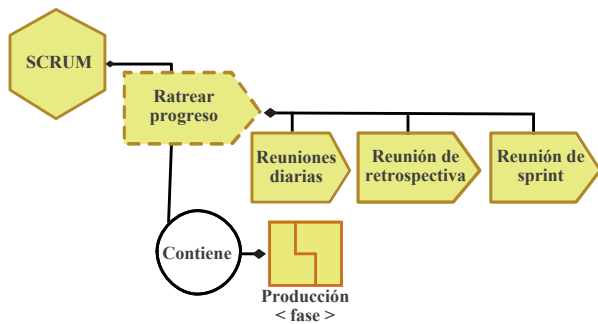
La Figura 8 representa la fase Producción que contiene el espacio de actividad “Coordinar Actividades” y las actividades: Definir sprint y Seleccionar HU de sprint. Estos elementos se asocian con la práctica “SCRUM”.





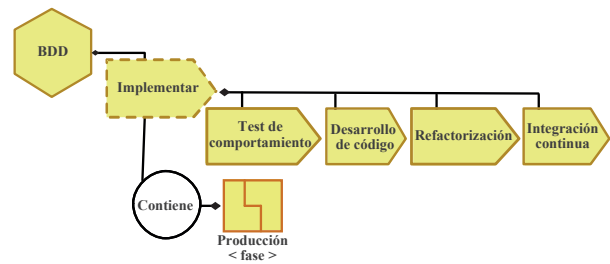
**Fig. 8.** Representación en el núcleo de SEMAT de la fase de Preproducción y el espacio de actividad “Prepararse para hacer el trabajo”

La Figura 9 representa la fase Producción que contiene el espacio de actividad “Rastrear Progreso” y las actividades: Reuniones diarias, Reunión de retrospectiva y Reunión de sprint. Estos elementos se asocian con la práctica “SCRUM”.



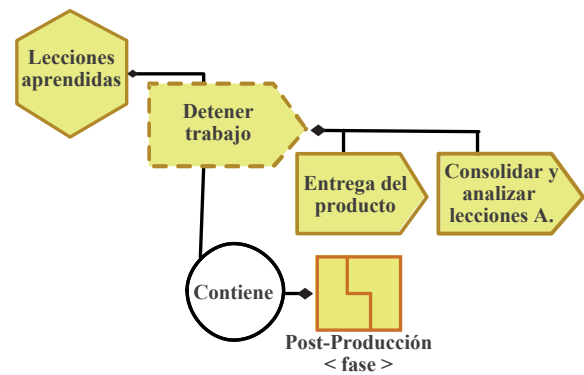
**Fig. 9.** Representación en el núcleo de SEMAT de la fase de Producción y el espacio de actividad “Rastrear progreso”.

La Figura 10 representa la fase Producción que contiene el espacio de actividad “Implementar” y las actividades: Test de comportamiento, Desarrollo de código, Refactorizar e Integración continua. Estos elementos se asocian con la práctica “BDD”.



**Fig. 10.** Representación en el núcleo de SEMAT de la fase de Producción y el espacio de actividad “Implementar”.

La Figura 11 representa la fase Postproducción, que contiene el espacio de actividad “Detener trabajo” y las actividades: Entrega del producto y Consolidar y analizar lecciones aprendidas. Estos elementos se asocian con la práctica “Lecciones Aprendidas”.



**Fig. 11.** Representación en el núcleo de SEMAT de la fase de Postproducción y el espacio de actividad “Detener trabajo”.

#### IV. CONCLUSIONES Y TRABAJOS FUTUROS

La representación con SEMAT facilita el entendimiento del flujo de datos de cada actividad que se ejecuta durante cada una de las etapas del modelo MADCE-TVD. Este modelo define unos lineamientos y buenas prácticas para aplicar al momento de desarrollar contenidos educativos para t-learning. El propósito del modelo es garantizar que los contenidos desarrollados sigan un proceso definido abarcando aspectos claves (técnicos, pedagógicos y de personalización) para este tipo de contenidos.

La representación gráfica del modelo a través de SEMAT permite evidenciar la trazabilidad y la consistencia en las prácticas que aborda el modelo, junto con los espacios de actividades que comprende y cada una de las actividades asociadas a ellos.

La representación en núcleo de SEMAT del Modelo MADCE-TVD permite a sus usuarios una mejor comprensión de los elementos que lo conforman. La representación permite identificar y seguir un orden de las fases para su implementación, y las actividades que se deben desarrollar para obtener los productos de trabajo asociados a cada actividad.

El trabajo futuro se relaciona con proponer nuevas representaciones SEMAT de los subprocesos, la representación de los productos de trabajo y los responsables de esos productos. Estas representaciones permitirán identificar las actividades y productos de trabajo del modelo MADCE-TVD.

## REFERENCIAS

- [1] I. Sommerville, Ingeniería de software, 9.<sup>a</sup> ed., Madrid: Pearson, 2011, pp. 1-79.
- [2] I. Jacobson, Pan-Wei Ng, P. McMahan, I. Spence y S. Lidman, “La Esencia de la Ingeniería de Software: El Núcleo de SEMAT”, Revista Latinoamericana de Ingeniería de Software, vol. 1, N.º 3, 2013, pp. 71-78.
- [3] A. Reyes, D. Soto y J. Jiménez, “MADCE-TVD-Model Agile Development Educational Content for Digital Television”, IEEE Latin America Transactions, vol. 13, N.º 10, pp. 3432-3438, 2015.
- [4] A. X. Reyes, y G. A. Moreno, “Aplicativo t-learning en la Televisión Digital Terrestre”, Repositorio Digital, 2015. Disponible en: <https://repositorial.cuaed.unam.mx:8443/xmlui/handle/123456789/3921>
- [5] C. Tomala, y S. Jazmin, “Métricas de calidad de sistemas de información: aplicación en certificación de la calidad de un sistema en una empresa del sector hidrocarburo”, Tesis de Licenciatura, 2016.
- [6] C. Durango, y C. Zapata, “Una representación basada en SEMAT y RUP para el Método de Desarrollo SIG del Instituto Geográfico Agustín Codazzi”, Ing. USBmed, vol. 6, N.º 1, pp. 24-37, 2015.
- [7] C. M. Zapata, R. E. Arango y L. D. Jiménez, “Mejoramiento de la consistencia entre la sintaxis textual y gráfica del lenguaje de Semat”, Polibits, N.º 49, pp. 83-89, 2014.
- [8] J. R. Laínez, Desarrollo de Software ÁGIL: Extreme Programming y Scrum, IT Campus Academy, 2015.
- [9] C. Solis y X. Wang, “A study of the characteristics of behavior driven development”, en Software Engineering and Advanced Applications (SEAA), 37th EUROMICRO Conference on. IEEE, 2011, pp. 383-387.
- [10] S. Nelson-Smith, Test-Driven Infrastructure with Chef, O’Reilly Media, Inc, 2013.
- [11] E. H. Uribe y L. E. V. Ayala, “Del manifiesto ágil, sus valores y principios. Scientia et technica, vol. 1, N.º 34, 2007.

## Capítulo VIII

# OBJETOS VIRTUALES DE APRENDIZAJE EN LA EDUCACIÓN SUPERIOR: UN CASO DE APLICACIÓN

*Victor Daniel Gil Vera - victor.gilve@amigo.edu.co*

Docente investigador, Universidad Católica Luis Amigó

*Lina María Montoya Suárez - lina.montoyasu@amigo.edu.co*

Docente investigadora, Universidad Católica Luis Amigó

*Ramiro Antonio. Giraldo Escobar - dec.ingenierias@amigo.edu.co*

Decano e investigador, Universidad Católica Luis Amigó

*Évelin Cristina Quiceno Sosa - evelin.quicenososa@amigo.edu.co*

Docente, Universidad Católica Luis Amigó

### I. INTRODUCCIÓN

Debido al ritmo acelerado y al desarrollo vertiginoso de las tecnologías de información y la comunicación (TIC) en la última década, sumado a los cambios de paradigmas relacionados con procesos de enseñanza y aprendizaje, hoy en día el modelo tradicional de educación basado en clases magistrales presenciales se ha visto obligado a incorporar el uso de la tecnología. En la actualidad, numerosas instituciones de educación superior a nivel mundial, incorporan el uso de sistemas expertos (SE), objetos virtuales de aprendizaje (OVA), plataformas educativas interactivas, cursos masivos abiertos y en línea (MOOC), etc., en el desarrollo de actividades académicas, tanto en cursos presenciales como a distancia.

Específicamente, los OVA tienen como objetivo principal emular el proceso de enseñanza y aprendizaje que se lleva a cabo en las aulas de clase y en los diferentes escenarios utilizados en la educación semipresencial, virtual y a distancia. Estos facilitan el acceso y la comunicación de los diferentes usuarios interesados en adquirir nuevo conocimiento [2]. Están diseñados para guiar el proceso de aprendizaje de cualquier persona interesada en acceder a un curso en cualquier momento y lugar, sin necesidad de

interactuar directamente con un experto o maestro humano [3]. Contienen recursos digitales, auto contenibles y reutilizables, actividades de aprendizaje y elementos de contextualización [4].

El objetivo de este trabajo es presentar una herramienta académica, analizar y diseñar un modelo de OVA para el manejo de hojas de cálculo en el programa MS Excel. Para la construcción del módulo de conocimiento (BC), se identificaron las principales temáticas dictadas en estos cursos, se emplearon guías y material didáctico especializado, posteriormente se construyó el módulo del estudiante, el módulo del docente y la interfaz, medio a través del cual se presenta el OVA al usuario final. En la construcción, se empleó el software libre de código fuente abierto XOT, desarrollado por la Universidad de Nottingham. El trabajo se divide de la siguiente manera, primero se presenta una contextualización general sobre los OVA y algunos casos de aplicación en la educación superior. Posteriormente se presenta la metodología utilizada y los resultados del caso de aplicación.

Se concluye, los estudiantes pueden presentar un mejoramiento significativo en su proceso de aprendizaje utilizando OVA, en este caso les facilitó la comprensión de las diferentes funcionalidades de MS

Excel. La mayoría de estudiantes se ven más motivados por aprender cuando disponen de herramientas móviles con contenido interactivo. En la revisión de la literatura se encontró que los OVA son de gran utilidad en el proceso de enseñanza. Investigaciones previas han demostrado que los estudiantes aprenden con OVA [16]. El uso de OVA conduce a importantes y grandes logros de aprendizaje [17].

## II. OBJETOS VIRTUALES DE APRENDIZAJE

Los objetos virtuales de aprendizaje (OVA) fueron utilizados por primera vez en el año 1970 como una manera de proporcionar mayor flexibilidad a la estrategia de aprendizaje y lograr mejor interacción con el usuario [6]. Los OVA han proporcionado un terreno fértil para la investigación en IA durante los últimos veinticinco años [7]. El objetivo principal de los OVA es capturar el conocimiento de los expertos para crear interacciones dinámicas con los usuarios, que les permita identificar fortalezas y debilidades sin tener que interactuar directamente con un experto humano [6]. Los OVA son sistemas sofisticados de software que pueden proporcionar instrucción personalizada a los estudiantes. Muchos de estos sistemas han demostrado ser eficaces en áreas como el álgebra, la física, etc. [8], fomentan y evalúan el aprendizaje a través de adaptaciones interactivas entre el estudiante y el sistema, y las instrucciones contienen el dominio específico del conocimiento pedagógico y el conocimiento del estudiante [9], [10].

Entre las diversas definiciones de OVA encontradas en la revisión del estado del arte, algunos autores los definen como ambientes interactivos de aprendizaje que apoyan la adquisición de habilidades cognitivas [5], los cuales pueden ser utilizados en diferentes plataformas y herramientas mediadas por las tecnologías de la información y la comunicación (TIC), y permiten al usuario adquirir nuevo conocimiento en cualquier área del saber [3].

Para otros autores se trata de entidades digitales o no digitales que pueden ser utilizadas para el aprendizaje y la capacitación en la educación [7]. Son herramientas que contienen material educativo, donde la temática y el contenido son claramente identificables y direccionables. Una de sus principales ventajas es la

posibilidad de reutilizarlos en distintos contextos de la educación virtual y a distancia [8]. Los OVA permiten obtener un enfoque asociado a la construcción de diferentes cursos para: facilitar el aprendizaje, tener adaptabilidad dinámica independientemente del lugar y el espacio, dinamizar los procesos de investigación, alinear el desarrollo de competencias, ahorrar tiempo, facilitar el aprendizaje autónomo, tener acceso simultáneo en tiempo real y promover el trabajo colaborativo [3].

En resumen, los OVA tienen como objetivo principal emular a un docente humano tanto en su saber como en sus habilidades pedagógicas y comunicativas [6] different dynamic, algorithmic and heuristic processes are encouraged, being useful for logic and constructivist student training. The proposed methodology is based on the construction of VLOs (Virtual Learning Objects, y cuentan con fichas de registro o metadatos que consisten en un listado de atributos, que además de describir el uso posible del objeto, permiten su catalogación y su intercambio [2].

Los OVA están basados en modelos de instrucción de contenido que especifican qué y cómo enseñar [2]. Tienen la capacidad de hacer inferencias sobre el dominio de temáticas con el fin de adaptar dinámicamente su contenido, y permitir a los estudiantes aprender haciendo, en contextos reales y significativos, y tener más control sobre su aprendizaje [7]. La principal ventaja de los OVA, en comparación con los sistemas tutoriales tradicionales, radica en su flexibilidad, tanto en el enfoque como en la adaptación del alumno. Los sistemas tutoriales tradicionales contienen una gran cantidad de reglas e información, lo que puede generar confusión [7].

El diseño de los OVA se basa en la hipótesis fundamental de que los estudiantes aprenden mejor haciendo, cometiendo errores y fortaleciendo sus conocimientos [11]. En resumen, los OVA tienen el potencial de introducir avances en el campo de la educación y el aprendizaje [8] y permiten que el aprendizaje pueda realizarse en cualquier momento y lugar [12]. Los OVA ofrecen grandes beneficios, pero son una clase de tecnología tradicionalmente diseñada para países desarrollados [13]. Esto se debe principalmente al desarrollo vertiginoso que ha tenido

la educación a distancia (B-learning), especialmente en países europeos y EE. UU. con un alto índice de desarrollo humano (IDH), en la cual se le da la posibilidad al estudiante de aprender en el momento y lugar que desee, transformando por completo el modelo tradicional de enseñanza-aprendizaje. Sin embargo, se debe añadir que una gran cantidad de universidades en países latinoamericanos han realizado avances considerables en la virtualización de la educación. Los OVA se basan en información detallada con dominio de conocimiento que es difícil de obtener y difícil de codificar [14]. Sus principales componentes son:

**-Módulo de conocimiento:** También conocido como base de conocimiento (BC). Consiste en una representación de las temáticas que se enseñarán en el OVA [15].

**-Módulo del estudiante:** Almacena información acerca del rendimiento del estudiante o su comportamiento, incluyendo su nivel de conocimientos, sus preferencias y su historial de aprendizaje. Ayuda a personalizar la estrategia docente [15].

**-Módulo del docente:** Contiene una representación de las estrategias pedagógicas de los docentes y proporciona instrucciones adaptables a los estudiantes [15].

**-Interfaz:** Proporciona los medios para que el alumno interactúe con el OVA [15].

### III. USO DE OVA EN LA EDUCACIÓN SUPERIOR

La incorporación de las TIC en la vida cotidiana está impactando los estilos de vida de las personas en el mundo, permeando de manera significativa la educación superior, en especial los sistemas de gestión del aprendizaje (LMS), y permitiendo romper paradigmas y cambios académicos [2]. En la actualidad, dispositivos móviles (celulares, tabletas, Ipad, etc.) que tengan alguna forma de conectividad inalámbrica, se constituyen como una de las principales herramientas potenciales para introducir los OVA en el proceso de enseñanza-aprendizaje, solo

basta con acceder a un enlace en la web o descargar el OVA al dispositivo móvil para poder tener acceso al conocimiento en cualquier momento y lugar (teoría, actividades prácticas, trabajos, etc.) [9]. Sin embargo, la inclusión de los OVA en la educación superior sigue siendo un gran desafío en universidades latinoamericanas [10].

Las TIC han incorporado la necesidad de generar materiales didácticos multimedia [9], [11], los cuales tienen como objetivo dinamizar y hacer más didáctico el proceso enseñanza-aprendizaje [10], [11] y facilitar el acceso y la conectividad a contenidos [12], [13]. Es importante considerar que el uso de las TIC facilita el aprendizaje autónomo; “en la enseñanza tradicional el docente ha estado destinado a ser el único responsable del aprendizaje de los estudiantes”, en la actualidad se gira en torno a los modelos colaborativos como mecanismo para mejorar el proceso de enseñanza y aprendizaje dentro de las salas de clase, y los OVA cumplen un papel fundamental para dicho proceso [14], [15].

#### A. Casos de Aplicación en la Educación Superior

En algunas disciplinas el uso de OVA se encuentra todavía en su fase preliminar [16]. La mayoría de las disciplinas que utilizan OVA son ciencias exactas y naturales, ciencias de la computación y ciencias médicas, como la enfermería [16]. El alto porcentaje de profesores que carecen de conocimientos acerca de los OVA y las habilidades técnicas para trabajar con ellos, refleja la deliberada y lenta actitud de muchas instituciones para prepararse al cambio en el proceso de enseñanza y aprendizaje [16].

El Ministerio de Educación Nacional de Colombia (MEN) ejecutó una prueba piloto de Objetos de Aprendizaje para las instituciones de educación superior (IES), para consolidar Bancos de OVA que tuvieran producción de material educativo digital de categorías como: cursos, tutoriales, enseñanza digital, etc., con el objetivo de analizar las experiencias [17]. En la actualidad, la mayoría de las instituciones carecen de sus propios repositorios para almacenar OVA o no están familiarizadas con los que están disponibles, lo que hace más lentos su incorporación y uso [13].



#### IV. METODOLOGÍA

Esta sección presenta el método de búsqueda utilizado para identificar trabajos relacionados con aplicaciones para dispositivos móviles sobre el manejo de hojas de cálculo en MS EXCEL. Se desarrolló una búsqueda de revisiones de literatura, artículos, artículos en desarrollo, libros, capítulos de libros y memorias de conferencias en bases de datos científicas. Adicionalmente, se revisaron artículos publicados en las siguientes revistas: *Journal of Computer Assisted Learning (JCAL)*, *Computers and Education*, *International Journal of Computer-Supported, Collaborative Learning and International Journal of Learning Technology*.

El rango de búsqueda de documentos comprendió desde el año 2010 hasta el año 2018. En total, se revisaron 24 memorias de conferencias sobre M-learning y 10 artículos de revistas especializadas. Las palabras claves de búsqueda utilizadas fueron: educación, OVA, MS Excel, Aprendizaje móvil, Aplicaciones. En total se revisaron 34 publicaciones. Las ecuaciones de búsqueda utilizadas fueron:

(1) (TITLE-ABS-KEY (Aprendizaje móvil) OR (Educación) AND (((MS Excel) OR ((OVA) OR ((Aprendizaje) OR (Aplicaciones)))))) >2009

(2) (TITLE-ABS-KEY (Aprendizaje móvil) AND (Educación) AND (((MS Excel) AND ((OVA) AND ((Aprendizaje) AND (Aplicaciones)))))) >2009

(3) (TITLE-ABS-KEY (Aprendizaje móvil) AND (Educación) AND (((MS Excel) OR ((OVA) AND ((Aprendizaje) OR (Aplicaciones)))))) >2009

(4) (TITLE-ABS-KEY (Aprendizaje móvil) AND (Educación) AND (((MS Excel) AND ((OVA) AND ((Aprendizaje) OR (Aplicaciones)))))) >2009

(5) (TITLE-ABS-KEY (Aprendizaje móvil) OR (Educación) OR (((MS Excel) OR ((OVA) OR ((Aprendizaje) AND (Aplicaciones)))))) >2009

De la búsqueda se identificaron aplicaciones para dispositivos móviles relacionadas con la enseñanza de MS Excel en Play Store, desarrollado para dispositivos móviles con sistema operativo Android (ver Tabla 1).

**Tabla 1.** Aplicaciones Móviles

Nombre	Desarrollador	Año
Formation Excel	NajmCV	2017
Learn Excel 2016	MS DeOVAper	2017
Learn Excel Offline	ASK2GURUJI	2017
Learn Excel Full	Mob-Educacións	2017
Learn MS Excel Course & Keys	RK Technolabs	2017
MS Excel Full Course	International Edu-cacional Apps	2017

De los resultados de la búsqueda no se identificaron trabajos relacionados con la implementación de aplicaciones para dispositivos móviles relacionadas con la enseñanza de hojas de cálculo. Adicionalmente, se buscaron aplicaciones en App Store, y no se encontraron resultados.

#### A) Proceso de Construcción del OVA

La metodología utilizada para la construcción de la base de conocimientos (BC) de “Cyber Excel”, fue la revisión sistemática de literatura [19], la cual permite identificar, evaluar e interpretar toda la investigación relevante disponible relacionada con una pregunta de investigación, temática o fenómeno de interés. Las investigaciones y estudios previos sirven como base para la realización de la RSL, los cuales son llamados estudios primarios. La RSL por sí sola es considerada como un estudio secundario [18]. Según [19], los pasos para realizar una RSL son los siguientes:

- **Planeación de la revisión:** La salida de esta fase es un protocolo de revisión que define el propósito y los procedimientos para la revisión [19].

- **Desarrollo de la revisión:** En esta fase se generan los resultados definitivos, la lista de publicaciones seleccionadas y el registro de evaluaciones de calidad, y se extraen los datos de cada una de las publicaciones seleccionadas [19].

- **Reporte de la revisión:** Generación de un reporte técnico en una revista o conferencia científica [19].

La pregunta de investigación considerada en este trabajo fue P1. ¿Cuáles son los principales tópicos y teorías sobre el uso de MS EXCEL que los estudiantes universitarios deben aprender? Las palabras clave utilizadas fueron: VLO, MS Excel, M-learning. La ecuación de búsqueda utilizada fue:



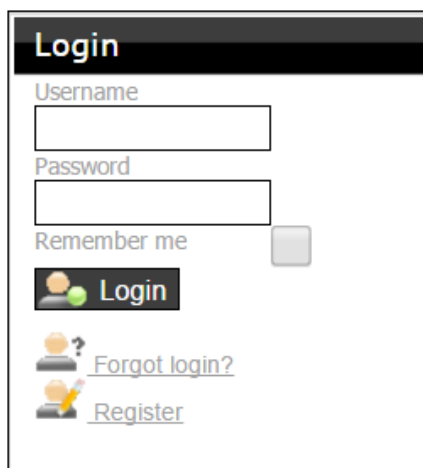
(6) (TITLE-ABS-KEY (VLO)) AND ((MS Excel) AND (M-learning) AND (LIMIT-TO (DOCTYPE, "cp")) OR LIMIT-TO (DOCTYPE, "ar")) OR LIMIT-TO (DOCTYPE, "re")) AND (LIMIT-TO (SUBJAREA, "COMP") OR LIMIT-TO (SUBJAREA, "ENGI") OR LIMIT-TO (SUBJAREA, "Exc"))

El rango de búsqueda estuvo comprendido desde el año 2010 hasta el año 2018. En total fueron seleccionadas diez publicaciones entre artículos, artículos de conferencias y libros (ver Tabla 2).

**Tabla 2.** Tipos de Documentos

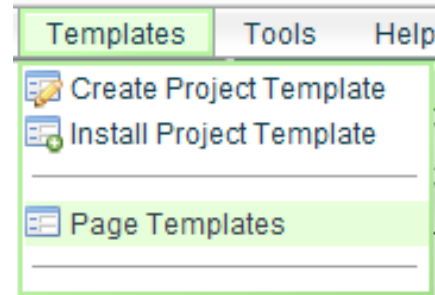
Tipo de documento	Cantidad	Referencias
Artículos	5	[19], [20], [21], [22] y [23]
Artículos de conferencias	4	[24], [25], [26] y [27]
Libros	1	[28]
Total	10	

Adicionalmente, se realizó una búsqueda de literatura gris, se recolectaron nueve manuales publicados por universidades colombianas, los cuales se utilizaron para la construcción de la base de conocimiento (BC). "Cyber Excel" fue desarrollado con el software XOT, el cual proporciona un completo conjunto de herramientas de código abierto para desarrolladores de e-learning y creadores de materiales didácticos interactivos. Se puede descargar de manera gratuita en la siguiente dirección: <http://www.xerte.org.uk/index.php?lang=es>. El usuario debe registrarse y descargar el instalador (ver Figura 1).



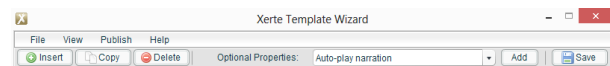
**Fig. 1.** Acceso. Fuente: XOT

Una vez descargado e instalado el programa, el usuario debe hacer clic en la opción "Plantillas de página" (ver Figura 2).



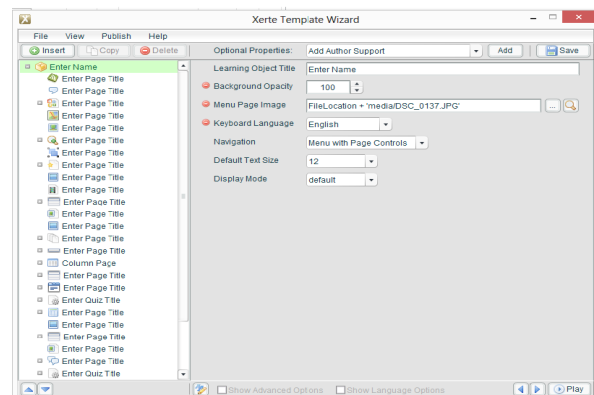
**Fig. 2.** Plantillas. Fuente: XOT

El programa muestra en la barra de menú las opciones: Archivo (proyecto nuevo, abrir, guardar, salir), Vista (vista previa), Publicar (paquete) y Ayuda. La barra de herramientas contiene tres botones: Insertar (texto, multimedia, interactividad, juegos, conectores, navegantes y Misc), Copiar, Eliminar; luego: Propiedades opcionales y Guardar (ver Figura 3).



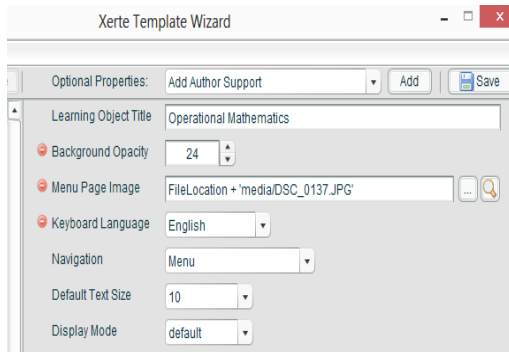
**Fig. 3.** Menú y barra de herramientas. Fuente: XOT

Todos los recursos que se añaden al OVA (actividades, recursos multimedia, imágenes, sonidos, etc.) se almacenan en la parte izquierda (ver Figura 4).



**Fig. 4.** Recursos. Fuente: XOT

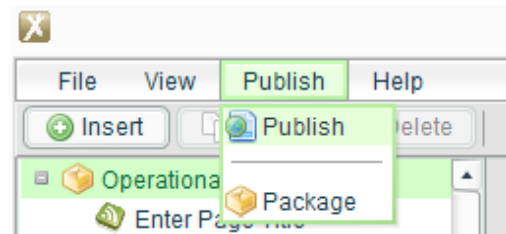
El usuario debe configurar los siguientes elementos: imagen de fondo, idioma (inglés, francés, alemán, italiano, croata), título, forma de navegación (lineal, menú, menú con controles de página), tamaño del texto (10 - 36) y modo de visualización. Para guardar los cambios se debe dar clic en Guardar (ver Figura 5).



**Fig. 5.** Configuración. Fuente: XOT

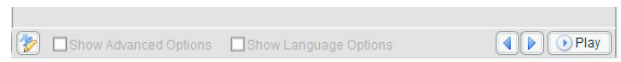
“Cyber - Excel” funciona para dispositivos móviles que cuenten con sistema operativo Android e iOS, y que tengan conexión a internet. Este permite al estudiante interactuar en cualquier momento y lugar sin necesidad de estar presencialmente en un aula de clase, lo que le facilita la construcción del conocimiento, la

Para publicar el OVA se debe dar clic en Publicar (ver Figura 6).



**Fig. 6.** Publicación. Fuente: XOT

Para utilizar el OVA se debe presionar el botón Play ubicado en la parte inferior derecha (ver Figura 7).



**Fig. 7.** Uso. Fuente: XOT

resolución de problemas y el desarrollo de habilidades y destrezas de manera autónoma. También puede ser utilizado en ordenadores de escritorio. El dispositivo móvil debe tener instalada una versión mayor o igual 11.1 de Adobe Flash Player. La Figura 8 presenta la interfaz inicial.

Cyber Excel (OVA)

## Objeto Virtual de Aprendizaje de Excel

# Cyber Excel<sup>®</sup>



**Fig. 8.** Interfaz del OVA.

La Figura 9 presenta la segunda interfaz del OVA; objetivos, audiencia, prerequisites y manera de utilizarlo:

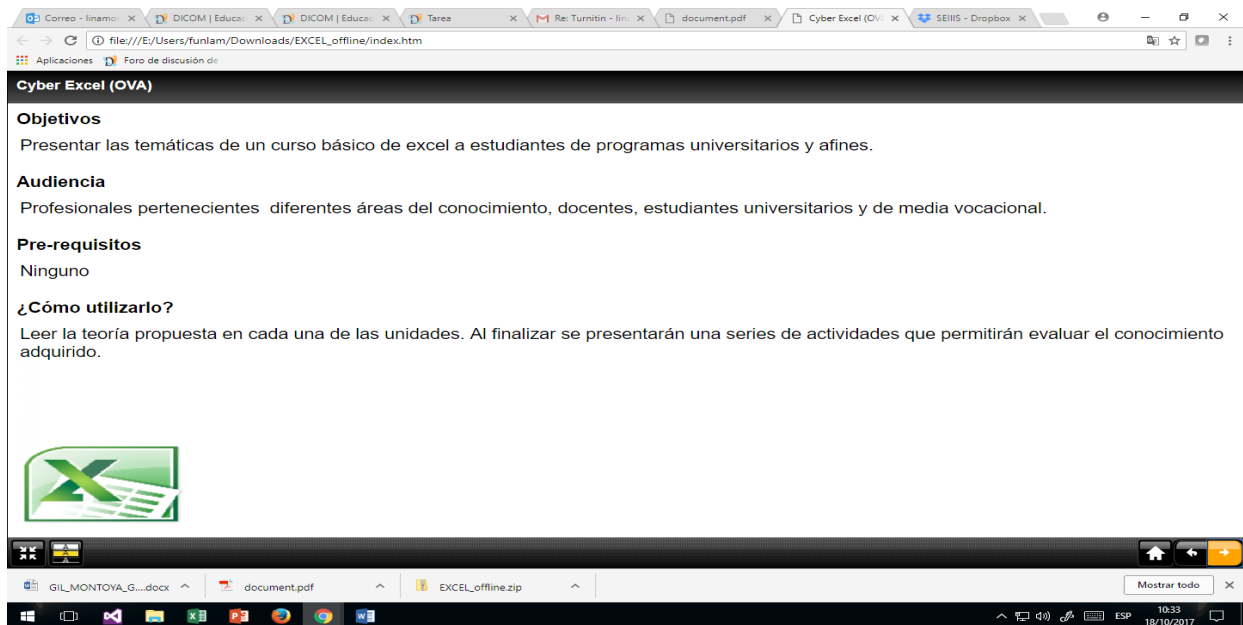


Fig. 9. Interfaz del OVA.

La Figura 10 presenta el contenido temático del OVA (conceptos básicos, principios de modelado, sistemas de primer y segundo orden, variables de estado y una breve introducción a la teoría de control). El estudiante

tiene la opción de seleccionar la unidad que desee conocer o bien desarrollar todas las unidades en orden secuencial. Todas estas temáticas constituyen el módulo de conocimiento o base de conocimiento del OVA.

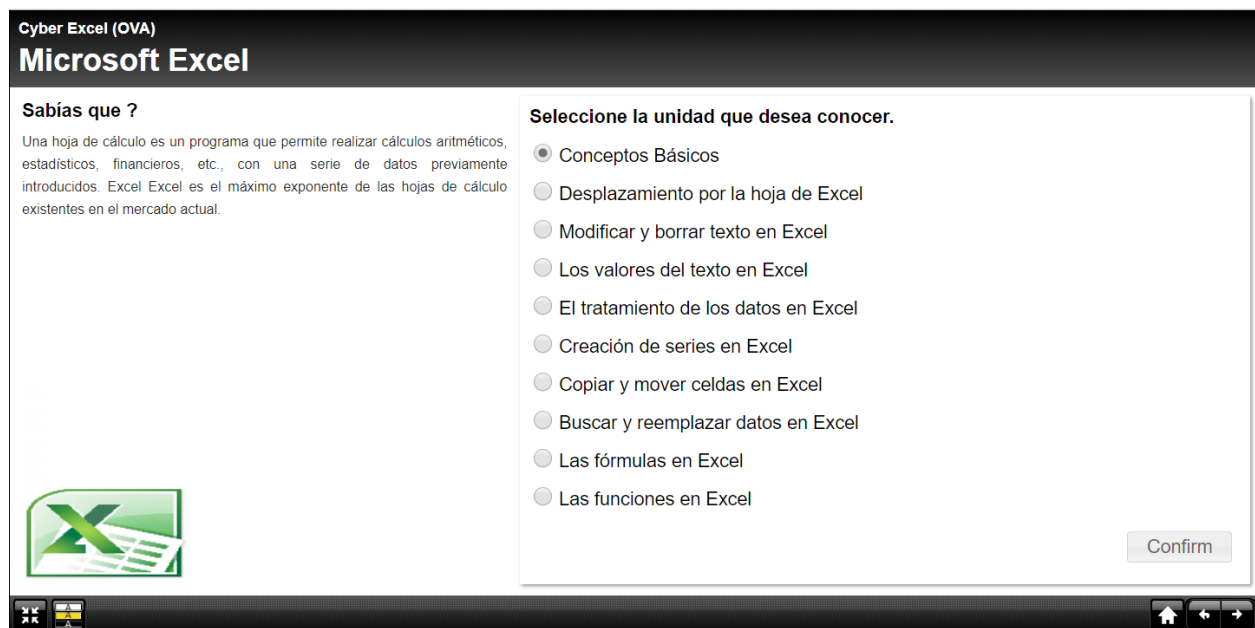
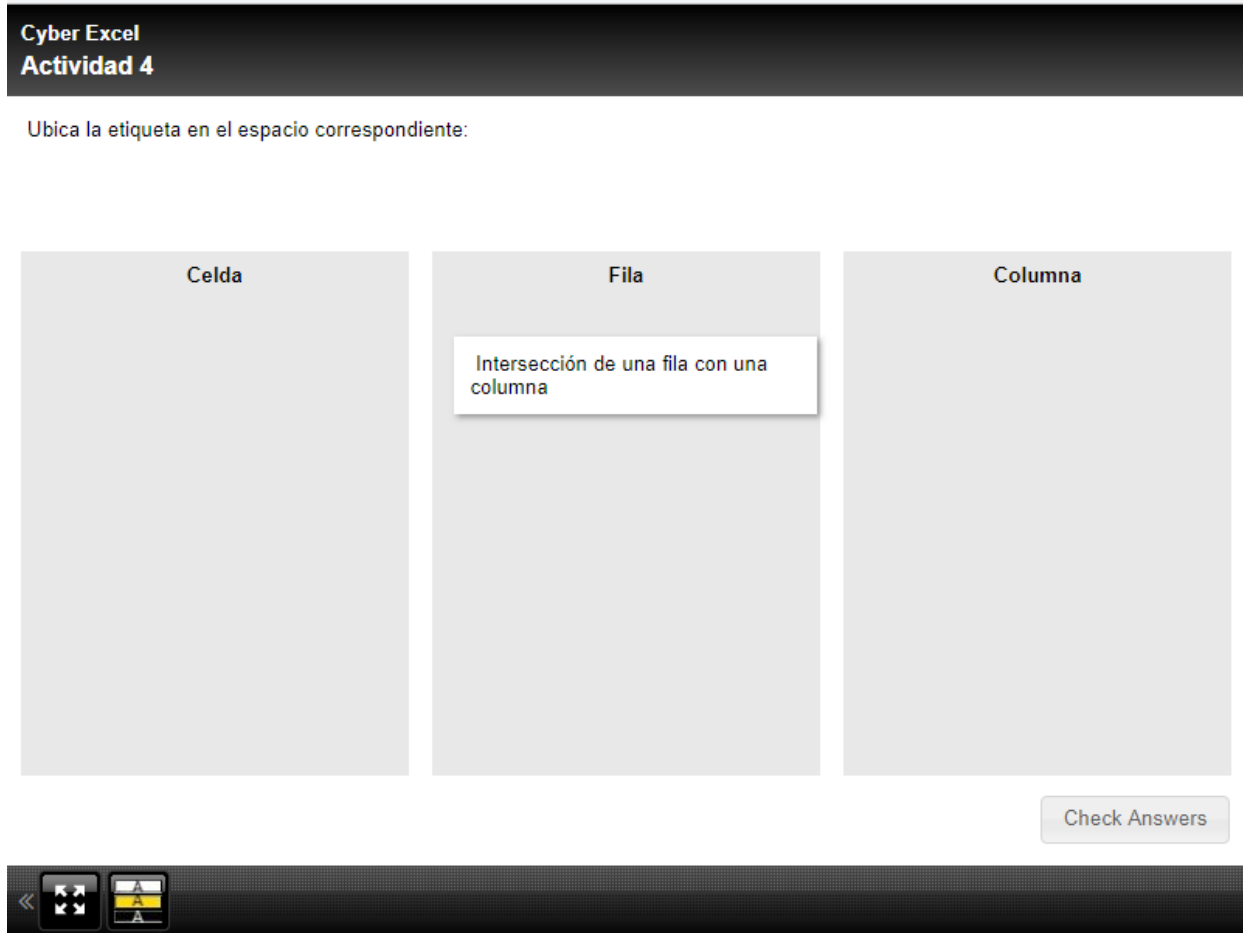


Fig. 10. Interfaz del OVA.

El módulo del estudiante del OVA va almacenando toda la información de las actividades que éste desarrolla y su historial de aprendizaje. Esta información se convierte en el insumo principal

del módulo del docente, ya que le permite definir estrategias personalizadas para cada estudiante según su desempeño. La Figura 11 presenta la actividad 4 del OVA.



**Fig. 11.** Interfaz del OVA.

Una vez culminadas todas las actividades, el componente de autoevaluación presenta al estudiante un examen de diez preguntas de selección múltiple, que le permitirán verificar la consolidación de los

conocimientos adquiridos. Al finalizar se presenta la puntuación obtenida. Se aprueba la unidad si el estudiante aprueba 6 de las 10 preguntas del examen (ver Figura 12 y Figura 13).

**Cyber Excel**  
**Prueba de conocimientos**

Seleccione una opción

**Pregunta 1 de 1**  
¿Qué es una hoja de cálculo?

- Son hojas tabulares que se usan como libros de contabilidad y este constituido columnas y filas que permiten registrar datos y fórmulas en cada celda
- Libro electrónico de trabajo que nos permite realizar cálculos mediante fórmulas predefinidas o fórmulas creadas por el usuario.
- Celda que esta seleccionada.
- Sirve para representar datos e información de manera visual




Fig. 12. Pregunta 1 - Examen.

**SD Objeto de aprendizaje**  
**Examen**

Seleccione la respuesta correcta

has completado el ejercicio

Bien  
Has puntuado 6/10





Fig. 13. Puntuación del OVA

## V. CONCLUSIONES

Este trabajo presenta el desarrollo del OVA “MS Excel - OVA”, orientado a la enseñanza y aprendizaje de la MS EXCEL. Los OVA facilitan el proceso de enseñanza aprendizaje en cualquier área de conocimiento, ya que estos tienen la habilidad de generar retroalimentaciones e instrucciones individuales a un usuario en tiempo real, lo que permite transformar el modelo tradicional de enseñanza aprendizaje. Por lo anterior, la enseñanza de la estadística puede dejar de convertirse en una ardua y difícil tarea, gracias al m-learning y a la enseñanza personalizada.

De la revisión del estado de arte se identifica que los OVA son mucho más utilizados en ciencias exactas y naturales que en ciencias sociales y humanas. Las ciencias exactas históricamente han permitido establecer una mayor estandarización de contenidos curriculares y pedagógicos, lo que permite compartir las “mejores prácticas de enseñanza - aprendizaje”.

## REFERENCIAS

- [1] Colombia Aprende, “Primer Concurso de Nacional de Objetos de Aprendizaje, 2005”, 2005. Disponible en: <http://www.colombiaprende.edu.co/html/directivos/1598/article-99368.html>
- [2] Kalhor, Q et al., “M-learning -an innovative advancement of ict in education”, en 2010 4th International Conference on Distance Learning and Education (ICDLE), pp. 148-151, 2010.
- [3] Y. S. Pascuas, C. O. Jaramillo y F. A. Verástegui, “Desarrollo de objetos Virtuales de Aprendizaje como estrategia para fomentar la permanencia estudiantil en la educación superior”, Revista Escuela de Administración de Negocios, N.º 79, pp. 116-129, jul.-dic. 2015.
- [4] Ministerio de Educación Nacional, “Objetos Virtuales de Aprendizaje e Informativos”, Portal Colombia Aprende, 2006.
- [5] T. Goh et al., “Getting ready for mobile learning-adaptation perspective”, J. Educ. Multimed. Hypermedia, vol. 15, N.º 2, p. 175, 2006.
- [6] J. Arango, D. Gaviria y A. Valencia, “Differential Calculus Teaching through Virtual Learning Objects in the Field of Management Sciences”, Procedia - Soc. Behav. Sci., vol. 176, pp. 412-418, 2015.
- [7] IEEE, Learning Object Metadata IEEE, 2017.
- [8] G. R. Ruiz, A. J. Muñoz y R. F. Álvarez, “La Calidad en los Objetos de Aprendizaje”, Av. en la Cienc. la Comput. [Internet], pp. 352-357, 2006.
- [9] C. Arturo, T. Gastelú, and L. Domínguez, “ITC integration process in basic education: The case of Veracruz city in Mexico”, Procedia - Social and Behavioral Sciences, vol. 176, pp. 819-823, feb. 2015.
- [10] M. C. L. de la Madrid y J. A. C. Espinoza, “La formación de profesores universitarios en la aplicación de las TIC”, Sinéctica, N.º 41, 2013.
- [11] U. Miranda, P. Bacelar-nicolau, F. J. P. Caetano, y S. Caeiro, “Education for sustainable development through e-learning in higher education: experiences from Portugal Education for Sustainable Development”, J. Clean. Prod., vol. 106, pp. 308-319, 2015.
- [12] J. C. Almenara, “Las TIC como elementos para la flexibilización de los espacios educativos: retos y preocupaciones”, Comun. y Pedagog. Rev. Nuevas Tecnol. y Recur. Didácticos, vol. 194, pp. 13-19, 2004.
- [13] J. M. Duart y F. Lupiáñez, “E-strategias en la introducción y uso de las TIC en la universidad”, DIM Didáctica, Innovación y Multimed., vol. 1, N.º 11, pp. 1-20, 2008.
- [14] R. Moreno y R. J. Martínez, “Aprendizaje autónomo. Desarrollo de una definición”, Acta Comport. Rev. Lat. Análisis del Comport., vol. 15, N.º 1, 2007.
- [15] B. Rubia, I. Jorri y R. Anguita, “Aprendizaje colaborativo y TIC”, en Tecnología Educativa. La Formación del Profesorado en la era Internet, J. DePablos, Coord., Málaga: Aljibe, 2009, pp. 191-214.
- [16] R. Griffith, “Learning objects in higher education”, Acad. ADL Co-Lab, vol. 1, pp. 1-18, 2003.
- [17] I. I. S. Medina, “Estado del arte de las metodologías y modelos de los Objetos Virtuales de Aprendizaje (OVAS) en Colombia”, Entornos, N.º 28, pp. 93-107, 2014.
- [18] K. Petersen, R. Feldt, S. Mujtaba y M. Mattsson, “Systematic Mapping Studies in Software Engineering Systematic”, en Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (Swinton, UK, 2008), EASE’08, British Computer Society, pp. 68-77.



- [19] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey y S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review", *Inf. Softw. Technol.*, vol. 51, N.º 1, pp. 7-15, enero 2009.
- [20] S. Alessi, "Designing educacional support in system-dynamics-based interactive learning environments", *Simul. Gaming*, vol. 31, N.º 2, pp. 178-196, 2000.
- [21] J. D. Sterman, "MS Excel modeling: tools for learning in a complex world", *Calif. Manage. Rev.*, vol. 43, N.º 4, pp. 8-25, 2001.
- [22] J. W. Forrester, "Learning through MS Excel as preparation for the 21st century", en *Keynote Address for Systems Thinking and Dynamic Modeling Conference for K-12 Education*, 1994, pp. 27-29.
- [23] J. A. M. Vennix, *Group model building: Facilitating team learning using system dynamics*, Chichester, UK: Wiley, 1996.
- [24] A. Borshchev y A. Filippov, "From MS Excel and discrete event to practical agent based modeling: reasons, techniques, tools", en *Proceedings of the 22nd international conference of the MS Excel society*, 2004.
- [25] J. B. Homer y G. B. Hirsch, "MS Excel modeling for public health: background and opportunities", *Am. J. Public Health*, vol. 96, N.º 3, pp. 452-458, 2006.
- [26] J. W. Forrester, "MS Excel, systems thinking, and soft OR", *Syst. Dyn. Rev.*, vol. 10, No. 2-3, pp. 245-256, 1994.
- [27] I. Winz, G. Brierley y S. Trowsdale, "The use of MS Excel simulation in water resources management", *Water Resour. Manag.*, vol. 23, No. 7, pp. 1301-1323, 2009.
- [28] K. R. Padiyar, *Power MS Excel*. BS Publ., 2008.



## Capítulo IX

# LA MODA DE CLOUD COMPUTING: DESAFÍOS CURRICULARES

*Milton J. Vera Contreras - miljeveco@gmail.com*

Docente e investigador, Universidad Francisco de Paula Santander, Cúcuta, Colombia

*Matías Herrera Cáceres - matiashc@ufps.edu.co*

Docente e investigador, Universidad Francisco de Paula Santander, Cúcuta, Colombia

*Óscar Alberto Gallardo Pérez - oagallardo@gmail.com ; oagallardo@ufps.edu.co*

Docente, Universidad Francisco de Paula Santander, Cúcuta, Colombia

### I. INTRODUCCIÓN

Cada vez que sale al mercado una nueva tecnología y comienza a popularizarse, los estudiantes se inquietan porque la universidad no les ofrece dicha tecnología. La respuesta más común de la universidad es abrir una asignatura electiva, que desaparece cuando la tecnología deja de ser importante en el mercado o se convierte en asignatura obligatoria cuando el mercado estandariza y mantiene dicha tecnología por varios años. Otra respuesta de la universidad es ignorar la inquietud de los estudiantes. En este caso, hay ocasiones en que se acierta y la tecnología sale del mercado rápidamente o tarda mucho tiempo en establecerse. Pero también hay ocasiones en que la Universidad se equivoca y, tardíamente, se ve obligada a introducir en su currículo la nueva tecnología, que ya puede ser vieja pero aún necesaria o incluso indispensable.

Este comportamiento es muy similar a lo que sucede en las organizaciones, por lo cual se puede estudiar desde la perspectiva de la Teoría de la Moda de Abrahamson [1–4]. No obstante, además del problema teórico, para la Universidad es importante abordar el problema práctico: ¿Adoptar o liderar modas? ¿Adoptarlas o ignorarlas? ¿Cuándo adoptarlas? ¿Cómo adoptarlas? En este documento se aborda, con fundamento en la Teoría de la Moda de Abrahamson, el problema práctico, específicamente el ¿cómo? para el caso concreto de Cloud Computing, que es uno de

los términos de moda en la actualidad, junto a Big Data, Internet of Things (IoT), entre otros.

El documento presenta primero una breve revisión de la Teoría de la Moda, argumentando que el mercado origina modas y algunas organizaciones (lo que incluye a las universidades) las adoptan. De acuerdo a la teoría de la moda, se plantea que la decisión de adoptar o no debe tomarse razonadamente, no por presiones del mercado ni por presiones internas a la organización, sino con criterios objetivos. Luego se presenta una revisión detallada de la literatura sobre Cloud Computing, para delimitar el alcance de la moda que se quiere adoptar. En tercer lugar, se proponen cuatro maneras diferentes para incorporar Cloud Computing en el currículo: (i) la perspectiva pragmática, como una herramienta, uso de TIC; (ii) la perspectiva administrativa, como un modelo de gestión, gestión de TIC/SI; (iii) la perspectiva económica, como un modelo de negocio, y (iv) la perspectiva científica, como un paradigma. Finalmente se plantean conclusiones y trabajos futuros respecto a la propuesta.

### II. REVISIÓN DE LITERATURA

#### A. Teoría de la Moda

La Teoría de la Moda aparece como una línea de pensamiento derivada de la Teoría de la Difusión de la Innovación (TDI) de Roger, que estudia la manera

como las personas o las organizaciones evalúan, rechazan o adoptan e implementan una innovación [5]. Se reconoce como autor principal de la Teoría de la Moda a Abrahamson [6–9], quien formuló la teoría en los años noventa, pero desde 1969 ya se había formulado la pregunta ¿existe moda en la ciencia? [10].

En esta teoría se plantea que una moda es la creencia temporal sobre los beneficios de una innovación, bien sea por su implementación en la práctica o por considerarla como objeto de estudio, en la docencia o la investigación. Un postulado importante en esta teoría es que, tanto en la práctica como en la academia, se difunden o rechazan modas sin criterios objetivos, en un proceso regulado por poderes y tendencias donde están involucrados dos tipos de actores: los promotores de la moda —*fashion setter*— y los adoptantes o usuarios de la moda —*fashion users-adopters*— (Figura 1) [6–9].

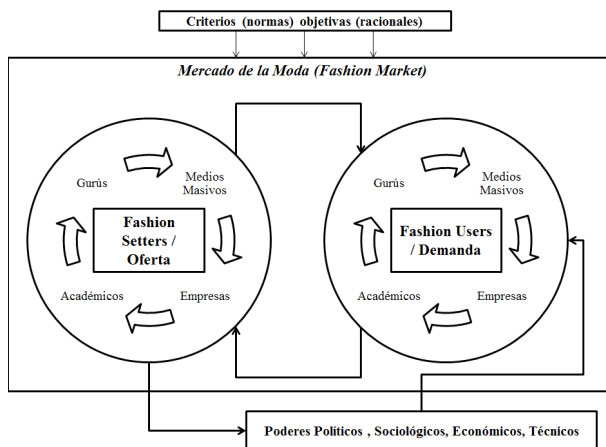


Fig. 1. Moda. Fuente: Adaptado de [2].

Abrahamson formula dos preguntas: (i) ¿cuándo y por qué innovaciones ineficientes se adoptan o difunden? y (ii) ¿cuándo y por qué innovaciones eficientes se rechazan? [6]. Estas dos preguntas generan cuatro posibilidades al difundir o rechazar innovaciones: (i) El mejor caso se presenta cuando se adopta o difunde una innovación eficiente. (ii) El peor caso es lo contrario, cuando se rechaza una innovación eficiente. (iii) La tercera posibilidad es considerada como un caso muy malo o inconveniente, adoptar o difundir una innovación ineficiente. (iv) La

última posibilidad se considera apropiada, rechazar una innovación que no aportará ningún beneficio. Abrahamson se enfoca en la segunda y la tercera posibilidades (Figura 2), que corresponden a sus dos preguntas, e indaga cuándo y por qué se presentan estos casos.

¿?	Adopción/Difusión	Rechazo
Innovación eficiente	I	II
Innovación ineficiente	III	IV

Fig. 2. Preguntas y posibilidades de adopción o rechazo de modas

Abrahamson propone cuatro perspectivas que dependen del grado de incertidumbre sobre las metas organizacionales y la eficiencia de la innovación, y del grado de influencia externa al contexto donde se adopta o rechaza la innovación. Tres de las perspectivas responden a las preguntas formuladas, como lo muestra la (Figura 3) [6].

¿?	La imitación NO determina la adopción/rechazo (baja incertidumbre)	La imitación SI determina la adopción/rechazo (alta incertidumbre)
Organizaciones de grupos a los cuales pertenece la organización base, influyen en la adopción/rechazo	 Selección Eficiente	 Moda( <i>fad</i> )
Organizaciones de grupos a los cuales no pertenece la organización base, influyen en la adopción/rechazo	 Selección Forzada	 Moda( <i>fashion</i> )

Fig. 3. Respuestas a las preguntas sobre adopción o rechazo de modas.

La primera perspectiva, llamada selección eficiente, indica que una innovación eficiente se adopta o difunde y una innovación ineficiente se rechaza con criterios objetivos, racionales. Esto sucede cuando la organización tiene total certidumbre sobre sus metas y planes y sobre los beneficios o eficiencia

de la innovación. La segunda y tercera perspectiva son modas *fad* y *fashion*. En estas perspectivas la organización tiene alta incertidumbre sobre sus metas o sobre los beneficios de las innovaciones, por lo que las adopta o rechaza por imitación. Abrahamson diferencia *fad* y *fashion* dependiendo de las influencias externas: mientras *fad* corresponde a imitaciones de organizaciones dentro de un mismo grupo, *fashion* corresponde a imitaciones de organizaciones dentro de grupos externos. Por ejemplo, *fad* ocurre cuando una universidad imita a otra y *fashion* cuando la universidad adopta algo que el mercado le ofrece como novedoso y exitoso. La cuarta perspectiva, llamada selección forzada, se presenta cuando grupos externos influyen altamente en la decisión, por ejemplo cuando disposiciones del gobierno o normatividades exigen adoptar o rechazar una innovación [6].

La Teoría de la Moda se ha usado ampliamente en trabajos de las Ciencias de Gestión y los Sistemas de Información. Estos trabajos reafirman la existencia de modas tanto en la práctica [11–13] como en la academia [14], [15], e invitan a que las universidades sean promotoras de modas y no simples espectadoras y seguidoras [14]. En el caso particular de las Tecnologías de la Información y la Comunicación (TIC), con soporte en la Teoría de la Moda, se puede afirmar que actualmente el mercado de las TIC promueve modas y las universidades las adoptan. Si bien muchas universidades generan conocimiento e innovaciones que luego van a la industria y de allí al mercado, hoy en día hay un crecimiento exponencial y acelerado en muchos campos, al punto que los usuarios finales, entre ellos las universidades, se ven casi obligados a seguir modas y no a liderarlas. En el caso concreto de Cloud Computing, el mercado ofrece diversidad de marcas, productos y servicios y no todos se adoptan. Adicionalmente, como lo plantea la Teoría de la Moda, suelen existir poderes de personas y grupos que lideran el establecimiento de una moda, independiente de los beneficios y sin ningún tipo de criterio objetivo.

De acuerdo a la Teoría de la Moda, es inconveniente adoptar o ignorar innovaciones y tendencias sin criterios objetivos, forzados por estructuras de poder o simplemente por la costumbre de la imitación. Esto es particularmente importante en el contexto

universitario, sobre todo en los procesos curriculares. En ese sentido, para proponer estrategias que orienten la adopción de *Cloud Computing* de manera natural, racional u objetiva y articulada con el currículo existente, a continuación se realiza una revisión de literatura para delimitar el objeto de estudio (la moda) que se quiere adoptar.

## B. Cloud Computing

¿Qué es *Cloud Computing*? ¿Es un concepto nuevo o un concepto olvidado que se retomó recientemente? ¿Es un concepto en la dimensión de las ciencias de la computación, de la ingeniería del software, de las TIC, de los sistemas de información o de economía y administración? ¿Cuál es el origen de *Cloud Computing*? ¿Cuáles los principales conceptos, herramientas, metodologías, tendencias y perspectiva de *Cloud Computing*? De todo lo que existe sobre *Cloud Computing*, ¿qué debe adoptarse en el currículo de Ingeniería de Sistemas?

Para responder estas y otras preguntas se llevó a cabo una revisión de literatura usando las dos fuentes homologadas por el índice Publindex de Colciencias: SCOPUS y Web of Science. Además, se utilizó la herramienta Tree of Science de la Universidad de Manizales [16], [17]. Se realizaron ocho búsquedas diferentes que se muestran en la Tabla 1 con sus resultados:

**Tabla 1.** Búsquedas y Resultados en SCOPUS y WoS

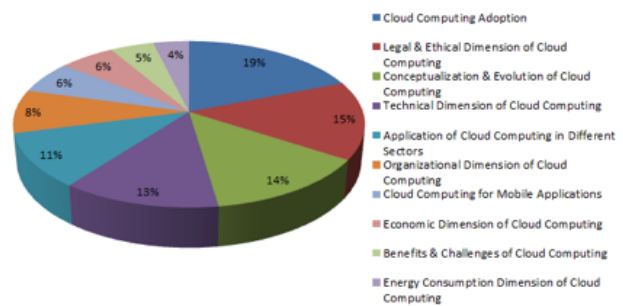
Campos de Búsqueda	Palabra buscada	SCOPUS	WoS
Resumen, título y palabras clave	“Cloud Computing”	13498	2836
	cloud and computing	50028	8444
Sólo título	“Cloud Computing”	12741	<b>2622</b>
	cloud and computing	57835	17173

Para reducir la complejidad por la abundancia de literatura se aplicaron los siguientes filtros: (i) Sólo literatura desde el año 2010 en adelante; (ii) sólo literatura con el término completo “*Cloud Computing*”

en el título, y (iii) categorizado como una revisión o estado del arte (review). Con este filtro el resultado se redujo a 185 documentos en SCOPUS y 101 en WoS. A los filtros anteriores se hicieron excepciones para documentos con alto número de citas o documentos recientes con un estado del arte exhaustivo. Finalmente, usando los filtros anteriores y la herramienta Tree of Science se llegó a un listado de diez documentos.

Cinco de los diez documentos plantean la idea de que Cloud Computing es una moda, usando términos en inglés como *hype*, *buzzword*, *fad* y *fashion* [18–22]. Todos los documentos anteriores al año 2012 plantean que no existe claridad respecto al concepto de Cloud Computing, el cual corresponde a un espectro muy grande que abarca tópicos como: (i) aspectos computacionales y de arquitectura [23–25]; (ii) cambios en los procesos de desarrollo de software [26]; (iii) aplicaciones en la educación [22]; (iv) vistas holísticas [20], [21], [27], [28] con poca especificación y detalle, y (v) conceptualización, origen e historia de Cloud Computing [18–20], [27].

De manera paralela a las publicaciones científicas mencionadas, en el año 2012 apareció un documento de la organización estadounidense NIST (National Institute of Standards and Technology). NIST fijó un precedente desde la industria y el gobierno al estandarizar Cloud Computing como un modelo [29]. La terminología, los paradigmas, conceptos y prácticas del documento NIST se corresponden con los resultados de una revisión publicada en el año 2016. Dicha revisión analizó 236 artículos desde 2009 de Emerald y ABI/INFORM. La revisión encontró que el 14 % de las publicaciones se centraron en conceptualizar y estudiar el origen y evolución de Cloud Computing, 19 % en estudiar su adopción, 13 % trataron aspectos técnicos especializados y 15 % abordaron aspectos éticos, legales y desafíos para la sociedad [21]. La Figura 4 resume todas las categorías o dimensiones de estudio de Cloud Computing según esta revisión.



**Fig. 4.** Dimensiones de estudios sobre Cloud Computing en 236 artículos de Emerald y ABI/INFORM [21]

También en 2016 un artículo planteó tres dimensiones blandas (soft) sobre Cloud Computing, obtenidas después de un estudio con diversos roles de TIC y altos directivos de varias organizaciones: (i) Preocupaciones sobre seguridad de información, en especial por aspectos como terrorismo. (ii) Retos en aspectos estratégicos y de gestión, por cambios en la dirección, administración y gobierno de las TIC, fundamentalmente un cambio de paradigma. (iii) Aspectos legales, éticos y culturales por los cambios que genera el mercado mundial de Cloud Computing, además de los problemas económicos y políticos de monopolios [30].

Según la literatura revisada, se puede proponer la siguiente definición de Cloud Computing: Es un paradigma de paradigmas que consigue agrupar holísticamente diversidad de conceptos y tecnologías de paradigmas previamente existentes, originando un sistema emergente, producto de la sinergia de la colaboración entre grandes empresas y grupos de personas del mercado TIC, que consiguen la integración y unificación de productos y servicios de TIC. Esta definición involucra tres ideas: (i) diversidad de conceptos y tecnologías de paradigmas previamente existentes; (ii) convergencia o integración sinérgica que origina un nuevo sistema, y (iii) la causa raíz de la sinergia, que corresponde al fenómeno social de la colaboración, como consecuencia de la globalización de las TIC, el concepto clásico de una aldea global que en la actualidad se estudia bajo los términos de *social network* y *crowdsourcing* y que aparece, entre otros, gracias a la Web 2.0 y la masificación del uso de las TIC [31], [32].



Para la tercera idea, sobre la sinergia que genera el fenómeno social de colaboración, la Teoría de la Moda de Abrahamson y la Teoría de la Difusión de Innovaciones de Rogers permiten apreciar lo que sucede cuando grupos de personas promueven o lideran y siguen o adoptan innovaciones, en el marco de una dinámica de poderes, como se explicó previamente y como se resumió en las figuras 1, 2 y 3. En la actualidad, consecuencia del auge de Cloud Computing, empresarialmente el mercado de las TIC tiene una clara división en dos grandes grupos: (i) El primero es el mercado gigante de Cloud Computing, que permite colaboración global y agrupa movimientos de software libre, grandes monopolios de TIC y millones de pequeñas empresas y emprendedores. Este grupo puede considerarse el grupo de *—fashion setter—*. (ii) El segundo son los usuarios *—fashion users-adopters—*. En este grupo se ubica cualquier otro sector de la economía y la sociedad, organizaciones y personas dedicadas a consumir los productos y servicios de Cloud Computing, en una dinámica de modas similar a la industria de ropa (*—clothing | garment industry—*). En ese sentido, la dinámica de Cloud Computing como una moda ha conseguido su masificación y crecimiento en los últimos años, y se ha logrado posicionar al punto de convertirse en una necesidad, primero en la industria y después en la academia.

Para las dos primeras ideas de la definición, la literatura muestra que Cloud Computing cubre muchos paradigmas, casi todos los paradigmas desde los inicios de la computación: Computación en Red (Network Computing), Computación de Alto Rendimiento (High-Performance Computing), Computación Paralela (Parallel Computing), Computación Distribuida (Distributed Computing), Computación en Cluster (Cluster Computing), Computación en Malla (Grid Computing), Computación Móvil (Mobile Computing), Bioinformática (BioComputing), Computación cuántica (Quantum Computing), Computación Óptica (Optical Computing) y Nano Computación (Nano Computing) [21–23], [28], [33]. Cloud Computing también reúne tópicos de ingeniería de software como SOA, metodologías ágiles, diseño guiado por modelos; tópicos de TIC como redes de comunicación, Web 2.0, virtualización, gestión de infraestructura; tópicos de sistemas de información,

en especial los tópicos blandos sobre aspectos económicos, sociológicos y culturales, y tópicos de ciencias de la computación como programación paralela, complejidad computacional, teoría de grafos, entre otros. Además, Cloud involucra de manera natural la Inteligencia Artificial (IA), que ya no es un tema de investigadores gurúes sino que se convierte en una utilidad fácil de usar y al alcance de cualquier usuario [20–22], [27], [33]. Por ejemplo: cuando una persona publica una fotografía en cualquier red social, automáticamente se ejecutan algoritmos de IA para reconocimiento de imágenes usando recursos de Cloud Computing.

En la literatura académica es evidente que Cloud Computing es un concepto difuso, “the definition of Cloud Computing is still cloudy” [19]. No obstante, desde el año 2015 las recomendaciones de NIST sobre Cloud Computing han comenzado a adoptarse como estándar y académicamente son siempre un referente conceptual. La Figura 5 muestra un sencillo mapa de conceptos que resume de manera muy general la visión de NIST sobre Cloud Computing.



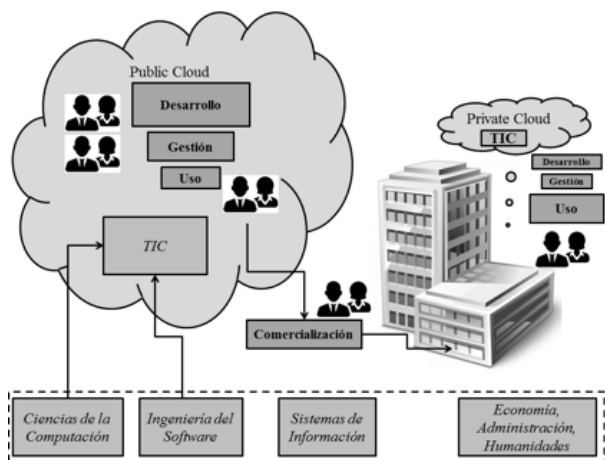
**Fig. 5.** Mapa de Conceptos NIST Cloud. Fuente: Elaboración propia con base en [29].

Explicar esta visión implica manejar más de diez conceptos diferentes, aunque interrelacionados. Además, se debe lidiar con los diversos proveedores, cada uno con arquitecturas y modelos propietarios. Cloud Computing pasa a conformar la lista de términos ambiguos que como sistema, sistema de información e incluso software, representan múltiples conceptos dependiendo del contexto y la perspectiva. Como en la metáfora del poema Los Ciegos y el Elefante de John Godfrey Saxe, según planteamientos de la Disciplina de los Sistemas de Información [34], Cloud Computing no es un concepto sino un cuerpo de conocimiento de muchos conceptos relacionados.

### III. PROPUESTA CURRICULAR

Considerando la revisión de literatura, una asignatura electiva no es suficiente para introducir Cloud Computing en el currículo. Por el contrario, al interpretar Cloud Computing como un paradigma de paradigmas, como un cuerpo de conocimiento, la adopción curricular de Cloud Computing implica modificar casi todas las asignaturas: la programación básica y especializada, las bases de datos, las redes de comunicaciones, los sistemas operativos, la ingeniería del software, la gerencia de TIC tanto de operación como de proyectos, los seminarios, las asignaturas de economía y administración e incluso las humanidades.

Para el caso específico del programa de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander (UFPS) de Cúcuta, su objeto de estudio, misión, visión y proyecto educativo dirigen los esfuerzos educativos hacia la Ingeniería de Sistemas concebida como el “desarrollo y gestión de sistemas de información”, articulando tópicos de ciencias de la computación, ingeniería de software y sistemas de información propiamente dichos, siempre utilizando las TIC como el artefacto principal.



**Fig. 5.** Diseño Curricular “desarrollo y gestión de sistemas de información” bajo el paradigma Cloud Computing. Fuente: elaboración propia.

La Figura 6 ilustra que esta concepción curricular es compatible con la definición de Sistemas de Información como una disciplina intermedia entre la industria TIC y las organizaciones, donde los profesionales pueden desempeñarse en cualquiera

de los extremos o como un intermediario o interfaz [34]. En ese orden de ideas, el currículo articulado con Cloud Computing para el Programa de Ingeniería de Sistemas de la UFPS Cúcuta, debe hacer énfasis en la separación entre la industria de TIC y las demás industrias o tipos de organizaciones, considerando las implicaciones conceptuales y prácticas para la docencia y la investigación.

**Tabla 2.** Diseño curricular “desarrollo y gestión de sistemas de información” bajo el paradigma cloud computing

Tendencia Cloud	Sector Económico Industria TIC			Otros Sectores Económicos		U
	D	G	E	D	G	
1) Herramienta TIC	✓	✓	✓	✓	✓	✓
2) Modelo de Gestión	✗	✓	✗	✗	✓	✓
3) Modelo de Negocio	✓	✓	✓	✗	✓	✓
4) Paradigma (investigación)	✓	✓	✓	✓	✓	✓

D = Desarrollo, G = Gestión, E = Emprendimiento, comercialización, U = Uso

La Figura 6 y la Tabla 2 buscan mostrar que Cloud Computing saca el desarrollo y la gestión de TIC/SI de las organizaciones, esa es la tendencia, de manera que las organizaciones se limitan a usar TIC. En la Tabla 2, el color y la dirección de las flechas y símbolos indican si el componente (desarrollo, gestión, emprendimiento, uso) sale del sector o se puede estudiar desde él. Como lo manifestó Nicholas Carr en su profético artículo del año 2003 (IT doesn't matter) hoy “las TIC son como los servicios de agua, energía, aseo y alcantarillado, un commodity” [35]: El desarrollo y la gestión pasan a la industria de TIC, que hoy domina la economía, pues ya no son viables los modelos “in house”. No obstante, surge la oportunidad del emprendimiento TIC, la posibilidad de ser parte de “la nube”, haciendo desarrollo y gestión de TIC, pero pensando como empresario, capaz de comercializar productos y servicios a organizaciones y personas. Y hacer investigación implica una perspectiva holística, intermedia entre la disciplina de los Sistemas de Información y las Ciencias de la Computación, una perspectiva de paradigma. Entonces, según la tabla, se plantean cuatro maneras o perspectivas para incorporar Cloud Computing al currículo, como se detalla a continuación.

#### A. Perspectiva Pragmática: Como Herramienta

Desde esta perspectiva, sin realizar cambios curriculares, la práctica docente debe comenzar a involucrar la nube. El uso de TIC Cloud (casi siempre

gratuitas), permite al estudiante involucrarse en el mundo globalizado, pasar a integrar redes mundiales y a realizar trabajo colaborativo: En los primeros cursos de programación de computadores, el uso de control de versiones con GIT, de IDE (entornos de desarrollo) y compiladores en línea, comunidades que promueven la programación de computadores, el uso de servidores IaaS en nubes, entre otros, permiten cambiar el paradigma de estudiantes y profesores. Por ejemplo:

- (i) El IDE en la nube de CodeChef  
<https://www.codechef.com/ide>
- (ii) La comunidad BlueJ y Greenfoot  
<https://blueroom.bluej.org/door>  
<https://www.greenfoot.org/home>
- (iii) VPL Laboratorio Virtual de Programación  
<http://vpl.dis.ulpgc.es/>
- (iv) La comunidad GIT  
<https://github.com/>
- (v) Acepta el reto  
<https://www.aceptaelreto.com/>
- (vi) Plataformas de Programación Competitiva  
<https://uva.onlinejudge.org/>

En cursos intermedios de análisis y diseño de sistemas, ingeniería de software, arquitectura de software, gestión de TIC, gerencia de proyectos, se pueden explorar múltiples herramientas que rompen restricciones de tiempo, lugar y recursos computacionales. Pero, además, se puede comenzar a promover el desarrollo de software usando código fuente abierto en millones de repositorios y comunidades mundiales. En lugar de desarrollar software a la medida, desde cero, ahora se adaptan enormes paquetes de software disponibles en la nube y se usan librerías (API) como servicios en línea ofrecidos por los grandes proveedores como Google, Amazon y Microsoft.

En general, esta perspectiva propone que Cloud Computing se incorpore en las diferentes asignaturas como una herramienta (virtual), y paulatinamente el estudiante podrá comprender que detrás de la nube hay todo un cuerpo de conocimiento, personas, organizaciones y oportunidades en un mundo global.

## B. Perspectiva Administrativa: Como Modelo de Gestión

En el caso de Colombia, en el sector oficial, el Ministerio de las TIC y la Agencia Colombia Compra Eficiente fueron pioneros al promover e implementar Cloud Computing (pública y privada) en las entidades estatales (<https://www.colombiacompra.gov.co>). También Fedesoft y ACIS, como agremiaciones colombianas de profesionales de Ingeniería de Sistemas, han promovido la moda Cloud Computing, al igual que muchas universidades en alianza con marcas como Google, Amazon y Microsoft. En el sector privado, Cloud Computing crece cada vez más en virtud de la reducción de costos y los beneficios que ofrece la nube para la gestión de TIC en las organizaciones.

Como consecuencia de la realidad profesional de Cloud Computing, las universidades deben procurar un contexto similar. Ya no se tienen centros de datos ni se desarrolla software “in house”, ahora todo es por servicio (as a Service) en la nube: Infraestructura (IaaS), Plataforma (PaaS) y Software (SaaS). En tal sentido, las asignaturas de gestión de TIC, gerencia de proyectos, ingeniería del software, arquitectura de software, aplicaciones móviles, aplicaciones web, entre otras, deben migrarse a la nube, contextualizando a la importancia en la gestión, a los cambios en las metodologías y técnicas, a la desaparición y aparición de roles, entre otros aspectos. Son tan profundas las transformaciones que ha generado Cloud Computing en el desarrollo de software, la gestión de TIC e incluso el gobierno de TIC, que ha surgido un nuevo término de moda “DevOps”, una tendencia a la agilidad en el desarrollo (Dev) y la operación (Ops) de las TIC.

Esta perspectiva representa un gran desafío académico, pues se deben estrechar los lazos con dos tipos de empresas: empresas de sector TIC y empresas de otros sectores que demandan productos y servicios TIC. En esta perspectiva, el reto docente es el equilibrio entre la rigurosidad académica y la relevancia práctica, que los estudiantes reciban una formación contextualizada a la realidad profesional que les espera.

### C. Perspectiva Económica: Como Modelo de Negocio

En Colombia el Ministerio de las TIC y Colciencias han logrado liderar iniciativas interesantes para promover el emprendimiento en los universitarios y en toda la comunidad. El reto es innovar y crear empresa, generar valor económico para el país, desarrollo, empleo, crecimiento financiero. El Programa de Ingeniería de Sistemas de la UFPS Cúcuta ha liderado en la región los programas de emprendimiento Apps.co y la implementación de Vive Digital Lab, procurando propiciar espacios para innovar y emprender.

Esta perspectiva implica articulación con las dos anteriores: se requiere formación de talento calificado en TIC para un contexto Cloud Computing, conocedor de las transformaciones en los niveles de desarrollo, gestión y gobierno de TIC. La articulación de las perspectivas pragmática, administrativa y económica es clave para tener un ecosistema completo, donde el currículo realmente ofrece una formación pertinente para un mundo en el que la economía es digital y también pertinente para una región como Cúcuta, con serios conflictos sociales de postconflicto, frontera con Venezuela, desempleo y debilidades de carácter político, económico y cultural.

### D. Perspectiva Científica: Como Paradigma

Como se mostró en la Tabla 1 al realizar la revisión de literatura, hay muchas oportunidades y fuerte competencia para hacer investigación y generar impacto en la literatura científica sobre Cloud Computing. Hay posibilidad de investigar problemas complejos (*hard*) de ciencias de la computación y el fenómeno blando (*soft*) en la disciplina de sistemas de información. También hay oportunidades de hacer desarrollo tecnológico en la nube en las disciplinas de TI e ingeniería de software. En este caso, Cloud Computing debe pensarse holísticamente, como un paradigma, hay que profundizar en la extensa literatura, escudriñar problemas recurrentes sin solución, contradicciones en el conocimiento, vacíos y ambigüedades, teorizar, experimentar.

Esta perspectiva puede abordarse en los espacios de seminarios, proyectos de semilleros de investigación

y proyectos de grado. Es la manera más compleja de involucrar Cloud Computing al currículo, la que genera más riesgo de caer en las situaciones inapropiadas que describe la Teoría de la Moda. Al demandar mayor rigurosidad, corresponde más a los ámbitos de maestría y doctorado. Esto no significa que en nivel de pregrado no sea viable, sino que el desafío curricular es más grande.

## IV. CONCLUSIONES Y TRABAJO FUTURO

El presente artículo planteó, soportado en la Teoría de la Moda de Abrahamson [1–4], que al adoptar tecnologías (como Cloud Computing) se corre el riesgo de adoptarlas o ignorarlas sin argumentos objetivos, apelando sólo a la costumbre e imitación o a los poderes que ejerce el mercado y grupos económicos, políticos o incluso académicos (los *fashion setter*). La teoría muestra que hay modas en la práctica [11–13] y en la academia [14], [15], y que las universidades deben ser promotoras de modas y no simples espectadores y seguidores [14]. Según la literatura revisada, Cloud Computing aparece como una moda (en inglés *fashion word*, *buzzword* o *hype*) [18–22]. Entonces es importante considerar las recomendaciones de la Teoría de Abrahamson y evaluar con más detalle su adopción en el contexto académico de la Universidad.

En segundo lugar, para comenzar a ser promotores de la moda de Cloud Computing se realizó una revisión de la literatura usando SCOPUS, WoS y la herramienta en la nube Tree of Science [16], [17]. La revisión corroboró que Cloud Computing es una moda. Además, la revisión permitió identificar falta de claridad y consenso sobre una definición, por lo que se propuso que “Cloud Computing es un paradigma de paradigmas” un “cuerpo de conocimientos”. Como consecuencia de la definición propuesta, se planteó que la adopción de Cloud Computing implica transformaciones en todo el currículo.

Finalmente se propusieron (Figura 6 y Tabla 2) cuatro perspectivas o maneras de articular Cloud Computing al currículo del Programa de Ingeniería de Sistemas de la UFPS, cuyo objeto de estudio, misión, visión y proyecto educativo dirigen los esfuerzos educativos hacia la Ingeniería de Sistemas concebida como el



“desarrollo y gestión de sistemas de información”. Se plantearon y discutieron brevemente cada una de las cuatro perspectivas: (i) pragmática, (ii) administrativa, (iii) económica y (iv) científica.

Este trabajo es un ejercicio de reflexión académica, soportado en la literatura científica y el contexto del Programa de Ingeniería de Sistemas de la UFPS. A futuro este ejercicio debe enriquecerse con trabajo exploratorio y empírico sobre el conocimiento y la percepción y prospectiva de la comunidad académica y la región sobre Cloud Computing y sobre otras modas como DevOps, IoT y Big Data.

Como conclusión final y planteamiento prospectivo, la revisión de la literatura, la reflexión realizada y la percepción de los autores permiten afirmar que la moda de Cloud Computing logró en reducido tiempo transformaciones mucho más grandes que la Web, lo que hoy se conoce como disrupción. Esa disrupción se mantendrá por algunos años más y desafía totalmente la educación universitaria, en especial las universidades pequeñas y de pregrado como la UFPS. Como Programa Acreditado de Alta Calidad, el programa de Ingeniería de Sistemas de la UFPS asume el desafío y en su proceso continuo de autoevaluación y evolución comienza a evaluar, planear e implementar transformaciones profundas en su currículo, para enfrentar la tercera década del siglo en el año 2020.

## REFERENCIAS

- [1] E. Abrahamson, “Managerial fads and fashions: the diffusion and rejection of innovations”, *Acad. Manage. Rev.*, vol. 16, N.º 3, pp. 586-612, 1991.
- [2] E. Abrahamson, “Management Fashion”, *Acad. Manage. Rev.*, vol. 21, N.º 1, pp. 254-285, 1996.
- [3] E. Abrahamson y G. Fairchild, “Management Fashion: Lifecycles, Triggers, and Collective Learning Processes”, *Adm. Sci. Q.*, vol. 44, pp. 708-740, 1999.
- [4] E. Abrahamson, “Necessary conditions for the study of fads and fashions in science”, *Scand. J. Manag.*, vol. 25, N.º 1, p. 235-239, 2009.
- [5] R.-L. Ochoa-Urrego y J.-I. Peña-Reyes, “Teoría de la Difusión de Innovaciones: Evolución y uso en los Sistemas de Información”, presented at the COGESTEC - III Congreso Internacional de Gestión Tecnológica e Innovación, 2012.
- [6] E. Abrahamson, “Managerial fads and fashions: the diffusion and rejection of innovations”, *Acad. Manage. Rev.*, vol. 16, N.º 3, pp. 586-612, 1991.
- [7] E. Abrahamson, “Management Fashion”, *Acad. Manage. Rev.*, vol. 21, N.º 1, pp. 254-285, 1996.
- [8] E. Abrahamson y G. Fairchild, “Management Fashion: Lifecycles, Triggers, and Collective Learning Processes”, *Adm. Sci. Q.*, vol. 44, pp. 708-740, 1999.
- [9] E. Abrahamson, “Necessary conditions for the study of fads and fashions in science”, *Scand. J. Manag.*, vol. 25, N.º 1, pp. 235-239, 2009.
- [10] D. Crane, “Fashion in science: Does it exist?”, *Soc. Probl.*, vol. 16, pp. 433-441, 1969.
- [11] P. Wang, “Chasing the hottest it: effects of information technology fashion on organizations”, *MIS Q.*, vol. 34, N.º 1, pp. 63-85, 2010.
- [12] A. V. Rossem y K. V. Veen, “Manager’s awareness of fashionable management concepts: An empirical study”, *Eur. Manag. J.*, vol. 29, N.º 1, pp. 206-216, 2011.
- [13] Gartner-Group y K. Panetta, “Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017”, 15-Ago.-2017.
- [14] R. Baskerville y M. D. Myers, “Fashion Waves in Information Systems Research and Practice”, *MIS Q.*, vol. 33, N.º 4, pp. 647-662, 2009.
- [15] S. Bort, “Fashion in Organization Theory: An Empirical Analysis of the Diffusion of Theoretical Concepts”, *Organ. Stud.*, vol. 32, N.º 5, pp. 655-681, 2011.
- [16] S. Robledo-Giraldo, G.-A. Osorio-Zuluaga y C. López-Espinosa, “Networking en pequeña empresa: una revisión bibliográfica utilizando la teoría de grafos”, *Vínculos*, vol. 11, N.º 2, pp. 6-16, 2014.
- [17] S. Robledo-Giraldo, G.-A. Osorio-Zuluaga y C. López-Espinosa, *Tree of Science ToS*, 2014.
- [18] L. M. Vaquero, L. Rodero-Merino, J. Caceres y M. Lindner, “A Break in the Clouds: Towards a Cloud Definition”, *SIGCOMM Comput Commun Rev*, vol. 39, N.º 1, pp. 50-55, dic. 2008.
- [19] W. Kim, “Cloud computing: Today and Tomorrow”, *J. Object Technol.*, vol. 8, N.º 1, pp. 65-72, 2009.

- [20] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing - Technical Report No. UCB/EECS-2009-28", University of California, Berkeley, 02-oct.-2009.
- [21] M. Bayramusta y V. A. Nasir, "A fad or future of IT?: A comprehensive literature review on the cloud computing research", *Int. J. Inf. Manag.*, vol. 36, N.º 4, pp. 635-644, 2016.
- [22] J. A. González-Martínez, M. L. Bote-Lorenzo, E. Gómez-Sánchez y R. Cano-Parra, "Cloud computing and education: A state-of-the-art survey", *Comput. Educ.*, vol. 80, pp. 132-151, 2015.
- [23] A. Jula, E. Sundararajan y Z. Othman, "Cloud computing service composition: A systematic literature review", *Expert Syst. Appl.*, vol. 41, N.º 8, pp. 3809-3824, 2014.
- [24] M. Armbrust et al., "A view of cloud computing", *Commun. ACM*, vol. 53, N.º 4, pp. 50-58, 2010.
- [25] A. U. R. Khan, M. Othman, S. A. Madani y S. U. Khan, "A survey of mobile cloud computing application models", *IEEE Commun. Surv. Tutor.*, vol. 16, N.º 1, pp. 393-413, 2014.
- [26] A. Khan, M. Fayaz, A. S. Shah y F. Wahid, "Critical Analysis of Cloud Computing Software Development Process Models", *Int. J. Softw. Eng. Its Appl.*, vol. 10, N.º 11, pp. 451-466, 2016.
- [27] M. Hamdaqa y L. Tahvildari, "Cloud Computing Uncovered: A Research Landscape", *Adv. Comput.*, vol. 86, pp. 41-85, 2012.
- [28] E. Gorelik, "Cloud Computing Models", Massachusetts Institute of Technology, 2013.
- [29] N. I. of S. y T. NIST, "The NITS Definition of Cloud Computing", 2012.
- [30] R. El-Gazzar, E. Hustad y D. H. Olsen, "Understanding cloud computing adoption issues: A Delphi study approach", *J. Syst. Softw.*, vol. 118, pp. 64-84, 2016.
- [31] Y. Zhao y Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction", *Inf. Syst. Front.*, vol. 16, N.º 3, pp. 417-434, jul. 2014.
- [32] O. Zuchowski, O. Posegga, D. Schlagwein y K. Fischbach, "Internal crowdsourcing: Conceptual framework, structured review, and research agenda", *J. Inf. Technol.*, vol. 31, N.º 2, pp. 166-184, 2016.
- [33] K. Chandrasekaran, *Essentials of Cloud Computing*. CRC Press Taylor & Francis Group, 2015.
- [34] R. D. Westfall, "An Employment-Oriented Definition of the Information Systems Field: An Educator's View", *J. Inf. Syst. Educ.*, vol. 23, N.º 1, pp. 63-70, 2012.
- [35] N. Carr, "IT Doesn't Matter", *Harv. Bus. Rev.*, vol. 81, N.º 5, pp. 41-49, 2003.



# Capítulo X

## GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN EN LAS ORGANIZACIONES

*Geiver Vianey Ríos Gutiérrez - gheiver@yahoo.es*

Grupo de investigación en Informática Avanzada MUISCA,  
Fundación Universitaria Juan de Castellanos (Tunja, Boyacá, Colombia).

*John A. Bohada Jaime - jbohada@jdc.edu.co*

Docente investigador, Fundación Universitaria Juan de Castellanos  
(Tunja, Boyacá, Colombia).

*Iván Andrés Delgado González - ivandelgadog@gmail.com*

Docente, decano e investigador, Fundación Universitaria Juan de Castellanos  
(Tunja, Boyacá, Colombia).

### I. INTRODUCCIÓN

La sociedad actual está desarrollando un sentido de responsabilidad y seguridad corporativas que debe ser cumplido estrictamente por las organizaciones, el cual busca un enfoque innovador y especial, que disponga de un conjunto de servicios avanzados de consultoría para alcanzar la seguridad corporativa completa. Al respecto, algunos expertos [1] manifiestan que la seguridad no es solamente implementar usuarios y contraseñas, sino también poner en funcionamiento políticas que garanticen la seguridad tanto física como lógica de la información. Asimismo, hacia los años sesenta empezaron a surgir soluciones tecnológicas para el campo de la seguridad en redes, que sirvieron de base para la generación de soluciones informáticas para la seguridad en grandes escalas, que tuvieron su auge a partir del año 2000 e incluyeron como aspecto importante la protección de la información al considerarla como uno de los activos más valiosos. De esta forma, y tomando como base la protección de la información, en la actualidad la seguridad informática es un aspecto muy importante en las organizaciones, donde los protocolos, las tecnologías (hardware o software), los dispositivos, las herramientas y las técnicas que permiten proteger los datos son ahora esenciales para la disminución

de las amenazas presentes, cuyo único objetivo es socavar la información con fines diferentes a los ya definidos para la organización [2] [3]. Actualmente se puede apreciar que las organizaciones ponen todos sus esfuerzos en prevenir el robo externo y/o interno de información, sin embargo, no tienen en cuenta la seguridad, o desvían su atención respecto a ella, que propiamente deberían tener las redes por las que circula su activo más valioso y las cuales son potencialmente vulnerables ante cualquier daño, intento de robo o modificación.

De igual manera, no deben subestimarse las fallas de seguridad provenientes del interior mismo de las organizaciones, siendo éste un aspecto desconocido y que en muchas ocasiones genera vulnerabilidades que son aprovechadas, incluso, por personal propio de la organización, para cometer actos que ponen en riesgo la integridad y la seguridad de la información. Por ello, la pregunta que ronda en muchos expertos de seguridad es: Teniendo en cuenta los avances tecnológicos y la creciente necesidad de evitar o prevenir los delitos informáticos, ¿qué deben tener en cuenta las organizaciones para aumentar o mantener la seguridad en las redes informáticas y prevenir el robo de información?

En este propósito, este artículo hace un compendio bibliográfico sobre la seguridad en redes corporativas y las políticas de seguridad que deben utilizar las organizaciones para garantizar la integridad de su información en cualquier momento y lugar, teniendo en cuenta que dentro del entorno de la red se debe asegurar la confidencialidad, integridad y disponibilidad de la información sin dejar de protegerla, para prevenir las operaciones de daños ya sean no intencionados o deliberados.

## II. METODOLOGÍA

Para el desarrollo del artículo, se realizó una revisión sistemática de documentos de organizaciones dedicadas a la seguridad y/o tecnología de la información, estudios científicos relacionados con esta área, consultas a bases de datos bibliográficas (Science Direct, Google académico, ProQuest, Scielo, entre otras) en diferentes idiomas (castellano, inglés, portugués), estudios de caso y demás fuentes que nos permitiesen tener una visión de la seguridad en redes corporativas. En el proceso de investigación, se tuvieron en cuenta aspectos como: aportes sobre seguridad de la información, metaanálisis, revisiones sistemáticas y estudios aleatorizados que tuvieran texto completo.

Tras la búsqueda inicial se localizaron aproximadamente 50 estudios, se organizó en una matriz sistemática de acuerdo con la pregunta y las palabras claves que se utilizaron para su búsqueda. Para proceder a la selección, se revisaron los resúmenes y en caso necesario los artículos completos con el fin de decidir si la información que contenían estaba o no relacionada con el tema de investigación del artículo. Posteriormente, se diseñó una matriz para el análisis de cada uno de los artículos seleccionados y la información analizada se estructuró de dos formas: en primer lugar, se tuvo en cuenta la revista o libro, título, autor, año, ciudad o país, páginas, editorial, URL, palabras clave, principales áreas temáticas, resumen, problema de investigación, metodología, tipo de estudio, instrumentos y técnicas, población y resultados. En segundo lugar, se tuvo en cuenta la referencia, tipo de estudio, población, intervención, comparación (si la hay), tiempo de seguimiento, método de evaluación, resultados, conclusión, comentario y evaluadores.

## III. RESULTADOS

### A) Gestión de Seguridad en las Organizaciones

Toda organización con un sistema de información seguro es un competidor fuerte en el mercado, sin embargo, puede ser un recurso empresarial en alto riesgo dadas las maniobras delictivas de terceros que siempre están al acecho de cualquier vulnerabilidad para ingresar y realizar sus actos delictivos [4]. Por tanto, para toda organización es indispensable implementar medidas para prevenir y proteger sus activos y dar garantía de disponibilidad, privacidad y veracidad de la información. La gestión del riesgo es la forma más adecuada de proteger los activos de información, y así lograr identificar y focalizar esfuerzos hacia aquellos elementos que se encuentren más expuestos [5]. En este sentido, se han propuesto metodologías que contribuyen al mejoramiento de la seguridad de las empresas, particularmente en el área de las Tecnologías de la Información (TI). Al respecto, en la Tabla 1 se muestra una de ellas, y se describe el proceso general para desarrollar la prevención de vulneración a la información de una organización, y cómo mediante cinco fases se contribuye a mejorar la confidencialidad, integridad y disponibilidad de la información y, por supuesto, de las TI que la contienen [6-9].

**Tabla 1.** Propuesta de Metodología para Prevención de Riesgos

Fase	Actividad
<b>Cero:</b> Requerimientos de inicio	<ul style="list-style-type: none"> <li>– Apoyo de la dirección</li> <li>– Selección del personal adecuado.</li> </ul>
<b>Uno:</b> Conocimiento de la imagen corporativa de la empresa	<ul style="list-style-type: none"> <li>– Identificación de la situación actual de la organización.</li> <li>– Obtención de la estructura jerárquica de la organización, incluyendo el área de TI.</li> <li>– Análisis del área de TI.</li> </ul>

Fase	Actividad
<b>Dos:</b> Identificación y análisis de la tecnología de la información de la empresa	<ul style="list-style-type: none"> <li>– Identificación de las relaciones y funciones de la organización, incluida el área de TI.</li> <li>– Identificación y análisis de los vacíos que pueda tener el área de TI en sus procesos.</li> </ul>
<b>Tres:</b> Diagnóstico de brechas operacionales y riesgos, de la tecnología informática	<ul style="list-style-type: none"> <li>– Identificación de los riesgos y amenazas que puedan existir en la organización y en el área de TI.</li> <li>– Evaluación de los riesgos y la posible técnica que debe ser utilizada.</li> <li>– Priorización de los riesgos y definición de la estrategia para contrarrestarlos.</li> </ul>
<b>Cuatro:</b> Diseño y desarrollo de actividades de control mediante una matriz.	<ul style="list-style-type: none"> <li>– Identificación de actividades de control en los procesos de TI.</li> <li>– Diseño de actividades de control en los procesos de TI.</li> <li>– Desarrollo de una matriz de riesgos, donde los objetivos, el control y los riesgos son su base.</li> </ul>
<b>Cinco:</b> Seguimiento a las actividades y capacitación del recurso humano	<ul style="list-style-type: none"> <li>– Realización de procesos de implantación de las actividades de control y preparación del personal para tal fin.</li> <li>– Realización de procesos de monitoreo de las actividades implantadas.</li> </ul>

## B) Amenazas contra la seguridad de la Información en las organizaciones

Citando textualmente del glosario de la ISO 27000 [5], la amenaza “es la causa potencial de un incidente no deseado, el cual puede causar el daño a un sistema o la organización”: amenaza lógica, aquella que afecta a la información almacenada en los activos; amenaza física, en la cual el atacante tiene diferentes tipos de acceso físico a la organización y puede ocasionar problemas [6], por lo cual, el análisis y la evaluación de riesgos, la verificación de la existencia de controles de seguridad, las pruebas con software y el monitoreo de los sistemas de información permiten establecer el estado actual de la organización, identificar las causas de vulnerabilidades y proponer soluciones de control que permitan su mitigación [7], aspectos importantes que se deben tener en cuenta en el área de TI de la organización. La Tabla 2 muestra las amenazas a las que está expuesta la información en una organización, las cuales se hacen desde su respectiva área, sus vulnerabilidades, sus amenazas y los riesgos potenciales que éstas ocasionan [8],[9].

**Tabla 2.** Amenazas y Riesgos Potenciales para el Activo Informático en la Organización

Área	Vulnerabilidad	Amenazas	Riesgos potenciales
<b>Hardware</b>	Falta de equipos o unidades de respaldo (UPS's) para contingencias.	Cortes de energía o sobrecargas.	Pérdida o daños de información.
<b>Software</b>	Software no licenciado o con problemas de seguridad.	Virus informáticos, malware y ataques de inyección.	Mal funcionamiento de sistemas. Intrusión no autorizada en los equipos.

Área	Vulnerabilidad	Amenazas	Riesgos potenciales
Seguridad Física	Falta de control a los equipos Informáticos.	No hay control de acceso, ataques intencionados.	Robo, destrucción, modificación o borrado de información.
Seguridad lógica	Deficiente control de acceso.	Suplantación de identidad	Robo, alteración, destrucción, de datos.
Redes de comunicaciones	Vulnerabilidad de navegadores utilizados.	Inyección de código SSI (Server Side Include), ataques con código XSS (Cross-Site Scripting)	Alteración de programas y sitios internos
Personal	Falta de una política de seguridad.	Ataques no intencionados.	Borrado, eliminación o destrucción de información.

Por otra parte, los delitos informáticos abarcan una gran variedad de modalidades [10-15], tales como, ataques contra sistemas y datos informáticos, usurpación de la identidad, distribución de imágenes de agresiones sexuales contra menores, estafas a través de Internet, intrusión en servicios financieros en línea, difusión de virus Botnets (redes de equipos infectados controlados por usuarios remotos), phishing (adquisición fraudulenta de información personal confidencial), entre otros. Se concluye que en la actualidad las organizaciones son cada vez más dependientes de sus redes informáticas y un problema que las afecte, por mínimo que sea, puede llegar a comprometer la continuidad de las operaciones. De esta forma, la carencia de medidas de seguridad en las redes es un problema que está en crecimiento. Cada vez es mayor el número de atacantes y cada vez están más organizados, por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios [16].

Los delitos informáticos pueden seguir en aumento. En el año 2016, la cantidad de usuarios de Internet ascendió a los 5 billones y, por supuesto, la de dispositivos, a los 20 billones, sumado a las empresas que nacen o crecen y necesitan conectar toda su infraestructura al mundo y requieren un espacio en la red [17],[18]. Teniendo en cuenta lo anterior, el aumento de los riesgos informáticos se da como consecuencia del crecimiento de la práctica delictiva por las nuevas aplicaciones y servicios, y las organizaciones han tenido que adaptar nuevas y mejores tecnologías para ofrecer calidad, seguridad y confiabilidad a sus clientes.

Ahora bien, los riesgos se pueden expandir porque la comunicación entre dispositivos se encuentra en constante aumento, En sus inicios, la comunicación se basaba en sistemas como el SCADA (Supervisory Control And Data Acquisition), diferente a los que actualmente se han desarrollado con nuevos conceptos como el internet of things [19], [20], de ahí que, si no se toman las medidas adecuadas, un atacante externo podría hacerse pasar por un local posiblemente burlando los controles de seguridad que se hacen cumplir sólo en comunicaciones de ciertos formatos. Sin embargo, quienes trabajan en el mundo empresarial deben recibir instrucciones claras y definitivas que los ayuden a garantizar la seguridad de la información en el complejo mundo de los negocios, por ello cada vez encontramos más gerentes interesados en entender las reglas del negocio, entre ellas las referentes a las políticas de seguridad informática [21].

### C) Seguridad Integrada-Predictiva para las Organizaciones

Si partimos del hecho de que no existe una solución definitiva para proteger un activo informático por el cambio constante de las maniobras de vulneración [7], una cultura de protección a la información no es una simple función dentro de la organización sino que requiere de acciones constantes de prevención [8], esto exige que las asignaciones presupuestales para esta inversión se sostengan e, incluso, se incrementen de forma sostenida para mejorar sistemas tecnológicos, infraestructura y espacios de capacitación para los involucrados en este proceso [9]. No se puede negar la necesaria protección de la información por parte

de los diferentes entes organizacionales, porque esta materia prima (la información) será el motor de la nueva sociedad, y en torno a ella surgirán profesiones y trabajos nuevos, o se readaptarán las profesiones existentes [10]. En la Tabla 3 se relacionan y describen los mecanismos que pueden evitar y prevenir riesgos de afectación a la información de la organización [11].

**Tabla 3.** Mecanismos de Prevención y Detección de Amenazas en Seguridad de la Información

Mecanismo	Descripción
Unicidad	Incluir un número de secuencia, fecha/hora, número aleatorio para verificar la integridad de los mecanismos.
Control de enrutamiento	Enviar información por zonas clasificadas y solicitar y establecer rutas alternativas en caso de violaciones de seguridad.
Tráfico de relleno	Enviar tráfico falso junto con los datos válidos, para que el atacante no pueda diferenciar los reales de los simulados
Cifrado/Encriptación	Obtener un texto cifrado, inentendible para entidades no autorizadas.
Gestión de claves	Generación, distribución, almacenamiento, tiempo de vida, destrucción y aplicación de las claves de acuerdo con la política de seguridad aplicada.
Cortafuegos	Aplicaciones o equipos ubicados entre dos redes que establecen la política de acceso entre las partes.
Filtrado de paquetes	Lectura pormenorizada de la cabecera de cada paquete, y en función a una serie de reglas se permite el paso de los paquetes o dicha trama es descartada.

Mecanismo	Descripción
Proxy de aplicación	Software encargado de eliminar las conexiones a servicios tales como FTP o Telnet, y permiten únicamente la utilización de servicios en donde se encuentra un proxy.
Monitoreo de la actividad	Seguimiento a la actividad de las aplicaciones de seguridad, para detectar a tiempo los ataques a los que puede estar siendo sometido el sistema.

#### D) Metodologías y mecanismos de prevención de riesgos para la seguridad de la información en las Organizaciones

Es fundamental relacionar parámetros metodológicos generales para prevenir, corregir y combatir los riesgos y vulnerabilidades a los que puede estar sometido el activo informático de las organizaciones. En relación con ello, existen diferentes metodologías que se articulan con los ISMS (information security management system) y permiten la planificación para la disminución de riesgos de seguridad de la información, para la prevención, visualización y detección de vulnerabilidades existentes en los sistemas que componen o hacen parte de las organizaciones y, en fin, ayudan en la toma de decisiones con mayor asertividad en materia de seguridad de la información. Ejemplos de metodologías podemos encontrar: Octave (Operationally Critical Threat, Asset, and Vulnerability Evaluation), Magerit (Methodology for Information Systems Risk Analysis and Management), Mehari (Methode Harmonisée d'Analyse de Risques), NIST SP 800-30 (National Institute of Standards and Technology), Coras (Construct a Platform for Risk Analysis of Security Critical Systems), Cramm (CCTA Risk Analysis and Management Method) o la metodología para la evaluación del riesgo Ebios, entre otras [7-10].

Tomando como base las metodologías anteriormente mencionadas, se pueden establecer algunas recomendaciones para analizar y evaluar riesgos en seguridad de las herramientas tecnológicas



de la organización. En general, se sugiere iniciar por diagnosticar la situación presente para establecer las diferentes vulnerabilidades o riesgos potenciales, apoyándose en las normas vigentes. En la Tabla 4 se muestra un ejemplo sobre la metodología que se debe seguir para evaluar riesgos al activo informático [9][22].

**Tabla 4.** Metodología que se Debe Seguir para Evaluar Riesgos al Activo Informático

Actividad
Diagnóstico inicial de la organización y, en especial, de su área de TI.
Diseño, implementación e implantación de un Sistema de Gestión de Seguridad de la Información (SGSI) alineado, en lo posible, al estándar ISO/IEC 27001.
Entender la norma ISO/IEC 27001 para determinar el alcance de su aplicabilidad.
Aplicar la metodología para realizar análisis y evaluación de riesgos con respecto a los criterios de información confidencialidad, integridad y disponibilidad de la información.
Verificar los controles de seguridad existentes en la organización y su aplicación. Este resultado servirá de base para el diseño, la implementación e implantación futuras de un SGSI como respuesta a los riesgos encontrados.
Explicar la metodología para aplicar el proceso de análisis y evaluación de los riesgos desde la fase inicial de conocimiento del sistema.
Establecer un sistema de control acorde a los hallazgos encontrados.

### E) Riesgos y soluciones en las Redes de una organización

Es importante que toda organización cuente con herramientas tecnológicas que garanticen la valoración de los riesgos a los cuales está sometida la información, e implementar procedimientos de solución que mitiguen su afectación. En la Tabla 5 se muestran los principales riesgos a los que está sometida la organización y una de las posibles soluciones.

**Tabla 5.** Riesgos y Solución para las Redes Corporativas

Riesgos de la información en una red corporativa	Solución a los riesgos a las que están expuestas las redes corporativas en una organización
<b>Riesgos de integridad:</b> asociados con la autorización, completitud y exactitud de la entrada, procesamiento y reportes de las aplicaciones utilizadas en una organización.	Para proteger la información en una red corporativa, una de las estrategias más utilizada es la técnica en capas [23] [24], la cual plantea que la seguridad de la red de una organización se debería basar en un conjunto de barreras que defiendan el negocio de diferentes maneras. Es así como:
<b>Riesgos de relación:</b> uso oportuno de la información creada por una aplicación.	<ul style="list-style-type: none"> <li>– Si falla una de las soluciones, se mantengan otras que protejan la compañía y sus datos de una gran variedad de ataques.</li> </ul>
<b>Riesgos de utilidad:</b> pueden ser enfrentados por el direccionamiento de sistemas antes de que los problemas ocurran.	<ul style="list-style-type: none"> <li>– Se protege la red contra ataques externos e internos.</li> </ul>
<b>Riesgos en la infraestructura:</b> en la organización no existe una estructura de información tecnológica efectiva.	<ul style="list-style-type: none"> <li>– Una de las capas sería la encargada de monitorear el tráfico de red y detectar anomalías en la actividad.</li> </ul>
<b>Riesgos de seguridad general:</b> Riesgos de choque eléctrico, incendio, niveles inadecuados de energía eléctrica, radiaciones, riesgos mecánicos.	<ul style="list-style-type: none"> <li>– Garantiza la privacidad de las comunicaciones en cualquier lugar y en cualquier momento.</li> <li>– Control exhaustivo de los usuarios que acceden a la red, que brinda la seguridad necesaria para acceder a la información evitando los riesgos.</li> </ul>



Particularmente, y haciendo referencia únicamente a la red corporativa, es importante tener en cuenta que la accesibilidad a las diferentes redes es la proporcionada por el crecimiento de modelos como el IP (Internet protocol) [23], la nueva generación de sistemas móviles que harán más fácil las comunicaciones y los negocios. Asimismo, estamos a la puerta de una nueva era donde se imponen las comunicaciones sin cable NGN (Next-Generation Networks), que permitan servicios de comunicación, información y entretenimiento, con mayor velocidad y eficacia [25].

Lo cierto es que en las actuales redes se puede conectar todo tipo de interesados sin importar el sitio y el momento, por esto es importante confeccionar un estudio muy detallado que permita identificar los riesgos a los que se encuentra expuesta la empresa, de esta forma se puede establecer cuál es la manera correcta de implementar medidas para contrarrestar esta deficiencia y salvaguardar los activos; una forma eficaz para realizar estos procesos es un análisis para la implementación de un Sistema de Gestión de Seguridad de la Información [26]. Los expertos aseguran que no existe ningún sistema de seguridad totalmente infranqueable, porque en los mecanismos delincuenciales también hay constantes cambios que buscan penetrar las barreras de control y acceso a la información [27]. Finalmente, se debe ratificar que, sin importar la clase de organización, es una necesidad real la existencia de un sistema de gestión de seguridad [28], es decir, si no se pierde el carácter confidencial del activo informático, la empresa corre el riesgo de desaparecer.

#### IV. DISCUSIÓN

Teniendo en cuenta que la información es un recurso que, como el resto de los activos, tiene valor para la organización y por consiguiente debe ser debidamente protegida, las políticas que se deberían seguir para garantizar su integridad exigen el establecimiento, seguimiento, mejora continua y aplicación de mecanismos que protejan la estructura de la información, para garantizar un compromiso ineludible de su protección frente a una amplia gama de amenazas. Las políticas de seguridad constituyen un conjunto de procedimientos, normas y directrices que sirven de ruta y definen los criterios de seguridad

para que sean adoptados, tanto a nivel local como organizacional, y cuyo objetivo sea el de establecer, estandarizar y normalizar la seguridad no sólo en el ámbito humano, sino también en el tecnológico [29].

La implementación de medidas de seguridad debe ser un proceso que abarque toda la organización, sin exclusión alguna, asegurar la confidencialidad de su información debe ser un proceso continuo, además, se debe prestar mucha atención a las posibles deficiencias en la seguridad de la red corporativa, porque un gran porcentaje de la información vital, por obligación, pasará por dicha red [30], [31]. La convergencia entre las redes, el incremento de la conectividad a otras redes y el uso de redes sociales corporativas hacen que cada vez existan muchas vulnerabilidades, que si no son tenidas en cuenta o la organización no toma los correctivos al respecto, la información vital para la continuidad del negocio se verá claramente afectada [32], [33].

De acuerdo con lo anterior, es claro que dominar la información dentro de una organización es cada vez más importante y se ha convertido en un requerimiento necesario y estratégico para la toma de decisiones, por tal razón, en la actualidad las organizaciones, tanto del sector público como del privado, están priorizando la seguridad de la información, que es catalogada como uno de los bienes más preciados para la continuidad del negocio y el punto de diferencia con la competencia, y que la posibilidad que tiene la información corporativa de ser vulnerada es variada a pesar de la implementación de controles para seguridad de acceso, seguridad de dispositivos, manejo de contraseñas, control de vulnerabilidades, etc.; cada uno de estos requiere un estudio, un presupuesto y una aplicación, ya sea preventiva o correctiva, sobre los temas de seguridad que se puedan encontrar. Además, es imposible encontrar sistemas completamente seguros, ya que cada día se descubren nuevos riesgos en distintos niveles [34-36]. Como vemos, por el valor que se le atribuye a la información, esta es objeto de diversas amenazas como robo, falsificación, fraude, divulgación y destrucción, entre muchas otras [37]. Las herramientas de los ciber-delincuentes han evolucionado si no más rápido, por lo menos paralelamente al desarrollo tecnológico, como ha venido sucediendo con los virus

informáticos. De igual forma, el almacenamiento de los datos constituye otro aspecto para tener en cuenta, y en muchas situaciones supone la posibilidad para un atacante de escalar privilegios tras una intrusión. No es desconocido para los delincuentes que en la actualidad los sistemas de transmisión e interconexión presentan varias vulnerabilidades, entre ellas, la facilidad de analizar tráfico que permite una tasa alta de ataques a los protocolos internos, por ello es necesario que servicios especializados realicen la implementación de redes resistentes, con características propias de la seguridad informática, sin importar el sistema operativo o propietario de los clientes finales [38].

La percepción sobre la idea de la seguridad informática ha estado en boga a nivel empresarial en los últimos años y se ha convertido en una necesidad, más que en un compromiso, frente a una tendencia competitiva cada vez más enfocada al uso de nuevas tecnologías y por ende a mejorar su participación en la Internet. De acuerdo con esto, en la actualidad las redes corporativas y las no corporativas son cada día más complicadas de administrar y controlar [39-44]. Los escenarios presentan cada vez más circunstancias de mayor dificultad en cuanto a la conectividad y su seguridad. Asimismo, la seguridad no es solamente crear usuarios y contraseñas, es implementar políticas que garanticen la seguridad tanto física como lógica de la información. Se debe asegurar la privacidad de la información y protegerla tanto de daños no intencionados como deliberados [45].

Por ello, además de tener clara la protección que se debe dar a la información según el medio en que se transporta, es decir, de la red corporativa como se indicó en la sección III.E, es importante que en la organización se definan unas políticas serias de protección de todos sus activos. La guía que propone el Ministerio de las Tecnologías de la Información y la Comunicación (Mintic) es una clara alternativa para mejorar dicha seguridad, donde la sensibilización a los empleados cumple un papel importante dado que son fundamentales en la prevención y protección de la información, la identificación de los activos de información de la organización, la determinación de las posibles vulnerabilidades y amenazas de la red corporativa o de la arquitectura tecnológica que procesa la información. La evaluación de los riesgos a

los que están expuestos esos activos permitirá generar procedimientos y estrategias que ayuden a prevenir y proteger la información de cualquier ataque ya sea interno o externo a su activo más importante: la información [46-50].

## V. CONCLUSIONES

El creciente auge que tienen las redes informáticas ha sido de enormes proporciones y por ello la seguridad también ha crecido a la par, día a día se trata de buscar la mejor forma para estar protegidos, pero al mismo tiempo, en cuanto sale algún software o hardware, personas mal intencionadas, conocidas como piratas informáticos, hackers, etc., encuentran los puntos débiles, y con estudio y gran intelecto crean programas para explotar dichas debilidades para su beneficio. Por esto ha sido indispensable crear contramedidas para evitar estos ataques y resguardar la información.

Igualmente, las políticas de seguridad, las cuales son realmente necesarias para todo tipo de organización cuyo activo más importante sea la información, e incluso para todos aquellos que posean un computador y en él guarden su información por medio de la red de redes, es decir, la Internet, ya que los atacantes emplean métodos iguales o casi iguales para cometer esos actos ilícitos. Por tanto, se hace indispensable crear e implementar políticas de seguridad para incrementar los niveles de seguridad al realizar actividades que a veces se piensan seguras, pero, tal vez, sean todo lo contrario.

El cumplimiento de la implementación de normas de seguridad, reconocida a nivel internacional como ISO 27001-2013, contribuye al fortalecimiento de controles de protección de la información de la organización, dado que por ser una norma estándar, ha sido probada y ajustada para garantizar el éxito del cumplimiento de los objetivos para la cual fue creada.

La información simboliza uno de los activos más significativos que una empresa tiene, y por eso se busca la protección necesaria para que no sea vulnerada, de manera que se deben evitar ataques maliciosos que alteren la integridad de la información existente en la red corporativa de la entidad.

Toda organización debe contar con políticas e instrumentos que garanticen la valoración de los riesgos a los cuales está sometida la información corporativa, implementar procedimientos de solución que minimicen la afectación de éstos, y crear o fortalecer las áreas TI que permitan hacer seguimiento y control para garantizar la integridad, confidencialidad y protección de su activo más valioso.

## REFERENCIAS

- [1] L. T. Angarita, A. L. Suárez y H. R. Rodríguez, Diseño del Plan de Gestión de Seguridad de la Información para controlar el acceso a las áreas restringidas de la empresa Ingepec Ltda. en la ciudad de Ocaña, Tesis Doctoral, 2014.
- [2] J. Joskowicz, Redes de Datos, Montevideo: Universidad de la Republica, Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, 2008.
- [3] P. Portilla y D. Oswaldo, Diseño de un sistema de gestión de seguridad de la información (SGSI) para el área de informática de la cooperativa del magisterio de Túquerres bajo la norma ISO 27001: 2013, 2015.
- [4] J. Melchor M., J. Lavín V. y N. A. Pedraza, “Seguridad en la administración y calidad de los datos de un sistema de información contable en el desempeño organizacional”, Contaduría y Administración, vol. 57, N.º 4, pp. 11-34, 2012.
- [5] T. V. Guachi A., “Norma de seguridad informática ISO 27001 para mejorar la confidencialidad, integridad y disponibilidad de los sistemas de información y comunicación en el Departamento de Sistemas de la Cooperativa de Ahorro y Crédito San Francisco. Ltda., Tesis de pregrado, Universidad Técnica de Ambato, 2012.
- [6] A. C. Núñez, “Conceptos de seguridad informática y su reflejo en la cámara de cuentas de Andalucía”, Auditoría Pública: Revista de los Órganos Autónomos de Control Externo, N.º 61, pp. 111-117, 2013.
- [7] F. N. S. Solarte, E. R. E. Rosero y M. del C. Benavides, “Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001”, Revista Tecnológica-ESPOL, vol. 28, N.º 5, 2015.
- [8] M. Miranda Cairo, O. Valdés Puga, I. Pérez Mallea, R. Portelles Cobas y R. Sánchez Zequeira, “Metodología para la Implementación de la Gestión Automatizada de Controles de Seguridad Informática”, Revista Cubana de Ciencias Informáticas, vol. 10, N.º 2, pp. 14-26, 2016.
- [9] D. Prado Oseguera, “Metodología para el Establecimiento de Objetivos de Control como un Medio de Seguridad en el Área de Tecnologías de Información”, Tesis Doctoral, 2010.
- [10] H. Alemán y C. Rodríguez, “Metodologías para el análisis de riesgos en los SGSI”, Revista Especializada de Ingeniería, UNAD, 9, pp. 73-86, 2015. Disponible en: <http://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1435>
- [11] Ernst and Young, “Seguridad de la información en un mundo sin fronteras. Es momento de replantear el tema”, Mancera, S.C., 2012.
- [12] J. C. Andrade, “Estudio de los beneficios de la implementación del modelo de gestión cloud computing en comparación al modelo de gestión tradicional. Caso de estudio: Corporación ADC-HAS Management Ecuador S. A.”, Tesis de Maestría, Universidad Andina Simón Bolívar, Sede Ecuador, 2013.
- [13] C. Belloch, “Las Tecnologías de la Información y Comunicación en el aprendizaje”. Material docente [En-línea],. Departamento de Métodos de Investigación y Diagnóstico en Educación. Universidad de Valencia, 2012. Disponible en: <http://www.uv.es/bellochc/pedagogia/EVA1.pdf>
- [14] F. N. S. Solarte, E. R. E. Rosero y M. del C. Benavides, “Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001”, Revista Tecnológica-ESPOL, vol. 28, N.º 5, 2015.
- [15] J. A. Loredó G. y A. Ramírez G., “Delitos informáticos: su clasificación y una visión general de las medidas de acción para combatirlo”, Celerinet, pp. 44-51, Ene.-Jun., 2013.
- [16] D. Maques y A. Marcano, Modelo de estrategias integrales de seguridad para la infraestructura de red de datos. Caso de estudio Universidad de Oriente, núcleo MONAGAS, 2012.
- [17] J. C. Becerra C., J. R. Simbaqueva B. y A. F. Valenzuela S., Diseño e implementación de redes IPv6 en MIPYMES: caso laboratorio de informática, 2013.

- [18] L. P. Aguirre, F. González y D. Mejía, “Aplicaciones de MPLS, transición de IPv4 a IPv6 y mejores prácticas de seguridad para el ISP Telconet”, *Revista Politécnica*, N.º 32, 2013
- [19] S. E. Abasolo Aranda and M. A. Carrera Paz y Miño, “Evaluación del modelo de referencia de Internet of things (IoT), mediante la implantación de arquitecturas basadas en plataformas comerciales, open hardware y conectividad IPv6”, Tesis de pregrado, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador, 2014.
- [20] S. Zylberberg, “Riesgos, problemáticas y amenazas en las redes sociales”, Universidad de Palermo, 2012.
- [21] F. Gont, “Security implications of IPv6 on IPv4 networks”, 2014.
- [22] J. P. Sarubbi, “Seguridad informática. Técnicas de defensa comunes bajo variantes del sistema operativo Unix”, Tesis, Universidad Nacional de Luján, Buenos Aires, 2008, pp. 3-4.
- [23] C. Villabona, W. E. Castellanos y H. Ortega, “Hacia las Redes de Próxima Generación”, Simposio de Investigación y Desarrollo de Electrónica y Telecomunicaciones, 2012.
- [24] C. E. Vinocungo V. y J. B. Rubio, “Implementación de seguridades basadas en capas con tecnología cisco para mejorar la seguridad de la información y productividad de la empresa Prodemco S.A. en la provincia del Azuay, capital Cuenca en la calle Nicanor Aguilar y Luis Moreno mora en el período 2014-2015”, Tesis de pregrado, Latacunga/UTC/2015, 2015.
- [25] M. G. Molina Villacis, “Análisis de la utilización de calidad de servicio (QoS) en redes de nueva generación (NGN) en el Ecuador”, Tesis de maestría, Espol, 2016.
- [26] G. Cepeda y L. Enrique, “Análisis para la implementación de un sistema de gestión de seguridad de la información según la norma ISO 27001 en la empresa Servidoc S. A.”, 2016.
- [27] K. D. R. Gaona, “Aplicación de la metodología Magerit para el análisis y gestión de riesgos de la seguridad de la información aplicado a la empresa Pesquera e Industrial Bravito S.A. en la ciudad de Machala”, Tesis de pregrado, 2013.
- [28] C. A. Dussán Clavijo, “Políticas de seguridad informática”, *Entramado*, vol. 2, N.º 1, 2006.
- [29] A. A. Angarita Vivas y C. A. Tabares Isaza, “Análisis de riesgos para el proceso administrativo: Departamento de informática en la empresa de Acueducto y Alcantarillado de Pereira S.A. ESP, basados en la Norma ISO 27005”, Tesis de pregrado, Pereira: Universidad Tecnológica de Pereira, 2012.
- [30] A. López Novas, “Un nuevo paradigma social en torno a las tecnologías de la información: la sociedad red”, Tesis de pregrado, 2015. Disponible en: [http://uvadoc.uva.es/bitstream/10324/17568/1/TFG\\_F\\_2015\\_22.pdf](http://uvadoc.uva.es/bitstream/10324/17568/1/TFG_F_2015_22.pdf)
- [31] L. B. Valderrama, “Jóvenes, ciudadanía y tecnologías de información y comunicación. El movimiento estudiantil chileno”, *Revista Latinoamericana de Ciencias Sociales, Niñez y Juventud*, vol. 11, N.º 1, 2013.
- [32] J. Kamlofsky, S. Abdel Masih, H. Colombo, D. Veiga y P. Hecht, “Ciberdefensa de infraestructuras industriales”, en XVII Workshop de Investigadores en Ciencias de la Computación, Salta, 2015.
- [33] S. Zylberberg, “Riesgos, problemáticas y amenazas en las redes sociales”, 2012.
- [34] A. Polanco Vélez, “Diseño de un manual de procedimientos del sistema contable en la empresa Fevecomex SAS, basado en la norma técnica colombiana para la seguridad de la información NTC ISO/IEC 27001/2006”, Tesis Doctoral, Universidad de Cartagena, 2014.
- [35] M. A. Mendoza, “Ocho pasos para hacer una evaluación de riesgos”, *Seguridad Corporativa*, 2014. Disponible en: <http://www.welivesecurity.com/la-es/2014/07/29/por-quenecesariofirewall-entornos-corporativos/>
- [36] W. Flórez, C. A. Arboleda y J. F. Cadavid, “Solución integral de seguridad para las pymes mediante un UTM”, *Ingenierías USBmed*, vol. 3, N.º 1, pp. 35-42, 2012.
- [37] P. Sánchez Fernández, “Sistema de Gestión de la Ciberseguridad Industrial”, Tesis de maestría, Universidad de Oviedo, 2013.
- [38] J. Kamlofsky, S. Abdel Masih, H. Colombo, D. Veiga y P. Hecht, “Ciberdefensa de infraestructuras industriales”, en XVII Workshop de Investigadores en Ciencias de la Computación, Salt, 2015.

- [39] O. N. Carrasco y A. V. Puerta, “Una visión global de la ciberseguridad de los sistemas de control, Revista SIC: ciberseguridad, seguridad de la información y privacidad, N.º 106, pp. 52-55, 2013.
- [40] J. Hidalgo, D. P. Tupiza y T. Sánchez, “Políticas de Seguridad de una Red Multiservicios Corporativa utilizando Hacking Ético”, 2016. Disponible en: [https://www.researchgate.net/publication/306219849\\_Politicas\\_de\\_Seguridad\\_de\\_una\\_Red\\_Multiservicios\\_Corporativa\\_Utilizando\\_Hacking\\_Etico](https://www.researchgate.net/publication/306219849_Politicas_de_Seguridad_de_una_Red_Multiservicios_Corporativa_Utilizando_Hacking_Etico)
- [41] P. Alpala y L. Olmedo, Propuesta de actualización, apropiación y aplicación de políticas de seguridad informática en una empresa corporativa, Propolsinecor, 2014.
- [42] R. B. Gutiérrez, W. N. Núñez, S. C. Urrea, H. S. Osorio y N. D. Acosta, “Revisión de la seguridad en la implementación de servicios sobre IPv6”, INGE CUC, vol. 12, N.º 1, pp. 86-93, 2016.
- [43] J. A. R. Franco, Estado del Arte de la (In) Seguridad Voip., Revista Ingeniería, Matemáticas y Ciencias de la Información, vol. 1, N.º 1, 2014.
- [44] J. E. Molina, “Propuesta de segmentación con redes virtuales y priorización del ancho de banda con QoS para la mejora del rendimiento y seguridad de la red LAN en la Empresa Editora El Comercio Planta Norte”, Tesis de pregrado. Universidad Católica Santo Toribio de Mogrovejo, 2012.
- [45] X. Soengas, “Los vínculos entre la información y el poder en la sociedad actual”, Revista ICONO14, Revista científica de Comunicación y Tecnologías emergentes, vol. 7, N.º 1, pp. 292-322, 2012.
- [46] E. E. Veloz Villavicencio, “Plan de seguridad de la información para la red de datos de una empresa aseguradora”, Tesis Doctoral, Universidad de las Fuerzas Armadas ESPE, 2015.
- [47] A. Torres y D. Hernando, “Evaluación de seguridad a sistemas de información en cuanto a ataques maliciosos con base en normatividad, tendencias, impacto y técnicas vigentes para ambientes empresariales a nivel nacional”, Tesis de pregrado, Universidad de la Sabana, 2015.
- [48] J. E. Jaén Solórzano, “Diseño e implementación del control de acceso a la red Cisco Identify Services Engine (ISE)”, Tesis de maestría, Espol, 2016.
- [49] P. Arias y S. Marcelo, “Modelo de un proceso de la gestión del riesgo de la seguridad de la información en entidades gubernamentales”, Tesis Doctoral, Quito, 2015.
- [50] D. A. Guerrero, J. Fera, C. Patiño Luna, G. Mendoza y G. Alison, “Telefónica Móviles today, the concept of security plays a key role in the success of a project because their mismanagement may lead to the occurrence of fatal accidents, to their delay”, Universidad de Piura, Perú, 2015.





## Capítulo XI

# SISTEMAS DE SOPORTE A LAS DECISIONES ESPACIALES. CASO DE APLICACIÓN: MAPIFICACIÓN DE SERVICIOS ECOSISTÉMICOS EN LA GESTIÓN TERRITORIAL URBANA

*Juan Carlos Valdés Quintero - jcvaldes@elpoli.edu.co*

Profesor titular – Politécnico Colombiano Jaime Isaza Cadavid

*Juan Camilo Parra Toro - jcparra@elpoli.edu.co*

Profesor asistente – Politécnico Colombiano Jaime Isaza Cadavid

*John Fernando Escobar Martínez - jfescobar@elpoli.edu.co*

Profesor asistente – Universidad de Antioquia

## I. INTRODUCCIÓN

Las innovaciones tecnológicas franquean diversas etapas desde su aparición hasta que son integradas exitosamente en la cotidianidad de las sociedades, según [1], este proceso puede sintetizarse en:

**Creación:** la innovación surge en un centro de investigación, en grupos relativamente cerrados, en los cuales la experimentación, el ensayo y el error muestran los potencialidades y limitantes de los nuevos desarrollos.

**Contagio:** las experiencias muestran cómo las innovaciones pueden adaptarse para encontrar soluciones a necesidades y problemas existentes, y esto motiva su divulgación y enriquecimiento por otros grupos especializados.

**Coordinación:** las aplicaciones más prometedoras gradualmente ganan aceptación y son desarrolladas interdisciplinariamente. La coordinación ayuda a

incrementar el alto potencial de la innovación y disminuir los costos de aplicaciones futuras.

**Integración:** las innovaciones son aceptadas e integradas dentro de las tareas rutinarias de un grupo social e incluso por la sociedad misma.

Los Sistemas de Información Geográfica (SIG) se han desarrollado, en gran medida, como respuesta a la necesidad de representar y analizar los fenómenos del mundo real por parte de las diferentes ciencias y disciplinas que abordan el espacio geográfico como objeto de estudio, este desarrollo puede sintetizarse en cinco acciones que describen su desarrollo histórico, a saber:

**Representar:** constituye la primera etapa de desarrollo de los SIG en la cual su función principal se orientaba a la representación digital de la cartografía convencional. En esta etapa se avanza en las interfaces y utilidades asociadas al dibujo asistido por computador (CAD, por su sigla en inglés)

**Medir:** esta etapa se caracteriza por la integración de los modelos matemáticos que permiten obtener una buena analogía del mundo real siguiendo las reglas, estándares y modelos aportados por la cartografía, la geodesia, la geografía y la agrimensura.

**Entender:** representa un salto cualitativo en el análisis del espacio geográfico y de ello se derivan los desarrollos que intentan estudiar los elementos y fenómenos espaciales como “sistemas” que sobrepasan la simple agregación de capas de información.

**Modelar:** se aborda la integración de datos y modelos buscando obtener no solo su representación visual, sino también las dinámicas asociadas a la interacción espacial y temporal de los elementos y variables bajo estudio.

**Simular:** se concreta en la tendencia actual que busca obtener escenarios de predictibilidad sobre espacios geográficos acotados, sometidos a las dinámicas naturales y a las acciones antrópicas a través de la búsqueda de respuestas a la pregunta ¿Qué pasaría si...?

Este proceso guarda sincronía con los desarrollos informáticos, tanto en el ámbito del software como del hardware, ya que son ellos los que han permitido abordar problemas cada vez más complejos con la integración de más variables y mayores volúmenes de datos. Esta dinámica ha llevado a que en la actualidad se presente una serie de propuestas de aplicación de los SIG como herramientas de soporte a las decisiones, y a pesar de que existe una gran discusión acerca de si ellos son generadores, submodelos de los Sistemas de Soporte a las Decisiones Espaciales (SDSS) en sí mismos, igualmente existe un relativo consenso en la importancia del enfoque espacial en el estudio de diversos problemas y en las propuestas de solución a estos.

Es así como los SDSS se han constituido en un área de gran impulso en la investigación académica, y si bien se encuentran varias definiciones, en general puede decirse que estos sistemas son plataformas que se centran en el apoyo a la toma de decisiones en problemas específicos semiestructurados y no estructurados, en las cuales las bases de datos, las interfaces, los modelos y el usuario decisor son

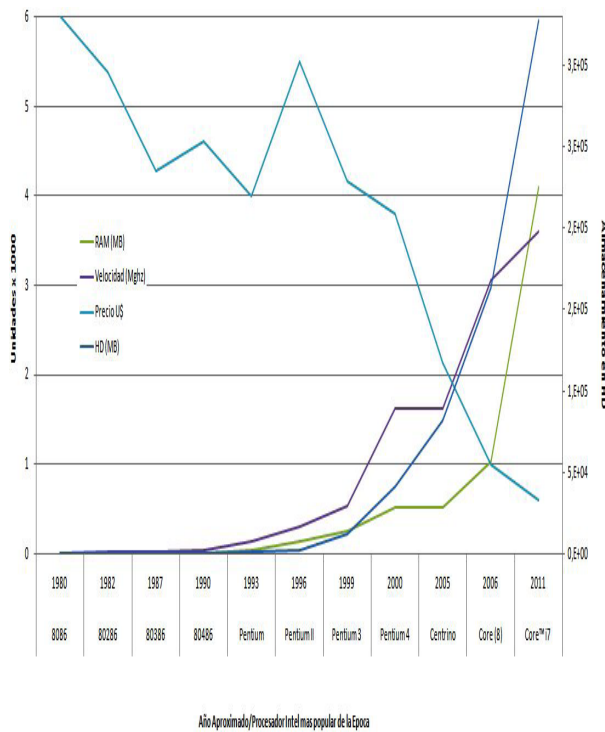
parte integral del proceso de toma de decisiones. Esta característica es de especial relevancia en la investigación de fenómenos naturales en los cuales los problemas tienden a ser semiestructurados, es decir, están constituidos por una serie de sub-problemas que son tratados, inicialmente, uno por uno a través de modelos que tienen un nivel de incertidumbre significativo donde solo se pueden evaluar e integrar algunas de sus variables, apenas se pueden precisar algunas relaciones y su solución es situacional, pero el espacio de decisión permanece abierto con una importante injerencia del decisor.

## II. ANTECEDENTES

La integración efectiva de las tecnologías geoinformáticas en diferentes campos de aplicación ha sido impulsada por el desarrollo y expansión de las tecnologías computacionales, las cuales han incrementado en forma exponencial la capacidad de almacenamiento y procesamiento, las capacidades de despliegue gráfico y el almacenamiento masivo en dispositivos portables y en servicios de la nube, esto a su vez ha producido un círculo virtuoso en el cual los costos del hardware disminuyen, al tiempo que aumenta la demanda de aplicaciones y software en general, que a su vez disminuye los costos e incrementa los servicios prestados.

A pesar de que puede ser un poco subjetivo, un rastreo de la publicidad de precios y características de los equipos de cómputo de escritorio más populares en diferentes años a partir de 1980, realizado en los archivos escaneados de google-books, oferta una visión de cómo el precio del hardware ha caído a cerca de una décima parte al tiempo que la velocidad se ha multiplicado por 29 y la memoria RAM y la capacidad de almacenamiento en disco se han multiplicado por un asombroso factor de 215 (ver Figura 1).

Esta evolución en precios y capacidades, y especialmente el enorme salto cualitativo y cuantitativo en las capacidades de almacenamiento, han impulsado fuertemente el uso de la informática en diversos contextos, y en este caso han propiciado el desarrollo de la geoinformática, la cual se entiende como la agregación de diversas áreas del conocimiento orientadas al estudio de problemas geográficos a través del uso de software y modelos de diversa índole.



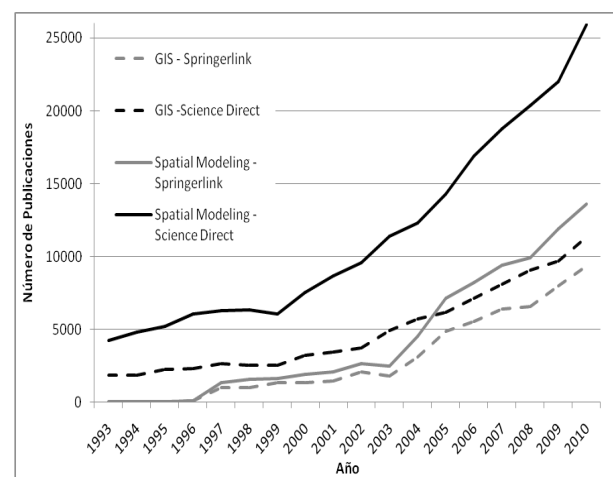
**Fig. 1.** Comportamiento del precio de los computadores de escritorio versus velocidad, memoria RAM y almacenamiento en disco desde 1980 a 2011.

En la actualidad, el establecimiento de la geoinformática como una disciplina madura ha propiciado el surgimiento de nuevas formas de ver y entender los elementos y procesos que se presentan en la naturaleza. Este hecho se soporta en la posibilidad que tienen los investigadores de unir datos y métodos con el objetivo de representar, analizar, entender y modelar fenómenos que son variables en el tiempo y en el espacio, haciendo uso de objetos espaciales explícitos y las capacidades de almacenamiento, integración, procesamiento y análisis que ofertan los SIG y otros instrumentos como los sensores remotos.

Son abundantes las referencias y tópicos de aplicación de la geoinformática y sus instrumentos de apoyo en el análisis de la naturaleza y los fenómenos que en ella se presentan. Estas aplicaciones han evolucionado desde los problemas relacionados con la representación cartográfica y la cartografía digital,

hasta el establecimiento de sistemas de soporte a las decisiones espaciales, pasando por una numerosa producción de modelos de interpretación, análisis y predicción de fenómenos que, según el campo de aplicación y las tecnologías utilizadas, han recibido diversos nombres como por ejemplo, geovisualización, geomodelación, y modelación espacial basada en Sistemas de Información Geográfica.

Si bien el concepto de modelación espacial se remonta a fechas anteriores a los SIG, éste comienza a popularizarse y expandirse en los albores de la década de los setenta, y sus principios rectores se expresan en el libro *Diseño con la Naturaleza*, escrito por el arquitecto paisajista Ian McHarg, en el cual se aboga por un diseño y planificación que tengan en cuenta los problemas medioambientales y sociales de tal forma que se asegure que los recursos se utilicen de manera adecuada y responsable, en la búsqueda de un futuro mejor para todos [2]. Posteriormente, con el surgimiento y expansión de los SIG, la modelación espacial adquiere una gran fuerza y difusión en diferentes campos del conocimiento, y esto se refleja en una evolución sincrónica del número de publicaciones en revistas especializadas. La Figura 2 muestra un comparativo de la producción en dos índices bibliográficos consultados para un periodo de 17 años con respecto a los Sistemas de Información Geográfica y la modelación espacial.



**Fig. 2.** Evolución de las publicaciones relacionadas con la modelación espacial y los SIG en los índices Springer y Science Direct.

En estas publicaciones especializadas, debe resaltarse la participación de los Journal Computers & Geosciences con la publicación de 588 artículos, que han crecido desde solo 16 artículos en 1991 hasta 110 en 2011, lo cual es un indicador del interés de los desarrolladores por abordar problemas complejos en el contexto espacial.

### III. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA COMO HERALDOS DE LOS SDSS

La integración de las TIC (Tecnologías de la Información y las Comunicaciones) en el tratamiento de problemas asociados a la gestión del territorio y los fenómenos naturales ha posibilitado el uso conjunto de las tecnologías de la información y las comunicaciones, las ciencias computacionales y los sistemas de modelación y soporte a las decisiones en la solución de problemas complejos del ambiente. Más específicamente, estas aplicaciones se refieren a los flujos de información a través de la cadena Datos – Modelos – Conocimiento – Decisiones – Impactos, los cuales incluyen la adquisición y el análisis de datos usando las TIC y los sensores remotos a través de modelos integrados; la optimización del conocimiento y participación de los actores en la gestión del fenómeno y/o el territorio; la difusión del conocimiento y la retroalimentación de los usuarios y grupos de interés mediante el uso de sistemas de comunicación como el Internet.

Aquí surgen dos de las primeras restricciones prácticas de los SDSS, a saber: la información necesaria para desarrollar un sistema y la definición del uso que se espera de él, ya que es imposible desarrollar un SDSS que cumpla con todos los propósitos y en muchos casos es difícil disponer de la información necesaria para un propósito específico.

El uso de los SIG como “generadores” de SDSS se debe a la capacidad de interacción que tienen estos softwares con técnicas refinadas de intercambio dinámico de datos (DDE, Dynamic Data Exchange), vinculación con objetos (OLE, Object Linking and Embedding) y la conectividad con bases de datos (ODBC, Open Database Connectivity).

Esta idea se basa en la estructuración de modelos computacionales orientados a objetos que permitan vincular los SIG y los softwares de análisis numérico basándose en el hecho que el mundo puede ser percibido como un conjunto de objetos que interactúan en diferentes formas [3], y que estas interacciones pueden ser representadas por medio de objetos espaciales y objetos temáticos.

Los objetos espaciales representan entidades del mundo real con sus respectivos atributos geográficos, físicos, ambientales y socioeconómicos, y los objetos temáticos constituyen métodos y temas de interés para los objetos espaciales, es decir que son funciones que describen, exploran o analizan las relaciones entre los objetos espaciales. Así, los modelos y funciones de los SIG pueden ser vistos como métodos aplicados a los temas, y la estrategia de integración de los modelos y las funciones de los SIG se convierte en una cuestión pragmática de elección de uno o más métodos que realizan una o más tareas sobre el o los objetos seleccionados [4].

En [5] se presentan una serie de trabajos desarrollados en años anteriores en los cuales, a pesar del estado incipiente de desarrollo de los SIG, se plantean las preguntas claves asociadas al desarrollo de aplicaciones SIG orientadas al conocimiento y la gestión del territorio y los fenómenos conexos. Estas preguntas son:

- A. ¿Cómo se representan los datos y las funciones asociados a la caracterización del fenómeno conservando las características espaciales y temáticas?
- B. ¿Cómo se integran la definición del problema, la manipulación de datos y las funciones de análisis?
- C. ¿En qué tareas las capacidades espaciales de los SIG se aplican a la formulación del modelo y el análisis?

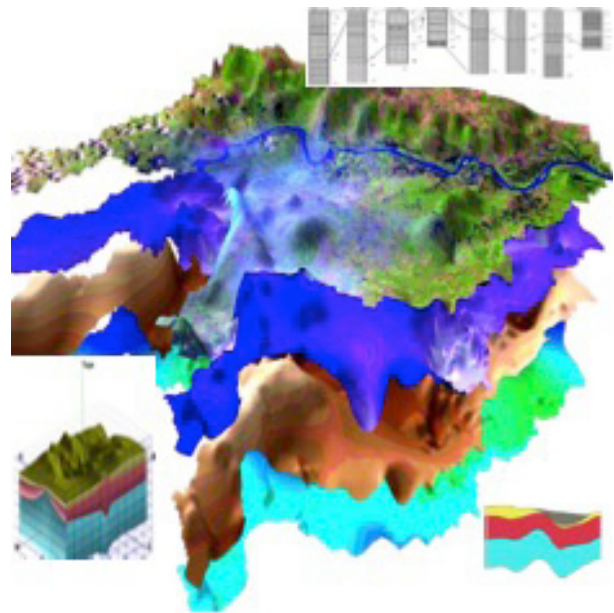
Estas preguntas pueden tener una respuesta parcial en el cumplimiento de los siguientes requerimientos generales desde la concepción y estructuración de los sistemas de soporte a las decisiones espaciales:

- Que se orienten a la solución de los semiestructurados y no estructurados que enfrentan los investigadores y gestores del territorio y sus recursos.
- Que combinen los modelos y las técnicas analíticas de las ciencias de la Tierra con las funciones tradicionales de acceso, recuperación y producción de datos y resultados pertinentes a la informática.
- Que se centren específicamente en las características que los hacen fáciles de utilizar, de un modo interactivo, por usuarios sin formación técnica profunda en todos los aspectos comprometidos en su desarrollo.
- Que enfatizen en la flexibilidad y adaptabilidad para acomodarse a los cambios requeridos en la toma de decisiones por parte del usuario.
- Que cuenten con los mecanismos que permitan la solución de problemas en forma interactiva y recurrente.

Estos propósitos tan generales pueden consolidarse a través de un esquema conceptual simple que da cuenta de las propiedades intrínsecas de la información espacial y que se basa en cuatro elementos básicos, a saber:

#### A. El Entorno

Es el “mundo real”, o la percepción que se tiene de él a través de los sentidos (por ejemplo, la topografía) y las herramientas que extienden la percepción humana (por ejemplo, los sensores remotos). Este tipo de información es bien descrita por los sistemas de información geográfica que contienen representaciones de superficies continuas y discretas bajo la forma de objetos espaciales, tales como puntos, líneas, superficies y volúmenes, cuya ubicación se puede considerar fija en el espacio en un momento dado del tiempo (ver Figura 3).



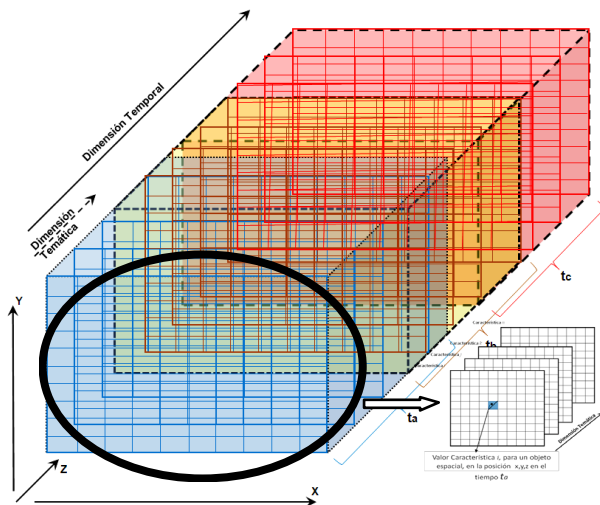
**Fig. 3.** Modelo espacial superficial y subterráneo de una región del Bajo Cauca Antioqueño (Colombia).  
Fuente: [6].

#### B. Los Geodatos

Se refiere al conjunto de mediciones obtenidas por sistemas de observación y redes de monitoreo, además de las propiedades distribuidas que pueden ser obtenidas a través de instrumentos diversos, por ejemplo, los sensores remotos. Dado que esta información puede ser muy voluminosa y compleja se requiere una base de datos relacional, un sistema de archivos estructurados y metadatos que permitan “operar” dicha información.

Habitualmente estas mediciones reflejan el comportamiento de una variable en un espacio de 4 dimensiones (las espaciales  $x$ ,  $y$ ,  $z$  y la temporal,  $t$ ) y la relación de dicha variable con las otras que ocurren en el mismo espacio (dimensión temática) (ver figura 4). En el establecimiento de estas relaciones surgen dos conceptos que pueden ser complejos de asimilar: el primero es la asignación semántica que se refiere al hecho de darles significado a las cosas, unido a la semántica cartográfica, que es el proceso por el cual se conectan “cosas” que tienen significados similares; el segundo es la asignación ontológica que se refiere al intercambio de información entre diferentes sistemas y entidades.





**Fig. 4.** Esquematización de un espacio n-dimensional a través de geodatos.

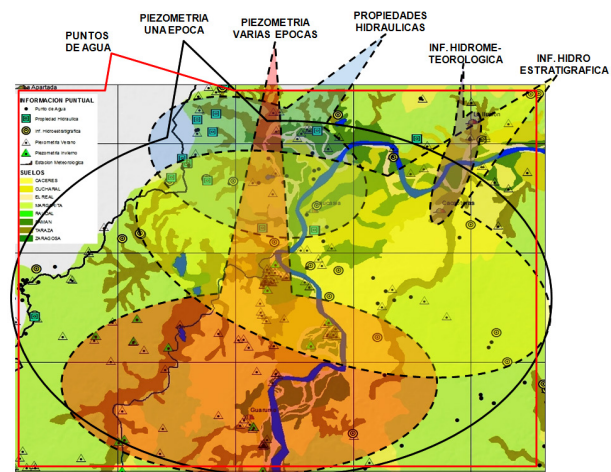
### C. El Espacio

Es la abstracción, generalmente simplificación, del espacio de análisis, que se centra en la hipótesis que se prueba. Por ejemplo, un territorio complejo puede reducirse a un pequeño número de unidades que se consideran homogéneas en algunas propiedades. A pesar de que la abstracción es generalmente subjetiva, los SIG han permitido la introducción de cierta objetividad en la asignación a través de los modelos espaciales robustos que reproducen las condiciones físicas o naturales.

Sin embargo, debe tenerse en cuenta que la simplificación de un sistema (espacio conceptual) difiere de los modelos que describen los fenómenos que en él ocurren (simulación espacial), y se constituyen dos grupos distintos de las hipótesis que, consecuentemente, deben evaluarse de manera independiente. Una contribución importante de los SDSS es que permiten construir y evaluar múltiples conceptualizaciones del mismo sistema.

### D. Simulación

La construcción y operación de SDSS incluyen la representación de los fenómenos a través de ecuaciones



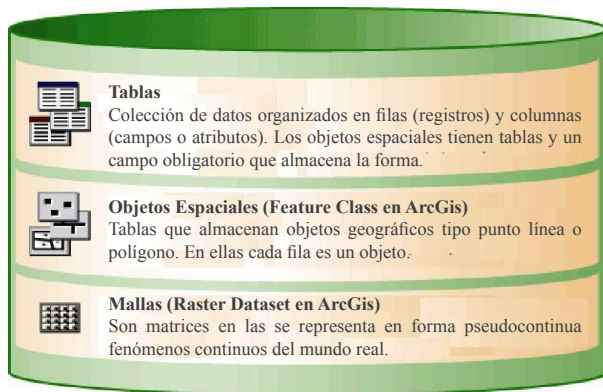
**Fig. 5.** Ejemplo de agrupación espacial y temática de conjuntos de datos espaciales puntuales, sobre una fuente distribuida. Fuente: [7].

o funciones que describen el funcionamiento de las unidades conceptuales. Sin embargo, debe entenderse que la simulación es solo un componente de un sistema de información, por lo tanto, la interacción entre los modelos de simulación y los modelos de datos espaciales es fundamental en la constitución de este tipo de sistemas.

Como puede intuirse, la consolidación de un SDSS, considerando todas las posibles dimensiones, puede ser compleja, y más aún si se intenta desarrollar una serie de asignaciones semánticas y ontológicas, ya que los conjuntos de datos se traslapan, interceptan, unen o distribuyen, tanto en el espacio geográfico como en las dimensiones temática y temporal.

Estas relaciones complejas encuentran su solución a través de una Base de Datos Espacial en la cual los datos requeridos, obtenidos como abstracciones y simplificaciones de la realidad, son almacenados como conjuntos de datos que se conectan a través de relaciones funcionales y espaciales. En general, cualquier sistema de información geográfica usa los tres tipos de datos básicos que se ilustran a continuación:





**Fig. 6.** Tipos de datos básicos en una base de datos geográfica, GDB (por su sigla en inglés). Fuente: ESRI, 2008. Traducido y adaptado por los autores.

## E. La Base de Datos Espacial

Una base de datos espacial, comúnmente llamada geodatabase, es el repositorio en el cual se almacenan los datos geoespaciales, los cuales describen la ubicación y características de los elementos que permiten relatar, interpretar y analizar el fenómeno bajo estudio. Las ubicaciones de las características espaciales se determinan por coordenadas. Las características de los elementos se expresan como atributos numéricos, temporales o temáticos.

Un dato geográfico posee tres componentes fundamentales que describen espacialmente a cualquier entidad. Estos son la ubicación en algún sistema de referencia, las características del objeto y las relaciones espaciales que posee con su entorno.

En un sistema de información geográfica los datos pueden ser almacenados usando dos estructuras básicas: los vectores y las matrices (también llamadas Raster o Grids). Los sistemas vectoriales son modelos en los cuales los objetos espaciales se representan por vectores que almacenan las coordenadas de los elementos como puntos-vértices. Esta estructura vectorial permite además la generación de las relaciones topológicas con el entorno. Estos elementos vectoriales pueden asumir, de acuerdo con el fenómeno que representan y la escala, las siguientes características:

**1) Puntos.** Representan aquellos objetos que a la escala de trabajo solo pueden ser representados por su posición  $x,y$ , y aun cuando no sean perceptibles en el mapa, son indispensables para representar un fenómeno.

**2) Líneas.** Son todos aquellos elementos constituidos por al menos dos puntos. Es así como la línea está constituida por todos los puntos que la forman (nodos) y los segmentos rectos que unen los nodos y así determinan su forma, e intrínsecamente asume una propiedad espacial, que es la longitud.

**3) Polígonos.** Son todos los conjuntos de puntos, donde el punto de inicio es igual al de cierre, formando así un objeto cerrado con un interior y un exterior. Poseen intrínsecamente las propiedades de área y perímetro.

Para cada uno de los objetos anteriormente mencionados, el sistema de información geográfica puede almacenar y asociar diferente información que lo califica en forma de tablas de atributos, que pueden relacionarse con los objetos espaciales usando un sistema de claves primarias y foráneas.

**4) Raster.** De otro lado, las estructuras RASTER consisten en la representación del mundo real por medio de una malla compuesta de celdas. Su creación responde a la necesidad de representar fenómenos continuos mediante un modelo pseudo-continuo a partir de datos obtenidos en forma discreta. La estructura simple de este tipo de datos, la capacidad de sobreposición, la posibilidad de generación a partir de modelos estocásticos y determinísticos, la afinidad con otros modelos basados en diferencias finitas y la evolución de los sensores remotos, han permitido que ella se convierta en uno de los pilares de la geomodelación.

## IV. NECESIDAD DE HERRAMIENTAS SDSS PARA LA GESTIÓN DEL RIESGO Y LA TOMA DE DECISIONES ASOCIADAS A CASOS DE INESTABILIDAD DE LADERAS

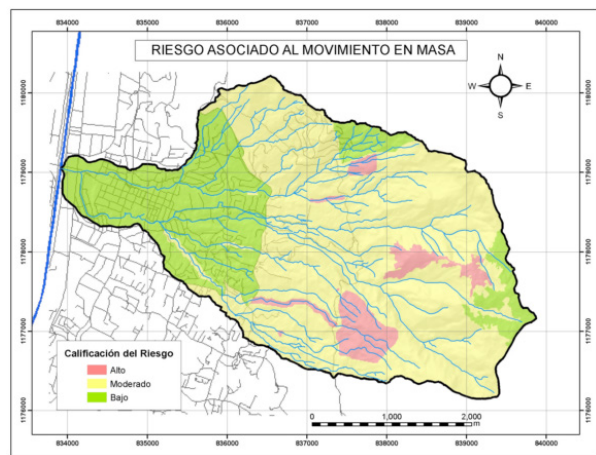
Las temporadas de lluvia en casi todo el territorio nacional causan una serie de eventos desafortunados a diferentes escalas espaciales y con niveles de

afectación cuyos costos tienen un amplio rango no sólo a nivel económico, sino también a nivel social y cultural. Basta mencionar sucesos como los diversos deslizamientos en la microcuenca La Presidenta y zonas aledañas, lo que ello ha implicado en términos económicos, pero sobre todo su implicación con la pérdida de vidas humanas.

Todo el escenario asociado a este tipo de eventos está relacionado con los riesgos hidrológicos y geotécnicos que comprenden, entre otras, la ocurrencia de eventos de inundación, sequías, vientos de gran magnitud y movimientos en masa por efectos de acumulación de agua en las laderas. La microcuenca de la quebrada La Presidenta atraviesa una comuna muy importante de la ciudad de Medellín, como lo es el sector de El Poblado, desde la parte alta en el corregimiento de Santa Elena, hasta su desembocadura en el río Medellín a la altura del Politécnico Colombiano Jaime Isaza Cadavid. En el corredor de este cauce se han presentado diversos problemas asociados al riesgo hidrológico y geotécnico, entre los cuales se destacan, por mencionar algunos, los movimientos en masa en la vía Las Palmas, los movimientos en masa en urbanizaciones residenciales de la Vía Cola del Zorro y las frecuentes salidas del cauce de la quebrada en la parte baja de la cuenca. Todos estos eventos generan pérdidas de vidas humanas, así como millonarios daños materiales, taponamiento parcial o total de vías, averías sobre obras de infraestructura aledañas, entre otros. Para Valdés, Parra y Herrera “los eventos hidrológicos extremos que allí se presenten pueden desencadenar inundaciones y movimientos en masa que afecten en gran medida obras de infraestructura y usuarios en particular” [8].

Aunque la ciudad cuenta actualmente con un sistema de alerta temprana conocido como SIATA (<http://www.siata.gov.co>), cuyo objetivo es “Alertar de manera oportuna a la comunidad sobre la posibilidad de ocurrencia de un evento que pueda causar daño ... mediante el monitoreo, el pronóstico, la generación de alertas y la actualización de los planes de emergencia” [9]. En dicho sistema existe un muy buen número de estaciones para la medición de diversas variables hidroclimatológicas, y se reconoce que su alcance inicial estuvo limitado a la generación de avisos al superar ciertos umbrales, que si bien son

necesarios para prevenir catástrofes, no acaban de satisfacer la necesidad de herramientas de mediano y largo plazo para el soporte de las decisiones de los gestores del riesgo a diferentes niveles. En la zona se han realizado modelaciones para estimar la evaluación del riesgo [10], a partir de la superposición de las vulnerabilidades ante cada amenaza y de la peligrosidad específica, y como resultado se tienen mapas de riesgo para cada amenaza, como se muestra en la Figura 7. Más allá de lo anterior, se identifica la necesidad de contar con herramientas para estimar la vulnerabilidad socio-ecológica frente a la pérdida de los ecosistemas presentes en la zona.



**Fig. 7.** Mapa de Unidades Superficiales y riesgo asociado a movimientos en masa de la microcuenca de la quebrada La Presidenta [11].

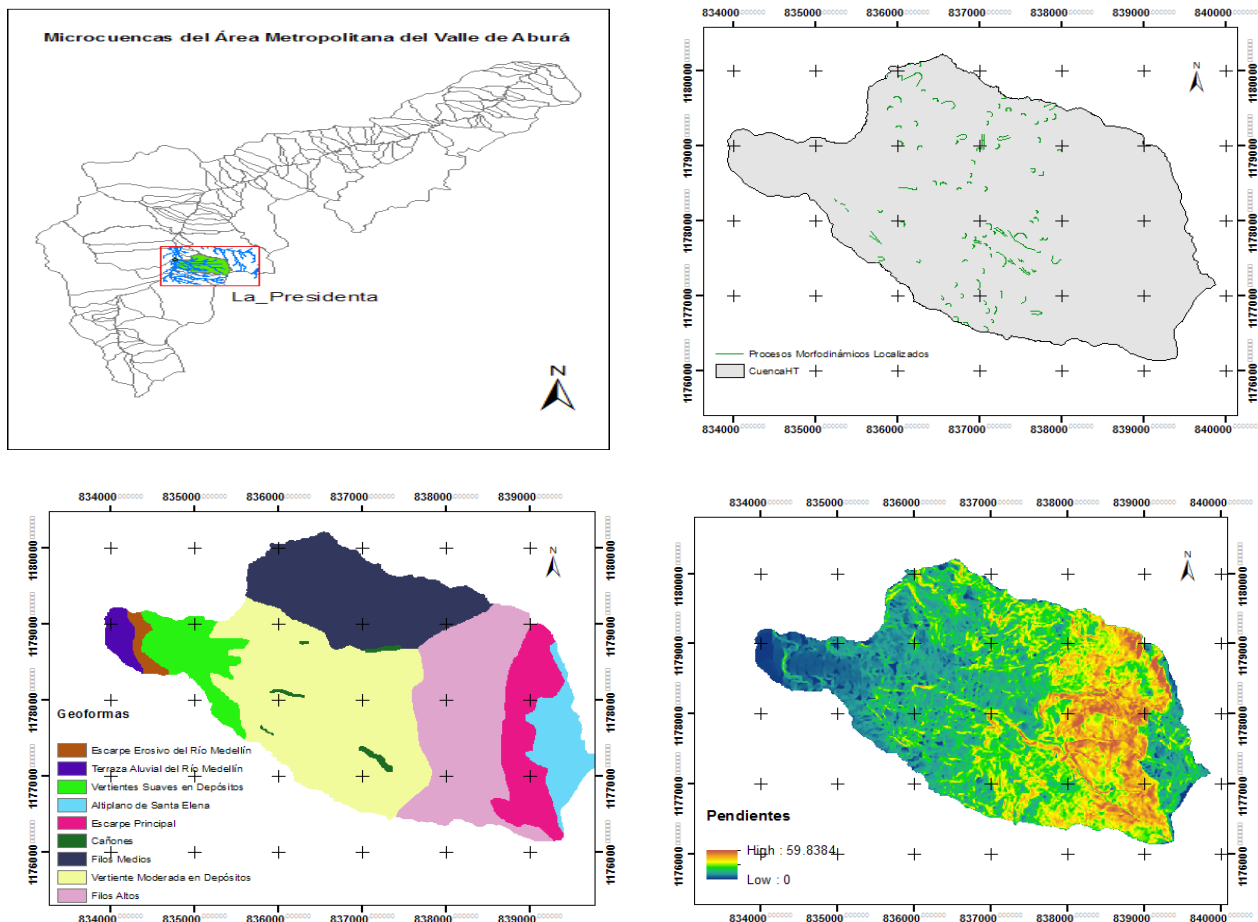
La microcuenca de la quebrada La Presidenta constituye un área de interés local para la ciudad, dado que allí se vienen desarrollando importantes proyectos de expansión urbana. La quebrada atraviesa grandes vías de comunicación como Las Palmas, las transversales Superior e Inferior, la Avenida 34, la Avenida El Poblado, la Avenida Las Vegas y la Avenida Regional; afecta un gran número de pobladores de diversos estratos sociales; está en una de las zonas de mayor influencia comercial de la ciudad y los posibles riesgos hidrológicos que allí se corren pueden afectar un sinnúmero de obras de infraestructura y a sus usuarios. Esto sin tener en cuenta que es una de las zonas donde el suelo tiene un alto valor comercial en términos de finca raíz.

Además de lo anterior, se escoge este caso de estudio debido a que en esta área se adelantan proyectos para la instrumentación que permitan establecer una red de monitoreo de variables hidrológicas y geotécnicas [5], en donde se podrían obtener a futuro datos e información que posibiliten alimentar los modelos en aras de definir zonas de retiro en los cauces, evolución de la oferta y demanda hídricas de la cuenca, y así atender uno de los faltantes fundamentales expuestos en el Plan de Ordenación y Manejo de la Cuenca del Río Aburrá [10]. Además, se cuenta con algunas intervenciones urbanísticas que son proyectos concebidos como una forma de protección y conservación de algunas funciones ecosistémicas.

### A. Caso de Estudio. Cuenca La Presidenta

La microcuenca de la quebrada La Presidenta tiene un área superficial aproximada de 15 km<sup>2</sup>, longitud del cauce principal de 7,17 km, pendiente promedio de 17,68 % y pendiente del cauce de 18,33%. Nace en la vereda El Plan del corregimiento de Santa Elena, aproximadamente en la cota 2.700 y desemboca en el río Medellín en la cota 1.488 m.

**1) Geología e hidroclimatología.** El área de estudio seleccionada para el desarrollo de este trabajo es la microcuenca La Presidenta, ubicada en el suroriente de la cuenca del valle del Aburrá (ver Figura 8). Esta cuenca presenta usos mixtos del suelo en diferentes ambientes urbanos y rurales en un área aproximada de mil quinientas hectáreas.



**Fig. 8.** Ubicación de la microcuenca La Presidenta en el valle de Aburrá y distribución de pendientes, composición de geoformas y procesos morfodinámicos localizados.

La parte superior de la cuenca presenta pendientes suaves y se encuentra ubicada en el altiplano del valle de Santa Elena. Posteriormente la pendiente sufre un aumento abrupto en el escarpe principal y constituye una zona apta para desprendimiento de material, se descuelga en fillos altos sobre la ladera oriental del valle de Aburrá y se normaliza en la parte media del trayecto en fillos medios lo que permite la formación de depósitos no consolidados producto del transporte de sedimentos, y por último se suaviza drásticamente sobre la vega del río Aburrá (el cual recibe el nombre de río Medellín, a su paso por la ciudad), donde se presenta el resto de depósitos identificados en su geomorfología.

Los depósitos hallados en los fillos altos y en la vertiente moderada de la cuenca dan lugar a procesos morfodinámicos puntuales y activos. El río Aburrá constituye un control hidráulico muy importante para el cauce principal de la quebrada La Presidenta, al generar algunos procesos de socavación regresiva en su desembocadura, así como dificultad para disipación de energía de la quebrada en el tramo de entrega de aguas al río. Gran parte de su cauce se encuentra intervenido mediante recubrimientos en el fondo y paredes laterales, y así mismo rectificado en ciertos lugares, lo cual ha cambiado las condiciones de disipación de energía naturales de la corriente. El transporte de depósitos en la parte media y baja de la cuenca da como resultado, bajo condiciones de lluvia cada vez más recurrentes, la salida de la corriente de su lecho confinado, en parte motivada por la urbanización producto de la presión antrópica sobre el territorio, la quebrada intentan ocupar la llanura de inundación que le corresponde, inunda sus planos laterales, genera socavación de estructuras civiles presentes, y deposita gran cantidad de sedimentos y material vegetal.

Según [12] los depósitos aluvio torrenciales (Qat) están constituidos por una gran cantidad de bloques rocosos con tamaños hasta métricos, mezclados en una matriz de textura limo arenosa, húmeda y

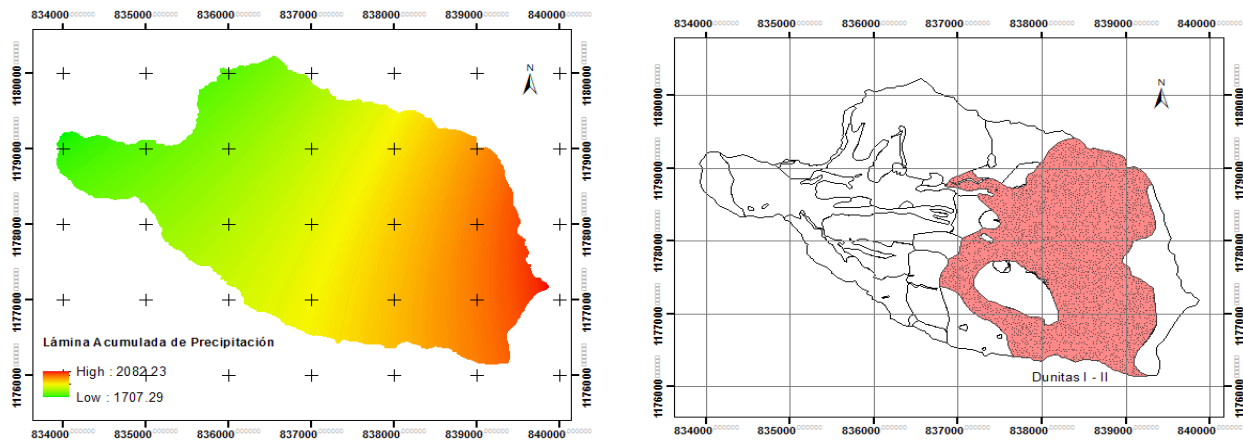
suelta. El espesor es variable y va desde menos de 1 metro hasta más de 20 metros, dependiendo de la topografía donde se depositan. Afloran en la margen izquierda del río Medellín, hacia las quebradas Ana Díaz, Altavista, Guayabala, Iguaná y La Gómez; y en la margen derecha hacia las quebradas: La Presidenta, La Volcana, La Poblada y Santa Elena.

La mayor parte de la precipitación sobre la cuenca ocurre sobre su parte alta, ubicada en el valle de Santa Elena. La unidad geológica predominante allí es la dunita altamente meteorizada, principalmente de grado V y VI [13], aunque con presencia de roca fresca en algunos casos, producto del afloramiento en el Oriente y Norte de Medellín de un cuerpo de forma alargada que hace parte de un fragmento de corteza oceánica desmembrada y que compone el Complejo Ofiolítico de Aburrá [14].

Según [15] la deformación, metamorfismo y transporte, inferido de las relaciones cinemáticas, no sólo de las dunitas y de las anfibolitas, sino también de los ortogneises augen (del alemán: ojos de cristales muy gruesos) a los cuales ellas están asociadas, apuntan a una historia geológica común a todo este conjunto de unidades litoestratigráficas. La Figura 9 ilustra la ubicación de dicha unidad geológica.

En el planteamiento del Plan de Ordenación y Manejo de la Microcuenca se describe la presencia de dunita como sigue [13]: “esta unidad aflora en el flanco sur “oriental del Valle de Aburrá, buenos afloramientos de roca fresca (grado I y II) se encuentran sobre la vía Las Palmas cerca al Instituto Musical Diego Echavarría”; y ya en el alto de Santa Elena “en las parcelaciones Cinturón Verde, El Monte; en las quebradas La Sanín, Escopetería, Escopetera, Sagrado Corazón, Chambul, La Marucha, La Chacona, La Poblada y La Presidenta. En algunos casos el cuerpo rocoso representa quiebres importantes en los cauces y alcanzan hasta 12 m de desnivel”.





**Fig. 9.** Distribución de la lámina anual de precipitación en la microcuenca La Presidenta y presencia de dunitas

A partir de lo citado en dicho texto, y proveniente de [16], se puede esperar un alto grado de infiltración puntual en la parte alta de la cuenca, sobre el Altiplano de Santa Elena, e igualmente inestabilidad en la parte media de la cuenca por el afloramiento de aguas infiltradas en la parte alta, ya que “es común encontrar el cuerpo diaclasado y con alto grado de fracturación, lo que en términos generales implica una inestabilidad moderada, con ocurrencia de deslizamientos puntuales y meteorización debido a filtración de aguas superficiales y por cortes en las laderas” [16].

Como se relacionó anteriormente, en la zona se identifica un clima subtropical templado y húmedo, con una temperatura promedio de 22 °C. La distribución espacial de la figura 3 muestra que las mayores precipitaciones se presentan en la parte alta de la microcuenca y las menores en la zona más plana variando entre 1.573 y 2.355 mm/año, con una precipitación promedio de 1.886 mm/año [10]. La evapotranspiración se encuentra en el rango de los 668 a los 1.032 mm/año [10]. Los caudales medios anuales de escorrentía superficial directa se establecieron entre las franjas de 0 a 0,978 m<sup>3</sup>/s [10]. A partir del modelamiento hidráulico de la red de drenaje [14] se determinó que la zona baja de la cuenca, con modelación de períodos de retorno de 100 y 500 años, presenta zonas de inundación a partir de problemas de insuficiencia hidráulica de los canales, causados por la presencia de un número importante de obras de cruce de vía (en su mayoría boxculverts) que causan fuertes controles hacia aguas arriba. A

lo anterior se suman las diferentes canalizaciones, rectificaciones de los cauces e invasión de las zonas próximas a las corrientes y los procesos urbanísticos que han restringido las corrientes que ocasionan la pérdida de zonas naturales de amortiguamiento de creciente en diferentes puntos de la red.

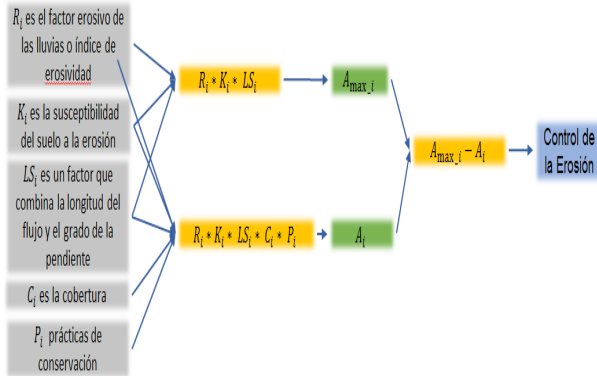
**2) Problemáticas sobre la gestión del riesgo y del territorio.** Los conflictos generados conforme sucede el crecimiento y desarrollo del territorio objeto de estudio pueden definirse, principalmente (aunque no de manera única), a partir del manejo del agua y el suelo, la presión sobre los ecosistemas y los flujos altamente incontrolados, tanto en lo demográfico como en la económico.

Así, pues, las obras hidráulicas realizadas responden a la solución de problemas puntuales y urgentes, y sus implicaciones no son tenidas en cuenta en obras ya existentes o en canales naturales adyacentes. El aumento de las superficies urbanizadas y pavimentadas hace que se genere una menor infiltración y una mayor escorrentía directa, lo que implica un aumento en el pico de los hidrogramas y un acortamiento en el tiempo de la base de éstos, lo que significa un tránsito de la creciente por los canales de las corrientes de agua que no corresponde al considerado en los diseños de muchas obras presentes en la cuenca. Muchos de los diseños de las obras hidráulicas existentes se realizaron con caudales máximos calculados desde condiciones de cobertura vegetal y usos del suelo muy diferentes a los actuales, por razón de la alta actividad antrópica



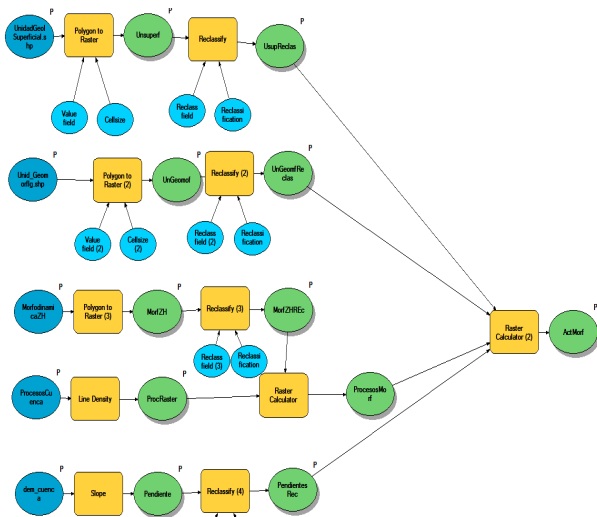


En concordancia con lo anterior, se plantea el siguiente modelo (Figura 11) para la mapificación de la FE Control de la Erosión:



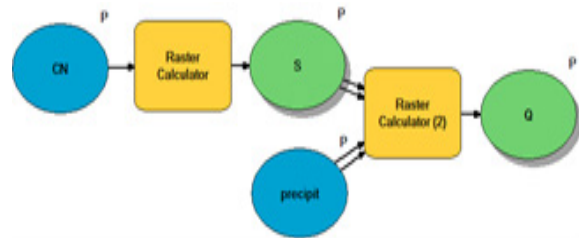
**Fig. 11.** Modelo para la mapificación de Función Control de Erosión en la Cuenca La Presidenta

Siguiendo la misma metodología, se planteó el siguiente modelo para recrear la función ecosistémica Actividad Morfodinámica (Figura 12):



**Fig. 12.** Modelo para la mapificación de la Función Actividad Morfodinámica en la cuenca La Presidenta.

De igual forma, para la Función Ecosistémica Escorrentía se adoptó el siguiente modelo (Figura 13):



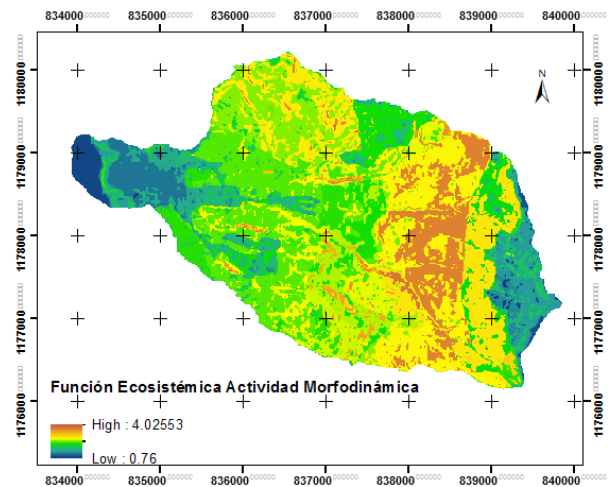
**Fig. 13.** Modelo para la mapificación de Función Escorrentía en la cuenca La Presidenta.

Según [17] “La provisión relativa de servicios ecosistémicos se obtiene a partir de la integración de los mapas de funciones ecosistémicas generados en el paso anterior. Como los mapas de funciones tendrán unidades diferentes (p. ej. TonC, kg de N/litro, etc.) deberán llevarse a una escala relativa (0-100) para su integración”. Esta transformación de cada FE a una escala relativa asegura la coherencia matemática de las variables involucradas en la expresión del SE.

**C) Resultados y Discusión**

A través de la utilización del Model Builder de ArcGIS y aplicando el modelo expuesto en la Figura 4, se logra la mapificación de SE de Estabilidad de Laderas en la microcuenca La Presidenta al relacionar las FE Actividad Morfodinámica, Control de Erosión y Escorrentía.

La Función Ecosistémica Actividad Morfodinámica muestra la susceptibilidad del terreno a presentar procesos morfodinámicos como movimientos en masa, o procesos de erosión concentrada y avanzada como cárcavas. Dicha función está condicionada por unidades geológicas superficiales, geomorfología (unidades geomorfológicas), procesos morfodinámicos y pendientes. En este sentido, los valores más altos de la función representan las zonas de la microcuenca más susceptibles a la combinación de los aspectos mencionados. Dichos valores se presentan en la geofoma de fillos altos donde la pendiente del terreno es mayor (Figura 14). Solo al final de esta franja comienzan a identificarse procesos morfodinámicos puntuales o localizados.



**Fig. 14.** FE Actividad Morfodinámica.

El mapa de la FE Control de Erosión (Figura 15), que presenta la diferencia entre la pérdida media anual de sedimentos por erosión para suelo desnudo y la pérdida de sedimentos en las condiciones reales de cobertura y prácticas de conservación del suelo, identifica zonas de la microcuenca La Presidenta donde el control de erosión se torna en un aspecto crítico y, en consecuencia, el servicio ecosistémico de estabilidad de laderas cobra especial relevancia. Tal es el caso del escarpe principal en la parte alta de la cuenca y en el desprendimiento de los filos altos y su contacto con el inicio de la vertiente moderada. En este último caso, el mapa de FE coincide con la presencia de procesos morfodinámicos localizados.

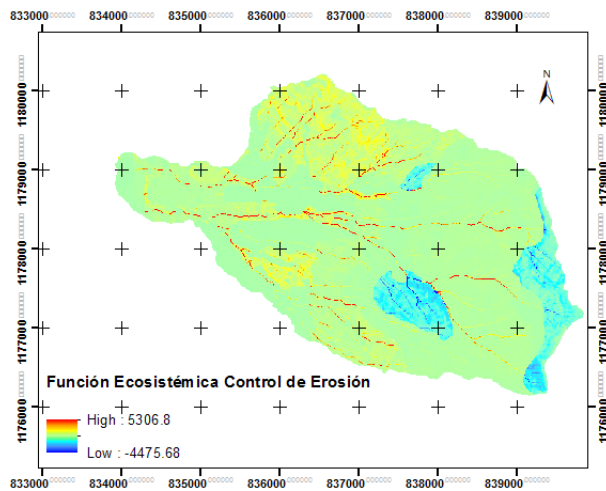


Fig. 15. FE Control de Erosión.

La FE Escorrentía relaciona los usos del suelo con la favorabilidad o desfavorabilidad en la estabilidad de una ladera, mediante el número de curva, las abstracciones máximas del terreno y la lámina de precipitación acumulada anual, lo que genera como resultado la lluvia que efectivamente produce escorrentía (lluvia neta). El mapa de la función permite identificar las zonas más altas de la microcuenca (geoformas correspondientes a Altiplano de Santa Elena, escarpe principal y filos altos) como aquellas donde se genera la mayor parte de la escorrentía que transcurre por la red hídrica hasta pasar por el punto de drenaje en el río Medellín. Esta escorrentía oficia como medio de transporte de los sedimentos producto de los procesos morfodinámicos activos en la parte media de la microcuenca, así como de aquellos que naturalmente se desprenden de las zonas de aporte, y que finalmente se depositan en la terraza aluvial del río (Figura 16).

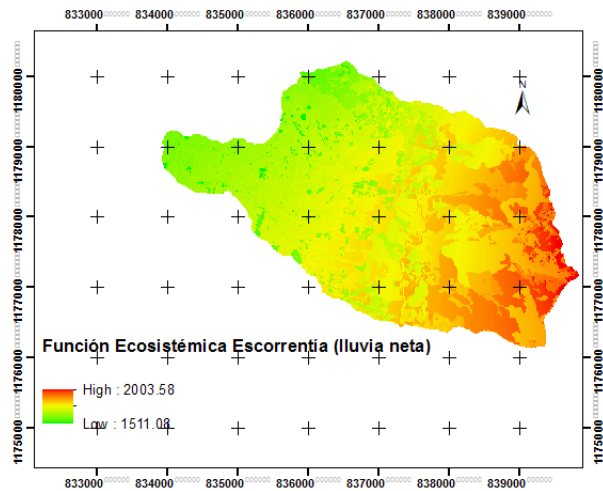


Fig. 16. FE Escorrentía.

Para evaluar el servicio ecosistémico Estabilidad de laderas (EL), se utiliza la matriz de Saaty para obtener los pesos de las funciones en el servicio a partir de matrices de opinión experta, y se obtiene la siguiente ecuación:

$$EL = 0,35 \times \text{Control de erosión} + 0,49 \times \text{Actividad Morfodinámica} + 0,16 \times \text{Escorrentía}$$

La mapificación del servicio (Figura 17) permite identificar zonas de provisión en la parte alta de la microcuenca (escarpe principal y filos altos). Ya en la zona media de la microcuenca, donde la susceptibilidad a la ocurrencia de movimientos en masa es mayor, el servicio ecosistémico no presenta valores de provisión tan elevados, pero por la misma susceptibilidad mencionada, y además por el hecho evidenciable de la materialización del riesgo geotécnico allí presente, es en esta zona donde el servicio ecosistémico cobra especial relevancia.

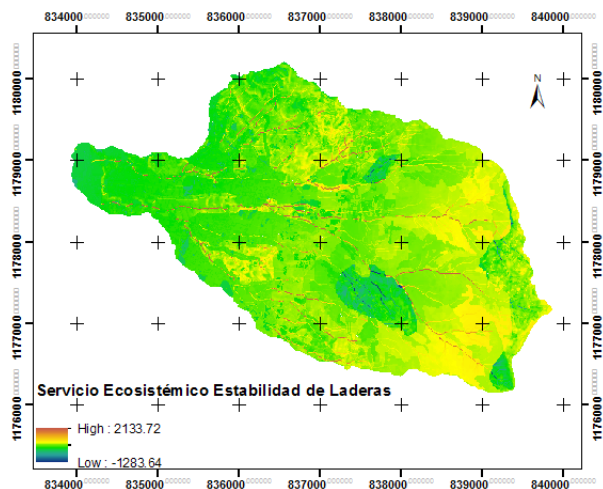


Fig. 17. SE Estabilidad de Laderas.

La provisión del servicio ecosistémico en el área de estudio evidencia una relación entre la cobertura vegetal en la zona rural de la microcuenca y en algunos sectores urbanos en donde los procesos de control de erosión y erodabilidad, los índices de erosividad y la implementación de obras hidráulicas, permiten obtener beneficios tangibles para la población. De igual manera, se realizaron modelaciones orientadas hacia los servicios de regulación climática y regulación hidrológica, obteniendo resultados coherentes con las variables involucradas en cada uno de ellos.

Uno de los grandes inconvenientes al momento de poder ejecutar los modelos es la ausencia de datos en el área de estudio relacionados con servicios, los cuales deberán ser obtenidos con información secundaria a partir de coberturas vegetales, pues la ausencia de instrumentación en la zona no proporciona la adquisición de datos primarios.

Trabajos como el que se está presentando poseen gran impacto en la gestión del territorio en localidades, pues a partir de los resultados obtenidos se podrán implementar metodologías para la valoración de servicios y, en tal sentido, determinar áreas de protección y conservación más coherentes, y su inclusión en la EEP realmente garantizará la oferta natural del territorio, razón por la cual su proyección hacia todas las microcuencas que componen el valle de Aburrá se hace fundamental para los planes de ordenamiento territorial del área metropolitana.

Resulta evidente que la gestión de los diferentes tipos de riesgos naturales presentes sobre un territorio (hidrológico, hidráulico, geotécnico) requiere la adopción de herramientas que permitan caracterizar sitios fuente de beneficios para las comunidades presentes, de manera que la inversión de recursos para la preservación del medio natural se haga de manera eficiente, y que de forma paralela se alcance un equilibrio entre el costo de la materialización del riesgo y el beneficio generado por la disponibilidad y el cuidado adecuado de los recursos presentes en los ecosistemas.

Más aún, en el largo plazo este tipo de modelos permiten soportar decisiones complejas encaminadas al logro del control sobre el crecimiento territorial, y

en el fondo, hacia la búsqueda de la sostenibilidad en el manejo y la explotación de los recursos naturales. De manera más inmediata, las funciones y servicios ecosistémicos pueden encaminar la construcción de las políticas de planificación que se traduzcan urgentemente en planes de intervención para sitios localizados, donde se presentan problemáticas ambientales y sociales puntuales y bien definidas, como es el caso abordado en el presente documento.

## REFERENCIAS

- [1] Curran Ch., “How Should CIOs Lead Technology Innovation?”, Boletín electrónico Optimize, Issue 46. 2006 [En línea]. Disponible en: <http://www.optimize.com/disciplines/strategic-innovation/showarticle.jhtml?articleid=167600561>).
- [2] J. Dangermond, “GIS, Designing our Future”, ArcNews, 2009. ESRI. [En línea]. Disponible en: <http://www.esri.com/news/arcnews/summer09articles/gis-designing-our-future.html>
- [3] P. Crosbie, “Object-oriented design of GIS: a new approach to environmental modeling”, en GIS and Environmental Modeling: Progress and Research Issues. GIS World Book, M. F. Goodchild et al. (Eds.), Fort Collins, 1996, pp. 383-386.
- [4] K. Fedra, “Distributed models and embedded GIS: integration strategies and case studies”, en GIS and Environmental Modeling: Progress and Research Issues. GIS World Book, M. F. Goodchild et al. (Eds.), Fort Collins, 1996, pp. 414-417.
- [5] D. McKinney y X. Cai X., “Linking GIS and water resources management models: an object-oriented method”, Environmental Modelling & Software, N.º 17, pp. 413-425, 2002.
- [6] T. Betancur, P. Palacio y F. Escobar, “Técnicas Geoinformáticas en la Modelación de sistemas acuíferos”, en Memorias V Seminario Hispanoamericano Sobre Temas Actuales de la Hidrología Subterránea. Salta, Argentina; 2011.
- [7] J. Escobar y T. Betancur, “La geomodelación en la exploración del recurso hídrico subterráneo” en Memorias Congreso Colombiano de Ingenierías Verdes CCIV, Medellín: Editorial Universidad de San Buenaventura. 2011, pp. 37-42
- [8] J. Valdés, J. Parra, S. Herrera, “Modelación Espacial para la Ubicación de Estaciones Meteorológicas

- en Cuencas Hidrográficas”, Revista Politécnica, Vol. 9, N.º 17, dic. 2013
- [9] Sistema de Alerta Temprana de Medellín — SIATA—, 2011. Disponible en: [www.siata.gov.co](http://www.siata.gov.co). [Accedido: 25 de febrero de 2011].
- [10] M. Área, Cornare, Corantioquia, y U. Nal, “Plan de Ordenación y Manejo de la Cuenca Hidrográfica del río Bogotá”, N.º 652, p. 104, 2006.
- [11] M. del V. de A. Área, CTA, and S. M. Ambiente, “Formulación del Plan de Manejo de la Microcuenca de la Quebrada La Presidenta. Caracterización Física”, N.º 4800000905, p. 75, 2005.
- [12] Área Metropolitana del Valle de Aburrá, “Formulación del plan de manejo de la microcuenca de la quebrada La Presidenta, zona suroriental de Medellín. Medellín: Área Metropolitana del Valle de Aburrá, 2007.
- [13] M. L. T. Cossio et al., “ECOSER - Protocolo colaborativo de evaluación y mapeo de servicios ecosistémicos y vulnerabilidad socio-ecológica para el ordenamiento territorial”, N.º 2, p. 58, 2015.
- [14] B. C. Sherrouse, J. M. Clement, and D. J. Semmens, “A GIS application for assessing, mapping, and quantifying the social values of ecosystem services”, *Appl. Geogr.*, vol. 31, N.º 2, pp. 748-760, 2011.
- [15] F. Villa et al., “A methodology for adaptable and robust ecosystem services assessment”, *PLoS ONE*, vol. 9, N.º 3, 2014.
- [16] D. Codato y B. Locatelli, “Evaluación y Mapeo de Servicios Ecosistémicos. Herramientas y Aplicaciones”, *Vidasilvestre.Org.Uy*, p. 20, 2012.
- [17] M. P. Barral, “Provisión de servicios ecosistémicos en paisajes rurales: Desarrollo de criterios y herramientas para el ordenamiento territorial rural,” Tesis Doctoral, p. 149, 2014.
- [18] K. G. Renard, G. R. Foster, G. A. Weesies y J. P. Porter, “RUSLE: Revised universal soil loss equation”, *Journal of Soil and Water Conservation*, vol. 46, pp. 30-33, 1991.

## Capítulo XII

# MODELO DE MONITOREO Y MEJORA DE PROCESOS PARA PYMES DEL SECTOR JURÍDICO

*Juan Camilo Giraldo Mejía - Jgiraldo1@tdea.edu.co*

Docente, Tecnológico de Antioquia

*Mauricio Amariles Camacho - Mauricio.amariles@usbmed.edu.co*

Docente, Tecnológico de Antioquia

*Fabio Alberto Vargas Agudelo - fvargas@tdea.edu.co*

Docente, Tecnológico de Antioquia

*Darío Enrique Soto Durán - dsoto@tdea.edu.co*

Docente, Tecnológico de Antioquia

*Claudia Elena Durango Vanegas - Claudia.durango@usbmed.edu.co*

Docente, Universidad de San Buenaventura

### I. INTRODUCCIÓN

Hoy en día las organizaciones buscan herramientas y métodos que les ayuden a mantener su competitividad. Un enfoque clave para que las organizaciones logren llegar a ser competitivas es tener una exitosa implementación de mejora en sus procesos [1], y para conseguirlo es necesario seguir un enfoque orientado a gestión por procesos de negocio, que es un aspecto relevante para las organizaciones. La revisión de aspectos tecnológicos modernos, como la gestión de procesos de negocio (Business Process Management, BPM) orientada a la distribución de las actividades de los procesos e inclusión de aspectos sociales relacionados con su ejecución, contribuye a mejorar los resultados en las etapas de despliegue, ejecución y monitorización de las actividades que se desarrollan en cada uno de sus procesos de negocio. Lo anterior se refleja en un modelado de procesos [2].

El modelado de procesos de negocios es un paso importante para apoyar el desarrollo de las actividades que se ejecutan en cada proceso. Esto involucra la revisión de los procesos del negocio y la descripción de cada uno de ellos, con el fin de fortalecer y estructurar

las actividades necesarias para la correcta ejecución de este proceso [3]. El modelado de procesos del negocio de las organizaciones refleja también el estado del flujo de dichos procesos. Un modelo bien definido incide en el éxito de las operaciones del negocio y ayuda a mejorar su eficacia, coherencia y transparencia. Esto es posible con la aplicación de la metodología BPM, la cual permite el modelado de procesos de negocio. BPM tiene como objetivo una mejor comprensión de los procesos, de manera que los decisores logren los objetivos estratégicos de la empresa [4]. Es una medida para evaluar cómo una organización gestiona sus procesos de negocio. La madurez lleva a un mejor rendimiento de los procesos y por ende de la organización. BPM contribuye para mejorar el desempeño y obtener un alto nivel de madurez [5].

Los modelos de procesos de negocio son un medio importante para diseñar, analizar, implementar y controlar procesos de negocios. Un modelo de procesos de negocio tiene que cumplir con ciertos requisitos de calidad sintáctica, semántica y pragmática de valor [6]. Desempeñan un papel importante en el análisis y la mejora del rendimiento de la organización.



Evidentemente, la calidad de un modelo de proceso de negocio tiene un efecto directo sobre el rendimiento del negocio, lo que ha motivado a proponer métodos y modelos adecuados para crear modelos de procesos de negocio que constantemente garanticen la calidad de la organización [7].

Un punto importante para garantizar la efectividad de los procesos es la simetría que debe existir entre las reglas y los objetivos del negocio. Cuando esto no se da, se presenta una disparidad, se produce un modelo separado que no es práctico ni aceptable y afecta los sistemas de información. Para solucionar esta situación debe hacerse una mejora continua de los procesos, y para resolver este problema, en [8] se propone considerar la variabilidad.

La Variabilidad servirá como una técnica de mejora de procesos de negocios para diseñar y ejecutar de manera eficiente una variable de proceso de negocios a lo largo de eventos diferentes que son similares entre sí en algunos aspectos pero se diferencian en otros [8].

Por lo anterior es necesario contar con un método que indique a través de fases o etapas cómo revisar, intervenir, y optimizar los procesos del negocio.

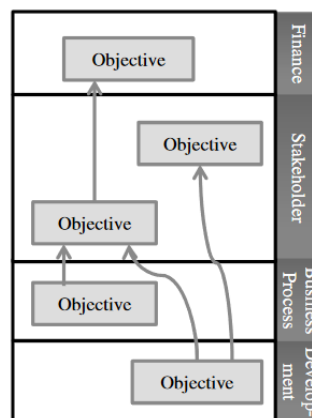
El artículo se estructura en cinco fases, mediante las cuales se muestra cómo se lograron los objetivos de la investigación.

En la fase 1 se presenta la revisión de literatura, con la cual se contextualiza sobre otros trabajos que tienen relación con el objeto en estudio, además se realiza una presentación de conceptos relacionados con la gestión y optimización de procesos de negocio. La fase 2 muestra la metodología que se siguió para alcanzar el método propuesto, soportado en las etapas de la gestión de procesos del negocio BPM. La fase 3 presenta los resultados de la investigación, correspondientes al modelo propuesto, la indicación y caracterización de sus componentes. La fase 4 muestra un caso de estudio, en el cual se valida el método propuesto. El caso de estudio corresponde al proceso de una organización del sector jurídico. Finalmente, el artículo muestra conclusiones derivadas de la aplicación del método sobre el proceso de la organización, a partir de lo cual se propone trabajo futuro.

## II. REVISIÓN DE LITERATURA

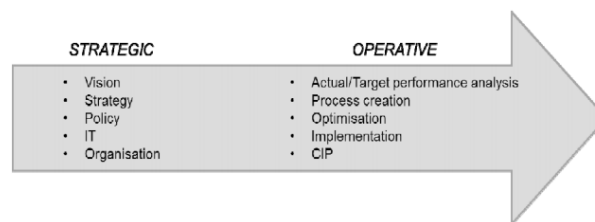
### A. Estrategias de planificación orientadas a inteligencia de negocios

En [9] se presenta una estrategia en el modelado del negocio que tiene los siguientes pasos: en primer lugar, la estrategia debería estar debidamente documentada, en el siguiente paso el proceso de modelado es refinado usando BPMN, finalmente, se propone una matriz de estrategias combinadas con el mapa de estrategia. La Figura 1 representa el mapa de estrategias por objetivos.



**Fig. 1.** Estructura de un mapa de estrategias BPM. Fuente: Adaptado de [9].

En [10] exponen una solución para las estrategias de planeación en pequeñas y medianas empresas Pymes. Es importante definir unas estrategias de acuerdo con las políticas de la organización y el contexto de TI. En la Figura 2 se muestra el modelo de actividades BPM para una pyme, los procesos pueden ser creados, optimizados e implementados.



**Fig. 2.** Modelo de actividad BPM de una pyme. Fuente: Adaptado de [10].



En [11] integran los conceptos del ciclo de vida de los procesos en BPM desde la ejecución de los procesos con una metodología llamada AB-BPM, que ofrece un proceso de validación de mejora en dos fases: (i) simulación en la que se extraen las probabilidades de decisiones y métricas desde el log de eventos de una versión del proceso existente, y (ii) se generan trazas para las nuevas versiones de los procesos basados en este conocimiento. Los resultados de la simulación producen dos tipos de pruebas A y B.

**B. Gestión de Procesos de Negocios**

BPM se enfoca en el análisis y la operación de los procesos de negocios dentro de las organizaciones con el fin de buscar mejoras del proceso [12]. Una mejora de un proceso es aquella que permite alcanzar los objetivos y estrategias de una organización a través de la optimización y estandarización de actividades y procesos de la empresa [13].

Los factores contextuales de BPM son los siguientes: metas de BPM, características del proceso, especificaciones acerca de la organización donde BPM puede ser aplicado, y el entorno empresarial donde BPM se encuentra incrustado [14].

**C. Modelos Metodológicos para Mapas de Procesos**

En [15] llevan a cabo una revisión sobre los requerimientos necesarios para realizar el modelado de procesos del negocio, los cuales pueden ser: contexto de la organización y prioridades de gestión.

En [16] se enseña un modelo metodológico para mapas de procesos aplicables en el desarrollo de grandes sistemas computacionales. Los autores presentan una metodología basada en dos fases principales: Premapeo y mapeo del proceso de negocio. La fase de premapeo es fundamental para garantizar un correcto entendimiento de los procesos de la organización y facilita la integración de los procesos. Esta fase se concentra en tres actividades: agrupar los procesos y definir la jerarquía, sintetizarlos descriptivamente y definir un ambiente de trabajo donde todos los miembros del equipo tengan acceso a los procesos mapeados. En la Figura 3 se muestra el modelo de jerarquización y nomenclatura basado en

las funciones, macroprocesos, procesos, subprocesos y actividades resultantes de la fase premapeo.

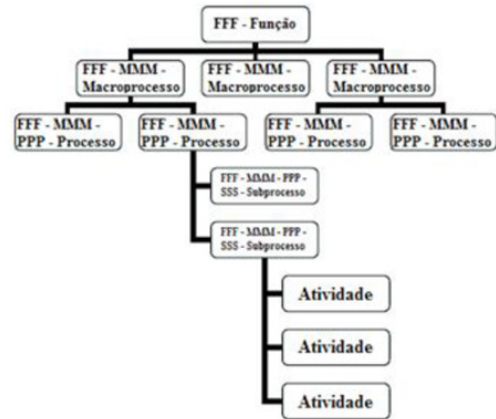


Fig. 3. Teoría de la Moda. Fuente: Adaptado de [16].

La fase de mapeo del proceso de negocio los autores la subdividen en tres etapas: iniciación, ejecución y validación del mapeado del proceso de negocio. La etapa de iniciación expone a los participantes del proyecto la agenda de trabajo, la metodología de trabajo y los macroprocesos o procesos que van a ser mapeados. En la etapa de ejecución se presenta el flujo de trabajo en el que se modela y analiza el proceso (ver Figura 4), en esta etapa se garantiza la calidad de la metodología propuesta al contar con cinco actividades claves del proceso: analizar la documentación, consolidar los resultados, enviar la lista de verificación, avalar las correcciones realizadas y construir los documentos con el mapeo y las correcciones requeridas (ver Figura 5 ).

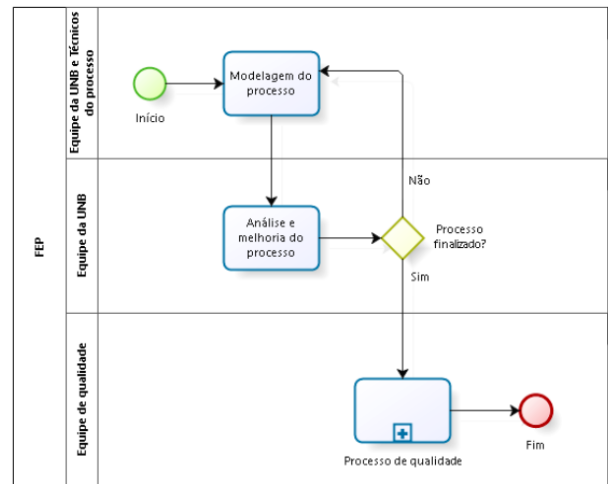


Fig. 4. Proceso de la fase de ejecución del mapeado del proceso. Fuente: Adaptado de [16].

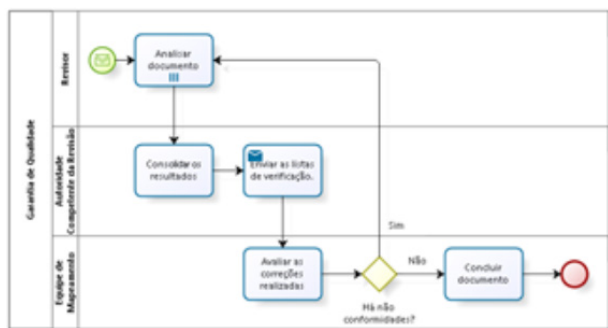


Fig. 5. Proceso de garantía de calidad del mapeado del proceso. Fuente: Adaptado de [16].

Cho y Lee [17] presentan un estudio sobre el modelo de evaluación y selección de procesos para la gestión de procesos de negocios. Se trata de un modelo basado en la web denominada Biz\_Tower para evaluar y seleccionar los procesos empresariales para BPM. El modelo utiliza BSC (Balanced Scorecard), Fuzzy AHP (Proceso de jerarquía analítica) y algoritmos de decisión para identificar una solución apropiada; los autores además muestran un ejemplo de la aplicación. Para lograrlo, utilizan AHP con el fin de generar la ponderación de las cuatro perspectivas del BSC y la ponderación de los criterios de evaluación, posteriormente construyen un modelo jerárquico en el que cada responsable de la decisión expresa la importancia relativa de los elementos de decisión (ver Figura 6).

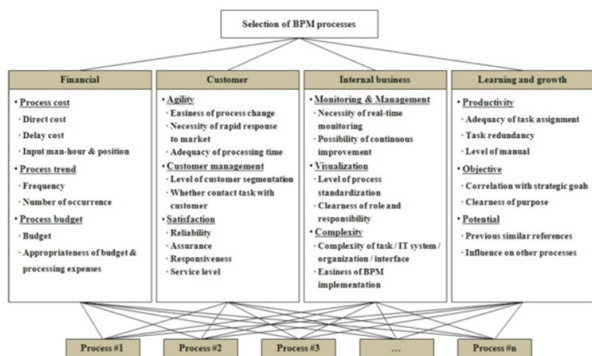


Fig. 6. Proceso de garantía de calidad del mapeado del proceso. Fuente: Adaptado de [17].

Posteriormente, utilizan Fuzzy para representar juicios subjetivos para capturar la incertidumbre. Finalmente, según los criterios de evaluación del proceso, los autores proponen un modelo de evaluación (ver Figura 7), cuyo objetivo es satisfacer los criterios para seleccionar el proceso BPM que se va a mejorar.

El estudio propuesto por los autores permite hacer un análisis de ponderación de los procesos, sin embargo, falta realizar una representación del proceso.

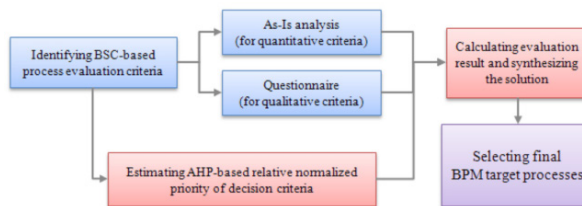


Fig. 7. Preguntas y posibilidades de adopción o rechazo de moda. Fuente: Adaptado de [17].

### III. MATERIALES Y MÉTODOS

La gestión del rendimiento del proceso es un sistema sistemático de identificación, visualización, medición, evaluación y mejora permanente de los procesos de la compañía usando métodos y principios basados en el enfoque del proceso [17]. Las principales características de la gestión de procesos son: orientación de procesos y gestión horizontal, al considerar la empresa como un sistema de procesos y actividades y de la filosofía del conocimiento del hombre. El aumento del rendimiento del proceso tiene como objetivo aumentar un valor añadido para los clientes y pertenece a los objetivos estratégicos básicos en las empresas modernas.

La gestión de procesos (BPM) como enfoque gerencial sistemático permite que el flujo de trabajo de la organización sea más efectivo, más eficiente y capaz de adaptarse a un entorno en constante cambio [18]. BPM intenta mejorar los procesos continuamente teniendo en consideración de las necesidades y requisitos del cliente.

El concepto de BPM no es nuevo, sin embargo, muchas compañías todavía están luchando con la implementación de BPM y con la aplicación del concepto y los métodos basados en el enfoque de procesos [19].

Se toma como base la metodología BPM, la cual consta de cinco etapas [20]: Definir (objetivos de la organización y sus procesos), detectar (problemas, causas y efectos), diseñar (procesos mejorados), implementar (procesos mejorados, soporte con

documentos), y monitorear (seguimiento, medición y control de procesos)

#### IV. RESULTADOS

A partir de estas etapas se propone un modelo de gestión por procesos compuesto por las siguientes fases (Figura 8):

- **Planeación estratégica:** en forma sistemática, este componente ayuda a especificar las características de la organización y los objetivos del proceso en función de los objetivos de la organización. Este componente permite hacer un mapeo de los procesos, identificando las actividades, tareas y variables que los integran; lo que facilita la detección de fallos en el flujo de datos. En esta etapa se agregan los nuevos elementos para mejorar el proceso.
- **Soporte tecnológico:** rediseño de flujo de trabajo (*workflow*). Cambio y adaptación de los procesos a nuevos elementos tecnológicos en el nivel de software. Aplicaciones que mejoren el flujo de los datos, garanticen disponibilidad de la información y reduzcan el tiempo de procesamiento.
- **Workflow y seguimiento:** seguimiento permanente a las actividades, tareas y variables del proceso. Para detectar cuellos de botella y otras eventualidades que se presenten durante el flujo de trabajo del proceso.

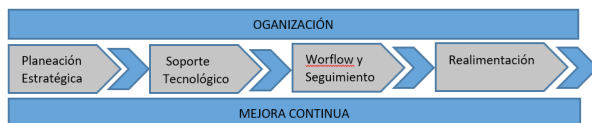


Fig. 8. Método propuesto. Fuente: elaboración propia

#### V. CASO DE ESTUDIO

Para la demostración del trabajo investigativo, se tomó un caso de estudio de una organización que presta servicios legales en diversas ramas del derecho. Uno de los servicios que presta, como actividad primaria, es el cobro prejurídico y la cobranza paralela

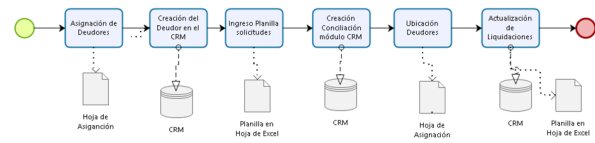


Fig. 9. Proceso de cobro prejurídico

El proceso del cobro prejurídico (Figura 9) tiene el siguiente flujo de actividades: En la primera fase, se realiza la asignación de los deudores mediante una hoja de asignación entregada por el cliente en formato Excel. La segunda actividad consta de la creación en un repositorio de datos de una CRM. El ingreso de los clientes al sistema se hace de manera manual. En la tercera fase del proceso, se ingresan las solicitudes en una planilla en formato Excel de manera manual. En la cuarta fase se crea una nueva tarea dentro del sistema CRM, y se asigna dicha tarea al administrador del sistema. En la quinta fase se realiza la ubicación de los deudores utilizando la información almacenada en la planilla de Excel. Por, último, en la sexta fase, se actualizan las respectivas obligaciones y costos de la cobranza prejurídica en el sistema CRM y en la planilla de Excel.

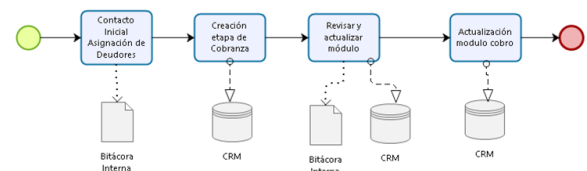


Fig. 10. Diagrama de procesos de cobranza paralela.

El proceso de cobranza paralela (Figura 10) tiene el siguiente flujo de actividades: En la primera fase del proceso se efectúa un contacto inicial con los deudores, y se diligencia manualmente una bitácora interna con la información del deudor y el valor por cobrar. En la segunda fase se crea dentro del sistema CRM la información del deudor. En la tercera fase se sincroniza la información del deudor que se encuentra almacenada en el repositorio del sistema CRM, con la información contenida en la bitácora interna. En la cuarta fase, se actualiza la información contenida dentro del repositorio del sistema del CRM con las actividades que realiza el gestor de cobro (etapas de negociación con el deudor, seguimiento del pago según acuerdo previo, fechas de pago, número de veces que logró exitosamente tener contacto con el deudor vía telefónica o por correo).

## VI. DISCUSIÓN

En la Tabla 1 se presenta una comparación entre varios modelos metodológicos de procesos y el modelo propuesto por los autores.

**Tabla 1.** Comparación entre Varios Modelos Metodológicos y el Modelo Propuesto

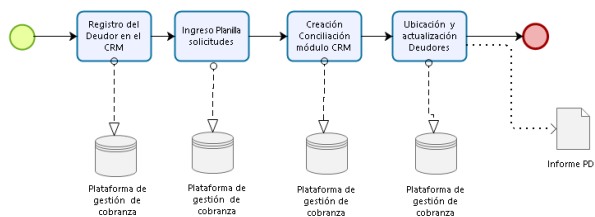
Modelo Metodológico de procesos [16]	Modelo de selección y evaluación de procesos [17]	Modelo propuesto
<p><b>Etapas:</b> dos fases denominadas premapeo y mapeo.</p> <p><b>Propósito:</b> contextualizar la organización, describir sus procesos, rediseñarlos, monitorearlos a partir de su flujo de trabajo.</p> <p><b>Nivel de detalle:</b> se jerarquizan los procesos a nivel de subprocesos o actividades, y tareas.</p>	<p><b>Etapas:</b> dos fases denominadas selección y evaluación.</p> <p><b>Propósito:</b> para lograrlo presentan un modelo basado en la web denominada <i>Biz_Tower</i>; para evaluar y seleccionar los procesos empresariales para BPM. El modelo utiliza BSC (<i>Balanced Scorecard</i>) y <i>Fuzzy AHP</i> (Proceso de jerarquía analítica), además de algoritmos de decisión para identificar una solución apropiada.</p> <p><b>Nivel de detalle:</b> análisis ponderado de procesos de la organización.</p>	<p><b>Etapas:</b> cuatro fases denominadas análisis de proceso, monitoreo, soporte tecnológico, mejora continua.</p> <p><b>Propósito:</b> contextualizar la organización, describir los procesos de la organización, identificar fallos en las actividades y tareas de los procesos. A partir de los errores, se proponen herramientas y estrategias para mejorar los procesos.</p> <p><b>Nivel de detalle:</b> análisis de procesos, mejora, monitoreo y optimización.</p>

Modelo Metodológico de procesos [16]	Modelo de selección y evaluación de procesos [17]	Modelo propuesto
<p><b>Funcionalidad:</b> durante la ejecución se presenta el flujo de trabajo donde se modela y analiza el proceso, en esta fase se garantiza la calidad de la metodología propuesta al contar con cinco actividades claves del proceso: analizar la documentación, consolidar los resultados, enviar la lista de verificación, avalar las correcciones realizadas y construir el documento con el mapeo y las correcciones requeridas.</p> <p><b>Aspectos para mejorar:</b> la propuesta metodológica presentada por los autores permite realizar el levantamiento de los procesos dentro de la organización para mapearlos, sin embargo, falta presentar estrategias de mejoramiento del proceso organizacional.</p>	<p><b>Funcionalidad:</b> se utiliza AHP para generar la ponderación de las cuatro perspectivas del BSC y la ponderación de los criterios de evaluación, posteriormente se construye un modelo jerárquico donde cada responsable de la decisión expresa la importancia relativa de los elementos de decisión. Luego utilizan <i>Fuzzy</i> para representar juicios subjetivos para capturar la incertidumbre. Finalmente, según los criterios de evaluación, se presenta un modelo de evaluación del proceso.</p> <p><b>Aspectos para mejorar:</b> según los criterios de evaluación del proceso, se presenta un modelo para seleccionar el proceso BPM que se va a mejorar. Falta realizar una representación del proceso, y faltan estrategias de mejoramiento del proceso organizacional.</p>	<p><b>Funcionalidad:</b> se especifican las características de la organización indicando los objetivos que se articulan con los procesos. Se identifica el proceso que se va a analizar. Posteriormente se analiza jerárquicamente hasta el nivel de variables. De acuerdo al estado del proceso, se mejora a partir de tecnologías y metodologías.</p> <p><b>Aspectos para mejorar:</b> reorientación permanente del proceso a partir de una base de datos de experiencias o conocimiento.</p>

Fuente: elaboración propia



El modelo propuesto (Figura 11) facilita el análisis de los procesos a través de su estrategia jerárquica de medición que llega hasta el nivel de variables, lo que no permiten los modelos de gestión de procesos, que delimitan su funcionalidad hasta un nivel de tareas y actividades. La forma modular del modelo flexibiliza su comportamiento en ejecución agilizando la comunicación de datos entre etapas, los cuales se comportan como insumo para determinar el estado del proceso en cada fase propuesta. La evaluación del proceso permite identificar los aspectos por mejorar y establecer las tecnologías necesarias para su optimización. Los modelos actualmente aplicados no relacionan los objetivos del proceso con los indicadores establecidos por la organización, lo que dificulta la interpretación de los resultados obtenidos durante las ejecuciones o corridas del proceso.



**Fig. 11.** Modelo propuesto para el proceso de cobro prejurídico y cobranza paralela

Los modelos actuales se centran en el rediseño y la ejecución del proceso, dejando a la organización la tarea de hacer un esfuerzo adicional respecto a la interpretación del proceso y su comportamiento. Se hace necesario tener una base de datos de conocimiento, que en forma automática alimente el sistema con historias o casos de experiencias que realimenten el proceso y faciliten su optimización.

## VII. CONCLUSIONES Y TRABAJO FUTURO

Las organizaciones se encuentran estructuradas y orientadas hacia los procesos. Para mejorar el uso de los recursos tanto financieros, como de tiempo y de personal, es importante conocer y monitorear periódicamente la gestión actual de cada proceso.

La metodología BPM permite diseñar, monitorear y mejorar continuamente las actividades y los procesos que se desarrollan dentro de las organizaciones, el resultado es una mayor competitividad y conocimiento de su negocio.

Los métodos actuales que utilizan la metodología BPM se enfocan en el rediseño y la ejecución del proceso, y dejando a la organización la responsabilidad de realizar esfuerzos adicionales respecto a la interpretación del comportamiento de sus procesos, lo que resulta en reprocesos y uso adicional de recursos.

El método propuesto posibilita a las organizaciones mejorar en los flujos de trabajo, lo que reduce el consumo innecesario de recursos y agiliza la comunicación entre cada una de las etapas. La evaluación del proceso permitió identificar los aspectos por mejorar y coadyuvó a establecer las tecnologías necesarias para su optimización.

Como trabajo futuro se propone diseñar y desarrollar un software que integre las fases del método propuesto. El software permitirá a las organizaciones analizar y modelar de una forma más eficaz y de mejor calidad cada una de las actividades que componen los procesos del negocio.

## REFERENCIAS

- [1] J. Mejía, E. Muñoz y M. Muñoz, “Reinforcing the applicability of multi-model environments for software process improvement using knowledge management”, *Science of Computer Programming*, vol. 121, Issue C, pp. 3-15, 2016.
- [2] P. Bazán, J. M. Garro, R. Giandini y J. Díaz, “Mejora de la monitorización y ejecución de procesos de negocio con integración y socialización”, en *Computing Conference (CLEI)*, 2015, pp. 1-12.
- [3] d. S. JR, R. T., E. Flavio, G. de Deus, B. A. de Sousa, N. G. Villapouca, M. T. Holanda y R. ... Santos, “Metodologia para Modelagem de Processo de Negócio”, en *10th Iberian Conference on information Systems and Technologies (CISTI)*, 2015, pp. 1-6.
- [4] S. S. S. Cherfi, S. Ayad y I. Comyn-Wattiau, “Improving business process model quality using domain ontologies”, *Journal on Data Semantics*, vol. 2, N.º 2, pp. 75-87, 2013.
- [5] R. Dijkman, S. V. Lammers y A. de Jong, “Properties that influence business process management maturity and its effect on organizational performance”, *Information Systems Frontiers*, vol. 18, N.º 4, pp. 717-734, 2016.
- [6] S. Zugal, P. Soffer, C. Haisjackl, J. Pinggera, M. Reichert y B. Weber, “Investigating expressiveness



- and understandability of hierarchy in declarative business process models”, *Software & Systems Modeling*, vol. 14, N.º 3, pp. 1081-1103, 2015.
- [7] N. Z. Haddar, L. Makni y H. B. Abdallah, “Literature review of reuse in business process modeling”, *Software & Systems Modeling*, vol. 13, N.º 3, pp. 975-989, 2014.
- [8] A. Yousfi, R. Saidi y A. K. Dey, “Variability patterns for business processes in BPMN”, *Information Systems and e-Business Management*, vol. 14, N.º 3, pp. 443-467, 2016.
- [9] M. Kurz, U. Lembecke, y M. Lederer, “Applying the Strategy-Oriented Business Process Modeling to S-BPM”, *BPM ONE - Application Studies and Work in Progress*, en *S-BPM ONE – Application Studies and Work in Progress: Proceedings of 6th International Conference, S-BPM ONE 2014, Eichstätt, Germany, April 22-23 2014*, pp. 67-84.
- [10] D. W. W. Christoph Piller, “Production Planning for SMEs – Implementation of Production Planning with Subject-Oriented Business Process Management (S-BPM)”, *S-BPM ONE - Application Studies and Work in Progress: Proceedings of 6th International Conference, S-BPM ONE 2014, Eichstätt, Germany, April 22-23 2014*, pp. 164-173.
- [11] S. Satyala, I. Webera, H.-Y. Paik, C. Di Ciccioc y J. Mendlinge, “Business process improvement with the AB-BPM methodology”, *Information Systems*, jul. 2018. DOI: 10.1016/j.is.2018.06.007
- [12] W. M. P. van der Aalst, M. La Rosa, F. M. Santoro, “Business Process Management. Don’t forget to improve the process!”, *Business & Information Systems Engineering*, vol. 58, N.º 1, pp. 1-6, 2016.
- [13] K. V., “Business Process Management”, en: *Business Process Management within Chemical and Pharmaceutical Industries*, Berlin: Springer, 2014.
- [14] S. Z. T. S. Jan vom Brocke, “On the role of context in business process management”, *International Journal of Information Management*, N.º 36, pp. 486-495, 2016.
- [15] S. N. Afef Awadid, “Consistency requirements in business process modeling: a thorough overview,” *Softw System Model*, 2017.
- [16] R. T. de Sousa, F. E. de Deus, B. A. de Sousa, N. G. Villapouca, M. T. Holanda, A. P. Araújo y A. J. M. ... de Moraes, “Methodology for business process modeling: A study case within the Brazilian Ministry of Planning, Budget and Management”, en *Information Systems and Technologies (CISTI)*, 2015, pp. 1-6.
- [17] C. Cho y S. Lee, “A study on process evaluation and selection model for business process management”, *Expert Systems with Applications*, vol. 50, pp. 6339-6350, 2011.
- [18] K. Ternai, M. Török y K. Varga, “Combining knowledge management and business process management—a solution for information extraction from business process models focusing on BPM challenges”, en *International Conference on Electronic Government and the Information Systems Perspective*, 2014, pp. 104-117.
- [19] A. Gazova, Z. Papulova y J. Papula, “The application of concepts and methods based on process approach to increase business process efficiency”, *Procedia Economics and Finance*, vol. 39, pp. 197-205, 2016.
- [20] Villanova University, “Villanova University”, 2015. [En línea]. Disponible en: <https://www.villanovau.com/resources/bpm/basic-bpm-methodology/>.
- [21] K. T. T. Varga, “Combining Knowledge Management and Business Process Management – A Solution for Information Extraction from Business Process Models Focusing on BPM Challenges”, en *International Conference on Electronic Government and the Information Systems Perspective*, Springer, Cham, 2014, pp. 104-117.

## Capítulo XIII

# PROCEDIMIENTO PARA ESPECIFICAR Y VALIDAR REQUISITOS DE SOFTWARE

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

Docente, Universidad Católica de Pereira

*Juan Guillermo Gálvez Botero - jggalvez@utp.edu.co*

Universidad Tecnológica de Pereira

*Luis E. Peláez Valencia - luis.pelaez@ucp.edu.co*

Docente, Universidad Católica de Pereira

### I. INTRODUCCIÓN

En la industria del software, uno de los principales objetivos es obtener productos de calidad. Para ello se aplican diferentes métodos y modelos bajo procesos estandarizados, que facilitan tanto la gestión del proyecto como el desarrollo del software. Tales métodos y modelos pueden ser acompañados por el uso de técnicas, procedimientos y herramientas que contribuyen al aseguramiento de la calidad de estos productos intangibles.

Es de anotar que los sistemas de software son cada vez más complejos, por lo que las exigencias de calidad sobre su desarrollo son mayores. Es necesario que el desarrollo de software sea más riguroso para obtener un producto de alta calidad, con el fin de reducir el número de errores presentes en su construcción. La identificación y el tratamiento de estos errores en etapas tempranas del proyecto de software es crucial para disminuir los costos de su operación y evitar errores en etapas posteriores del desarrollo.

Una de las principales etapas en el desarrollo de software es el desarrollo de los requisitos, la cual a su vez se divide en cuatro etapas: identificación de requisitos, análisis de requisitos, especificación de requisitos y validación de requisitos [1]. Entre ellas cobra especial importancia la especificación de requisitos, que se refiere típicamente a la producción

de un documento o a su equivalente electrónico, que puede estar sistemáticamente repasado, evaluado y aprobado [1 p41].

Para lograr un producto de software de alta calidad, los requisitos deben cumplir con propiedades como: completo, correcto, realizable y verificable, entre otras [2-3]. Sin embargo, en algunas ocasiones los requisitos no satisfacen dichas propiedades, lo que se traduce en errores y mala calidad del producto.

Comúnmente, durante el trabajo con los requisitos es cuando se interactúa por primera vez con el cliente, quien carece de una solución o presenta una necesidad. Este momento es fundamental para conseguir que el producto desarrollado cumpla con las expectativas trazadas por el usuario.

Por esto, la gestión de requisitos en la Ingeniería de Software se ha convertido en una de las principales estrategias para garantizar, desde las primeras etapas del proceso de desarrollo de software, la calidad de las aplicaciones resultantes [4-5].

De igual forma, los requisitos marcan el punto de partida para actividades como la planeación, en lo que se refiere a las estimaciones de tiempos y costos, así como a la definición de recursos necesarios. Asimismo, se convierten en un insumo importante para obtener una definición completa del alcance del proyecto [6].

Al respecto, se menciona también que “un gran porcentaje de los proyectos de software que fracasan es atribuible a un deficiente tratamiento de los requisitos” [6], y esto ocurre porque la definición, especificación, y administración de los requisitos no son las adecuadas. Igualmente, su inadecuada gestión puede llevar a que se presenten problemas relacionados con el poco involucramiento del usuario, la incompletitud de requisitos y su mala trazabilidad, como lo afirman [7] y [8].

En la misma línea, la especificación y validación de requisitos cumplen un papel fundamental en el proceso de desarrollo de software, pues se centran en la adecuada definición de lo que se requiere producir. Llevarlas a cabo permite definir especificaciones correctas, claras y menos ambiguas en relación con las necesidades propias de los clientes, y así se evitan problemas causados por un deficiente desarrollo de los requisitos [9].

Lo anterior permite vislumbrar la importancia que tiene para la calidad del software, implementar en cada proyecto de desarrollo de software una correcta ingeniería de requisitos. Por ello, el proyecto de investigación se basa en definir un procedimiento para especificar y validar requisitos de software que puedan ser utilizados por las Mipymes desarrolladoras de software, con la intención de contribuir al mejoramiento de la calidad de sus procesos y productos. La revisión de estudios previos realizados alrededor de los requisitos, en el ámbito regional, nacional e internacional, fue fundamental para el desarrollo del procedimiento.

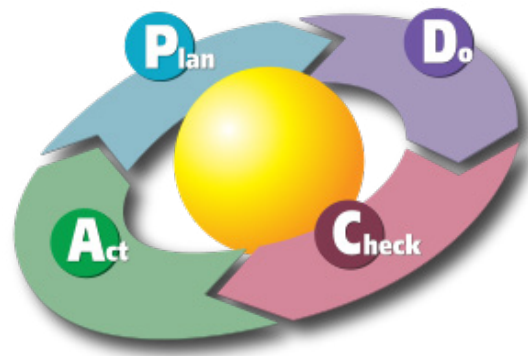
Para lograr lo anterior, el proyecto se adelantó siguiendo las etapas propuestas por el Ciclo PHVA (Planear-Hacer-Verificar-Actuar), también conocido como círculo de Deming o espiral de mejora continua, considerando que se desea elaborar un procedimiento como un elemento de mejora para uno de los procesos de desarrollo de software, mediante una investigación que consta inicialmente de una exploración respecto a la especificación y validación de requisitos, que tiene en cuenta tanto el estado del arte como una descripción y caracterización de una muestra por conveniencia representativa de la población de empresas desarrolladoras de software de la ciudad de Pereira.

De igual forma, se elabora un procedimiento para especificar y validar requisitos de software, así como para validar el mismo procedimiento a través de su aplicación mediante dos pilotos de aplicación en dos unidades de análisis, representadas por dos Mipymes desarrolladoras de software.

## II. METODOLOGÍA

El proyecto se realiza siguiendo las etapas propuestas por el Ciclo PHVA, del cual puede verse un esquema general en la Figura 1. Se considera que el procedimiento que se elabora es un elemento que se ajusta a las necesidades de las empresas, con el fin de que ellas puedan usarlo para mejorar uno de los procesos de desarrollo de software: el relacionado con los requisitos.

Es de anotar que definir un nuevo procedimiento para una empresa (que reemplace lo que se hacía antes) es en sí una mejora de procesos. A continuación se explican las actividades que se llevaron a cabo en cada una de las etapas del ciclo PHVA.



**Fig. 1.** Estructura del ciclo PHVA

Fuente: Tomado de [14]

### A. Etapa 1: PLANEAR

Esta etapa a su vez se divide en dos grandes actividades: 1) La primera actividad es una exploración que se hizo con respecto a la especificación y validación de requisitos, teniendo en cuenta tanto el estado del arte como una descripción y caracterización de una muestra por conveniencia representativa de una población (empresas desarrolladoras de software de la ciudad de Pereira). 2) La segunda actividad

consiste en la generación del procedimiento para la especificación y validación de requisitos de software.

Es fundamental explicar en detalle cada una de las actividades, a fin de proveer la información necesaria para determinar cómo se consiguió la finalidad de cada una.

**Actividad 1: Exploración.** Durante la investigación se realiza un levantamiento de información en diferentes fuentes de información como repositorios institucionales, bases de datos especializadas y bibliografía física de las diferentes bibliotecas de la región, con el propósito de consolidar un referente contextual que comprendiera una descripción del área problemática y una revisión de antecedentes, conocido también como estado del arte de la investigación, el cual estuvo desagregado desde el ámbito regional, nacional e internacional; además se consolidó otro capítulo denominado marco teórico, donde se detallaron las diferentes teorías y conceptos relacionados con el proceso de desarrollo de los requisitos de software y en especial con su especificación y validación.

**Instrumentos utilizados:** Para la recolección e identificación de los antecedentes, tanto del orden regional, como nacional e internacional, se usaron tablas como la que muestra la Figura 2, que describen de manera concreta la información relevante de los antecedentes:

**Tabla 1.** Instrumento utilizado para recolección de antecedentes

AUTOR	TITULO	UNIVERSIDAD	TÉCNICAS ESPECIFICACIÓN	TÉCNICAS VALIDACIÓN	FALENCIAS DEL PROYECTO	APORTE ESPECIFICO

Posteriormente, se llevó a cabo un encuentro con los empresarios del software de la ciudad de Pereira, que contó con el acompañamiento de Parquesoft Pereira que incluye a las empresas que hacen parte de este modelo de emprendimiento. Los criterios de selección de la muestra fueron los siguientes:

Empresas cuya actividad económica esté catalogada en la División 62, Desarrollo de sistemas informáticos (planificación, análisis, diseño, programación, pruebas), consultoría informática y actividades relacionadas, pero más específicamente en el

desarrollo de sistemas informáticos (planificación, análisis, diseño, programación, pruebas) con código CIU 620 6201, según la clasificación Industrial Internacional Uniforme impartida por el Departamento Administrativo Nacional de Estadística [10].

- Empresas que se encuentren ubicadas en la ciudad de Pereira.
- Empresas prestas a suministrar la información requerida relacionada con el proyecto.

La información de la muestra se presenta en la Tabla 1

**Tabla 2.** Información de la Muestra

<b>Diseño muestra</b>	Muestra por conveniencia
<b>Población objetivo</b>	Mipymes desarrolladoras de software de la ciudad de Pereira
<b>Tamaño de la muestra</b>	17 empresas de la ciudad de Pereira (de 30 empresas convocadas).
<b>Momento estadístico</b>	Octubre de 2015 – enero de 2016.
<b>Financiación</b>	Recursos propios.

El encuentro sostenido con las empresas tuvo las siguientes finalidades:

- Presentar el proyecto con el ánimo de poner en contexto al sector productivo sobre el trabajo de investigación que se estaba llevando a cabo.
- Aplicar una encuesta como instrumento de levantamiento de información, cuyo objetivo era diagnosticar, proponer y describir mejoras en pro de tener un proceso de desarrollo de requisitos de calidad haciendo énfasis, en este caso, en la especificación y validación de requisitos como objeto de estudio de la investigación.

Un ejemplo del instrumento utilizado para consolidar la información obtenida de las encuestas se muestra en la Figura 3.

**Tabla 3.** Instrumento utilizado para consolidar información de encuestas

TÉCNICA	Empresa 1	Empresa 2	Empresa 3	% de uso
Técnica 1				
Técnica 2				

**Actividad 2: Descriptiva.** Se recibe como entrada el marco teórico, los antecedentes y la descripción y caracterización que detallan cómo las empresas desarrolladoras de software llevan a cabo la especificación y validación de requisitos (Actividad 1). Esta información fue analizada a través de una comparación con los antecedentes y el marco teórico, y se determinaron las entradas, las precondiciones, las actividades y tareas del procedimiento, las salidas y el objetivo.

La comparación se realiza usando tablas, como las que se muestran en la Figura 4, donde se identificaron los elementos que contemplan los diferentes autores del marco teórico que fueron referentes del proyecto para la especificación y validación de requisitos, las técnicas que proponen los autores estudiados en los antecedentes y las prácticas usadas en las empresas, junto con el cubrimiento que logra cada práctica del elemento.

**Tabla 4.** Instrumento utilizado para recolectar información del marco teórico

Técnica	Antecedente 1	Antecedente 2	Antecedente 3	Puntaje
Técnica 1				
Técnica 2				
...				

Técnica	Autor 1	Autor 2	Autor 3	Puntaje
Técnica 1				
Técnica 2				
...				

Después de haber determinado las técnicas mencionadas o recomendadas desde el marco teórico y los antecedentes, se realizó un análisis de la información a través de tablas de cruce como las que presenta la Figura 5, que permitieron encontrar las mejores prácticas, que luego se organizaron y complementaron para que incluyeran todos los elementos requeridos en un procedimiento.

**Tabla 5.** Ejemplo de tabla de cruce de información

Técnica	MARCO TEÓRICO	ANTECEDENTES	Total	Cubrimiento*
Técnica 1				
Técnica 2				
...				

\*Requerido, Recomendado, Opcional

Finalmente, se efectuó un análisis de las técnicas requeridas y recomendadas usadas por las empresas, teniendo en cuenta los resultados de la encuesta planteada en la Etapa 1.

El diseño del procedimiento para la especificación y validación de requisitos de software se llevó a cabo bajo el estándar SPEM soportado con una descripción textual explícita, así como con un conjunto de guías, formatos y recomendaciones. Lo anterior, con la intención de facilitar su presentación y despliegue.

Las tablas descritas en las figuras anteriores fueron el insumo principal para establecer el diseño del procedimiento para la especificación y validación de requisitos de software, inicialmente en Excel, pues se consideró por conocimiento de los autores que es la herramienta que con mayor frecuencia usan las Mipymes para especificar sus procesos y procedimientos. En la Figura 6 se presenta el formato que se propuso para el procedimiento.

**Tabla 6.** Formato inicial del procedimiento

Título:								
Descripción:								
Precondiciones:	No.	Precondición						
Entradas:	No.	Entrada						
Actividades-Tareas:	No.	Actividad	No.	Técnica	No.	Tareas (pasos)	Responsables	Guías y Herramientas
Salidas:	No.	Salida						

Para una mejor presentación y despliegue del procedimiento, inicialmente diseñado en Excel, se modeló bajo el estándar SPEM (Software Process Engineering Metamodel, es una especificación de OMG Object Management Group —OMG— basada en Meta Object Facility —MOF—. SPEM permite representar una familia de procesos de desarrollo de software y sus componentes) utilizando el Eclipse Process Framework Composer [12], como se puede evidenciar en la Figura 7.





Fig. 2. Procedimiento modelado en SPEM

Se utilizó esta herramienta porque provee un conjunto de elementos de modelado de procesos para describir cualquier proceso de desarrollo de software, sin agregar modelos o restricciones de alguna área o disciplina específicas, tal como gestión de proyectos o análisis [11].

## B. Etapa 2: HACER

Esta etapa comprende dos pilotos de aplicación con dos unidades de análisis, mediante los cuales se presentó la experiencia vivida en dos Mipymes desarrolladoras de software, en cada caso un investigador a cargo realiza la aplicación del piloto del procedimiento, y posteriormente se consolidaron los resultados en un documento que describe cada una de las actividades propuestas para su aplicación.

Las actividades llevadas a cabo en esta etapa fueron:

- Selección del proyecto donde se aplicó el piloto en la empresa.
- Recopilación de datos de proyectos anteriores y determinación de aquellos que sirvieron para analizar los resultados del piloto. Por ejemplo: tiempo invertido en requisitos, errores, cantidad de modelos/documentos elaborados, revisiones realizadas, satisfacción de los desarrolladores mediante la aplicación de una encuesta de expectativas y una encuesta de rendimiento.
- Definición de la forma en la cual se le haría el seguimiento y se medirían los resultados (a partir de los datos identificados en el paso anterior).
- Identificación de los actores que participarían y el rol de cada uno.

- Comprometer a los participantes, explicándoles cómo se haría el seguimiento y la importancia de su participación crítica.
- Capacitación sobre el procedimiento.
- Ejecución del proyecto con el nuevo procedimiento.
- Recopilación de los datos necesarios (de acuerdo con lo definido en la planeación).

**Instrumentos utilizados:** se empleó una encuesta de expectativas y una encuesta de rendimiento con el ánimo de determinar el nivel de satisfacción, las cuales condujeron a aprender de la experiencia, pues aportaron a los ajustes del procedimiento que se realizaron en la etapa 1 (Actuar), además se utilizó un formato donde reposa la información obtenida con el ánimo de analizar y reflexionar sobre la experiencia. De cada paso se tienen registros: documentos, actas de reuniones, etc.

**Esfuerzo invertido en esta etapa:** nueve horas semanales por persona (dos personas) durante seis meses.

## C. Etapa 3: VERIFICAR

Esta etapa tiene como propósito verificar que la aplicación del procedimiento se haya realizado de la forma indicada, con la intención de determinar los resultados obtenidos y las mejoras o ajustes por implementar, todo ello debidamente documentado.

Las actividades llevadas a cabo en esta etapa fueron:

- Análisis de los datos recolectados.
- Documentación de los resultados, destacando cuándo se presentaron logros importantes o mejoras y cuándo no. En los casos en los que no se lograron mejoras se identificaron y documentaron las causas.
- Seguimiento de la aplicación del procedimiento (realización de las actividades y aplicación de las técnicas propuestas) y registro de los casos en los que no se aplicó el procedimiento de manera correcta.
- Documentación de lecciones aprendidas: lo bueno (que se debe continuar) y lo malo (que se debe corregir).

**Esfuerzo invertido en esta etapa:** nueve horas semanales por persona (dos personas) durante tres meses.

#### D. Etapa 4: ACTUAR

En esta etapa se hacen los ajustes al procedimiento propuesto de acuerdo a las dificultades y mejoras obtenidas durante la etapa de verificación, con la intención de lograr un procedimiento más ajustado a las necesidades de las Mipymes. Para ello se llevaron a cabo las siguientes actividades:

- Realizada la aplicación y la validación del procedimiento propuesto para la especificación y validación de requisitos, se retomaron las consideraciones pertinentes y los resultados obtenidos con el ánimo de hacer los ajustes correspondientes (rediseño del procedimiento mediante modelado en SPEM para una mejor presentación y despliegue).
- Se efectuaron los ajustes necesarios en el procedimiento.
- Se comunicaron los resultados a las empresas para que puedan usar el nuevo procedimiento en otros proyectos.

**Esfuerzo invertido en esta etapa:** nueve horas semanales por persona (dos personas) durante tres meses.

### III. RESULTADOS

Los resultados obtenidos permiten describir los logros alcanzados en cada una de las etapas de la metodología propuesta. La primera etapa tuvo como propósito realizar un levantamiento del estado del arte de los requisitos, así como una descripción y caracterización de la forma en que las Mipymes desarrolladoras de software de la ciudad de Pereira llevan a cabo la especificación y validación de requisitos, también se diseñó el procedimiento para especificar y validar requisitos de software.

Así, pues, una vez analizado el cubrimiento de las técnicas aportadas por los autores del marco teórico, y las obtenidas de los antecedentes a través de los instrumentos mencionados, se determina que las

técnicas requeridas para la especificación de requisitos de software son las siguientes:

- Documento de especificación de requisitos del software (SRS, por sus siglas en inglés).
- Diagramas de casos de uso.

De igual forma, se determina que las técnicas *recomendadas* para la especificación de requisitos de software son las presentadas a continuación:

- Diagrama de flujo de datos (DFD).
- Prototipos
- Historias de usuario

Las demás técnicas se establecen como opcionales y eventualmente podrían aportar al desarrollo de alguna de las diferentes actividades propuestas en el procedimiento.

Con respecto a las técnicas para la validación de requisitos de software propuestas por los autores del marco teórico y las obtenidas de los antecedentes, se pudo concluir que las siguientes técnicas se requerirán para el procedimiento:

- Prototipos

En la misma línea se define que, de acuerdo al análisis realizado, aquellas técnicas recomendadas para la validación de requisitos de software deberán ser las siguientes:

- *Walkthrough* o recorridos guiados

Una vez hecho el análisis de las técnicas que, según el resultado de las encuestas, usan las Mipymes desarrolladoras de software de la ciudad de Pereira según la muestra, se encuentra que la mayoría de las técnicas son conocidas y en varios casos utilizadas para llevar a cabo los procesos de especificación y validación de requisitos de software.

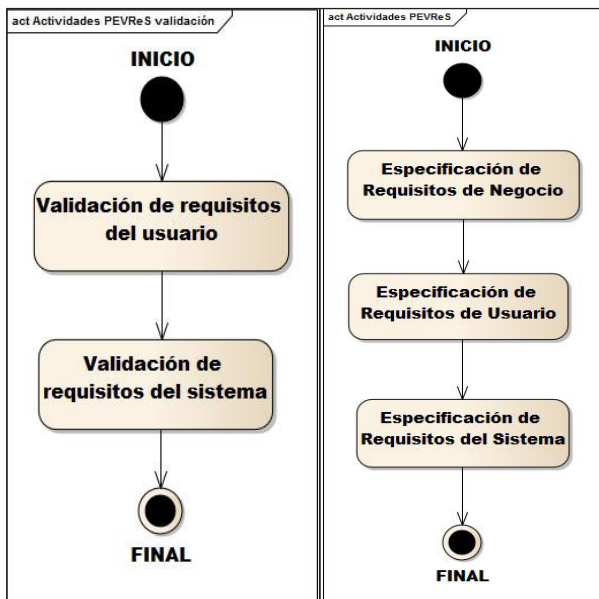
Pese a lo anterior, se nota claramente que solo dos de las 17 empresas (un 12 %) aproximadamente, hacen uso de las técnicas de especificación requeridas y el resto sólo hace uso de alguna de ellas. Por el contrario, para las técnicas recomendadas se encuentra que solo

una de las 17 empresas emplea al menos dos de las tres técnicas mencionadas, mientras que las demás solo emplean una técnica o no usan ninguna.

Con respecto a las técnicas para la validación de requisitos con carácter de requeridas, se encuentra que el 35 % de las empresas las usa, mientras que las técnicas recomendadas las usa también un 35 % de ellas, lo que equivale a seis de las 17 empresas.

Lo anterior permitió inferir que era necesario motivar la implementación de estas técnicas para facilitar el desarrollo de los requisitos y contribuir a su calidad. Sin embargo, las técnicas empleadas por el procedimiento no necesariamente deberían tener el mismo criterio de cubrimiento obtenido en el análisis anterior (requerida, recomendada, opcional), pues el mismo procedimiento podría proponer un diseño con criterios propios sin dejar de lado estos resultados.

El diseño del procedimiento para la especificación y validación de requisitos de software que hace parte de la metodología (Etapa 1: Descriptiva), se llevó a cabo con diagramas de actividades soportados con una descripción textual explícita, y contaban además con un conjunto de guías y recomendaciones.



**Fig. 3.** Actividades que componen el procedimiento de especificación y validación de requisitos de software elaborado

Es importante aclarar que, para el procedimiento, los requisitos de software (funcionales y no funcionales) se derivan de los requisitos del sistema, entendiendo que el sistema tiene componentes de software, tal como se manifiesta en [1 p34].

En la segunda etapa se realizó la validación mediante dos pilotos de aplicación en dos unidades de análisis (empresas desarrolladoras de software de la ciudad de Pereira), la tercera etapa consistió en la verificación de los datos recolectados en la etapa inmediatamente anterior y, finalmente, en la cuarta etapa, se realizaron los respectivos ajustes al procedimiento.

Como parte de la planeación general para ambos pilotos de aplicación, se muestran los criterios de selección que se tuvieron en cuenta para elegir los proyectos que se han de analizar en ambas empresas. No se consideraron como criterios otros aspectos, como lenguaje, plataforma, líneas de código, uso de frameworks, etc., que podrían representar variaciones en cada proyecto, pues no se consideran relevantes para la etapa de requisitos, y de este modo se da mayor flexibilidad a la aplicación del procedimiento, pero, por supuesto, dejando abierta la posibilidad a trabajos futuros. También se llevó a cabo la presentación y capacitación sobre el procedimiento al equipo de desarrollo.

**Tabla 7.** Planeación General del Piloto de Aplicación

CRITERIOS DE SELECCIÓN DEL PROYECTO	
Empresas para visitar	Mipymes desarrolladoras de software de la ciudad de Pereira.
N.º de responsables	Máximo dos personas encargadas del desarrollo de los requisitos.
Tipo de proyecto para intervenir	Marco tradicional de los datos (transaccionales).
Duración del proyecto (tiempo)	Entre 4 y 6 meses incluyendo desarrollo, pruebas y entrega.
Dependencias para visitar	Área de desarrollo de software.
Rol de las personas involucradas	Analistas, desarrolladores, líderes de procesos, directores de proyectos, representantes legales de las empresas.

CRITERIOS DE SELECCIÓN DEL PROYECTO	
Estado del proyecto nuevo	No haber llegado a la etapa de especificación de requisitos.
INFORMACIÓN PARA RECOLECTAR	
N.º de revisiones de los requisitos	Cantidad de veces que fue verificada la especificación de requisitos.
Tiempo en requisitos	Tiempo total invertido en el desarrollo de los requisitos.
Tiempo en especificación	Tiempo invertido en la especificación y verificación de requisitos.
Tiempo en validación	Tiempo invertido en la validación de requisitos.
Técnica para especificar	Técnica que se usó para especificar requisitos.
Técnica para validar	Técnica que se usó para validar requisitos.
N.º de defectos en la verificación de requisitos	Número de defectos encontrados en la especificación de requisitos durante la verificación.
N.º de defectos en la validación de requisitos	Número de defectos encontrados en la especificación de requisitos durante la validación.
N.º de defectos en producción	Número de defectos atribuibles a los requisitos encontrados en la etapa de producción mediante el uso de técnicas no suministradas por PEVReS y reportados por las pymes.
Nivel de satisfacción del equipo	Nivel de satisfacción del equipo desarrollador respecto al trabajo realizado con los requisitos.
N.º de modelos elaborados	Cantidad de diagramas/modelos elaborados durante la especificación de los requisitos.
N.º de documentos elaborados	Cantidad de documentos resultantes de la especificación de requisitos.

De igual forma, fue necesario hacer un seguimiento y acompañamiento a la aplicación del procedimiento en cada piloto de aplicación, con el ánimo de recolectar las dificultades o mejoras que contribuyeran a su refinamiento para aplicaciones futuras; asimismo, se hizo el seguimiento a la aplicación del piloto para determinar si se realizaron las actividades propuestas y se aplicaron las técnicas establecidas, llevando además el registro de los casos en los que no se aplicó el procedimiento de manera correcta.

Como parte de la Etapa 3 de la metodología del proyecto, y con respecto a los datos obtenidos en la fase inmediatamente anterior, es importante indicar que una vez analizados los datos obtenidos en cada uno de los proyectos del piloto de aplicación N.º 1, se puede observar, según la Figura 9, que al aplicar el procedimiento para especificar y validar requisitos se presenta un aumento notable en el tiempo empleado, tanto para especificarlos como para validarlos, lo que afecta a su vez el tiempo total para desarrollarlos.

Esto se atribuye, entre otras razones, a que existe una curva de aprendizaje de la cual las Mipymes deben ser conscientes, pues el procedimiento establece la ejecución de actividades bien definidas, documentadas paso a paso, que requieren tanto la capacitación del personal destinado para su aplicación como la disciplina que ello exige. En la Figura 9 se muestran gráficamente los tiempos invertidos para ambos proyectos.

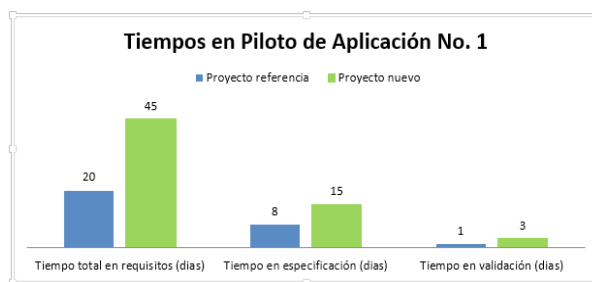


Fig. 4. Comparación de tiempos del piloto de aplicación N.º 1

Lo anterior tiene incidencia directa sobre el número de defectos encontrados en la verificación de la especificación, pues al ser revisada por el responsable de los requisitos inmediatamente después de haber sido elaborada, y mediante la aplicación del formato sugerido por el procedimiento, se logró detectar mayor número de defectos, para este caso, 60 % más.

Asimismo, al haber hallado mayor número de defectos en la verificación, se redujeron en un 33 % los defectos encontrados en la validación realizada con el cliente, lo que a su vez generó una disminución, en un 75 %, en los errores encontrados en producción atribuibles a los requisitos. La Figura 10 refleja el impacto favorable para el proyecto por la detección significativa de errores en etapas tempranas.

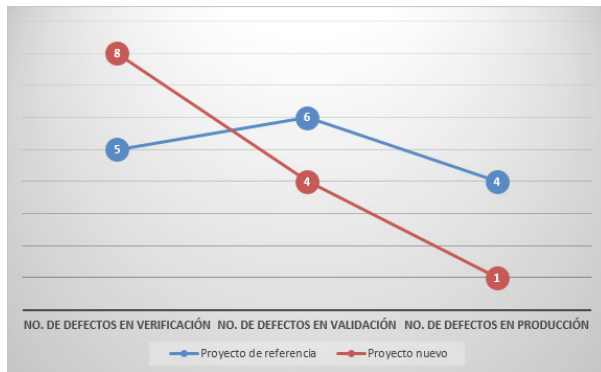


Fig. 5. Defectos encontrados en ambos proyectos

Se concluye, igualmente, que si bien para la aplicación del nuevo procedimiento fue necesario emplear más tiempo, se considera que el esfuerzo fue recompensado por el hallazgo de mayor número de defectos en etapas tempranas, lo que en definitiva constituye una ganancia respecto al costo de corregir dichos errores en etapas posteriores.

Finalmente, antes de usar el procedimiento elaborado, se aplicó una encuesta en la que se determinaron las expectativas que tenían las personas responsables de su aplicación en ambas empresas, como también una encuesta después de la aplicación con el objetivo de determinar el rendimiento percibido, es decir, el “resultado” que los equipos percibieron una vez aplicado el procedimiento, con el ánimo de conseguir el nivel de satisfacción que equivale a la sustracción entre el rendimiento percibido y las expectativas.

### A. Análisis de Consistencia de las Encuestas de Expectativas y Rendimiento

Para estimar la fiabilidad de las encuestas de expectativas y de rendimiento se utilizó un procedimiento que permite calcular la confiabilidad y validez de los instrumentos denominado Alfa

de Cronbach [13], el cual indica que la validez se refiere al grado en que el instrumento mide lo que se pretende calcular y la confiabilidad hace referencia a la confianza que se concede a los datos.

Como se indica en la Figura 11, un Alfa de Cronbach superior a 0,8 indica que existe fiabilidad en el instrumento que se aplicó, mientras que si no supera el 0,6 se habla de un instrumento inconsistente y posiblemente inestable.



Fig. 6. Nivel de consistencia Alfa de Cronbach [13]

Para determinar el Alfa de Cronbach de ambas encuestas se aplicó el método Varianza de los Ítems mediante la siguiente fórmula:

$$a = \frac{K}{K-1} \left[ 1 - \frac{\sum V_i}{V_t} \right]$$

Donde:

- $\alpha$  = Alfa de Cronbach
- K = Número de ítems (preguntas)
- $V_i$  = Varianza individual de cada ítem
- $V_t$  = Varianza total

Dado que el resultado del Alfa de Cronbach obtenido fue de 0,81 para la primera encuesta, y de 0,87 para la segunda, se considera que existe evidencia de que los elementos miden el mismo constructo, es decir que las preguntas formuladas apuntan a determinar las expectativas de los encuestados de ambas empresas y las respuestas permiten medir de forma adecuada lo que esperan del procedimiento PEVReS.

Asimismo, el procedimiento se modeló bajo SPEM utilizando el Eclipse Process Framework Composer para una mejor presentación y despliegue. A este procedimiento se le dio el nombre de Procedimiento para la Especificación y Validación de Requisitos de Software (PEVReS).



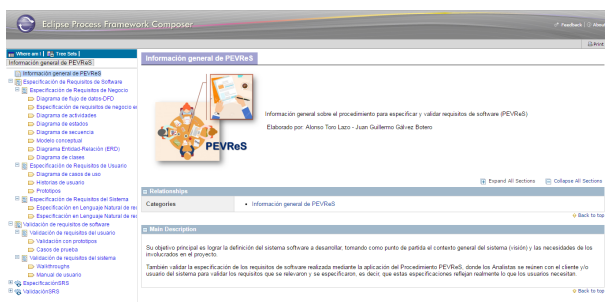


Fig. 7. Información sobre el PEVReS

De acuerdo al seguimiento realizado a la aplicación del procedimiento en los pilotos de aplicación, se tienen en cuenta las dificultades presentadas y las mejoras sugeridas para hacer los ajustes al procedimiento de especificación y validación de requisitos y para dar cumplimiento a la Etapa 4 de la metodología del proyecto, relacionada con el actuar respecto al ciclo PHVA. Los ajustes al procedimiento propuesto se hicieron de acuerdo a las dificultades y mejoras obtenidas durante la verificación de los pilotos de aplicación en ambas empresas (Etapa 3 de la metodología). Los ajustes realizados se muestran en la Tabla 3.

Tabla 8. Ajustes al Procedimiento de Especificación y Validación

TIPO*	DESCRIPCIÓN	ACCIÓN
M	Se sugiere poner espacio para firmas tanto del usuario como del ingeniero encargado, en los formatos del procedimiento para la validación.	Implementada
D	No se comprende claramente la descripción de la fuente de los datos de entrada de los requisitos funcionales presentada en el formato del Documento SRS.	Se cambia la redacción

TIPO*	DESCRIPCIÓN	ACCIÓN
M	Se recomienda que no existan especificaciones en lenguaje natural de requisitos de usuario, sino más bien que dichas especificaciones sean los prototipos, casos de uso o historias de usuario.	Implementada
D	Existe ambigüedad en el nombre de la columna “Descripción de la tarea” en el formato para la validación de prototipos.	Se cambia la redacción
D	No se encuentra conexión entre las dos actividades presentes en el procedimiento para la validación de requisitos.	Se cambia el nombre de la actividad N.º 2 del procedimiento por “Validación de Requisitos del Sistema”
M	En las entradas del procedimiento para validar requisitos no sería necesario incluir los prototipos elaborados, ya que éstos están incluidos en el Documento SRS realizado.	Implementada
M	En los responsables de ejecutar la técnica “Casos de prueba” del procedimiento para validar requisitos, debería incluirse una persona del equipo de pruebas.	Implementada
M	Se sugiere que la técnica para especificar requisitos de funcionales en lenguaje natural sea requerida para garantizar que los mismos sean incluidos en el Documento SRS.	Implementada

TIPO*	DESCRIPCIÓN	ACCIÓN
M	Se recomienda que antes de ejecutar el procedimiento para la validación de requisitos se tenga como precondition haber realizado una planificación de la validación.	Implementada
M	El procedimiento propone la verificación de la especificación realizada por parte de quien la desarrolló, sería interesante que además dicha verificación fuera hecha por otra persona (Inspección).	Implementada
D	El paso 5 de la técnica prototipos "Verificar el prototipo con el cliente e identificar otros requisitos (Evaluación)" presenta un error debido a que la verificación no debería llevarse a cabo con el cliente, sino sólo por el responsable de la técnica.	Se corrige la inconsistencia
D	El paso 7 de la técnica prototipos "Realizar la documentación necesaria para su implementación", es ambiguo, pues no especifica a qué documentación hace referencia.	Se elimina dicho paso, pues se considera innecesario
D	El procedimiento de validación de requisitos no muestra una secuencia lógica que permita relacionar las actividades propuestas, pues la Actividad N.º 2 puede contener la Actividad N.º 1.	Se corrige la presentación de las actividades para que haya una secuencia entre ellas.

Donde, D: Dificultad, M: Mejora

#### IV. DISCUSIÓN DE RESULTADOS

La selección de las técnicas del marco teórico, de los antecedentes y de las usadas por las Mipymes desarrolladoras de software de la ciudad de Pereira que fueron analizadas como insumo principal para el diseño del procedimiento PEVReS, se hizo en consenso entre los autores del proyecto y el asesor a través de juicio de expertos, se identificaron además sus principales ventajas y desventajas con la intención de argumentar su pertinencia en el procedimiento. Mediante este análisis se lograron identificar los elementos de proceso relacionados con las actividades, técnicas, roles, guías, etc., requeridos para el procedimiento de especificación y validación de requisitos de software.

Por otro lado, la etapa de la metodología, en la cual se diseñó el procedimiento, tuvo como resultado un procedimiento estructurado a través de una secuencia de actividades puntuales que establecen el uso de técnicas soportadas en los referentes teórico y contextual y en el diagnóstico de las empresas. El procedimiento estructurado proporciona además un conjunto de guías, formatos y herramientas que lo diferencian de otras propuestas, y que permiten a la persona encargada de especificar los requisitos conseguir una definición correcta de ellos en un documento SRS alineado con el procedimiento, basado en estándares mundialmente reconocidos y validado por el usuario. Por consiguiente, se da por cumplido el tercer objetivo del proyecto, cuyo principal resultado fue el diseño del procedimiento que incluye una especificación en SPEM.

Con relación a la etapa en la que se realizó la verificación de los datos obtenidos una vez aplicado el procedimiento, se observa un aumento en la identificación de defectos en etapas tempranas, y una reducción de defectos en etapas posteriores, lo que no solo representa una mayor calidad en el producto final, sino una disminución de costos y de tiempo en correcciones y reprocesos, y de ese modo se compensa el tiempo de más invertido en la especificación y validación de requisitos.

Igualmente, la ejecución de esta etapa obtiene como resultado una serie de mejoras y recomendaciones en el procedimiento para aplicaciones futuras, producto de las dificultades encontradas durante su

aplicación, como también niveles de satisfacción de 6 y 5 (satisfecho) en cada una de las empresas, respectivamente, resaltando que con la aplicación del procedimiento no solo se lograron mejores resultados en el proceso de desarrollo de los requisitos, sino también satisfacción en el uso del procedimiento por parte de los desarrolladores. Tanto la encuesta de expectativas como la de rendimiento, que ayudaron a determinar los niveles de satisfacción respecto al uso del procedimiento, fueron analizadas mediante el Alfa de Cronbach para estimar la confiabilidad y validez de los instrumentos.

De la misma manera, en la cuarta etapa se logró obtener un procedimiento ajustado a las empresas en las que se llevaron a cabo los pilotos de aplicación. Se resalta que a pesar de todo lo nuevo propuesto en el procedimiento, se presentaron pocos errores durante su aplicación y las recomendaciones muestran apropiación del proceso por parte de las empresas. Sin embargo, se considera que al aplicarse en otras organizaciones pueden surgir ajustes adicionales. Como mejora adicional se modeló el procedimiento a través de SPEM para su mejor presentación y despliegue, aunque su aplicación en las empresas hará parte del trabajo futuro.

## V. CONCLUSIONES

En la investigación sobre especificación y validación de requisitos de software en Mipymes desarrolladoras de software de la ciudad de Pereira, basado en estudios previos en la región, se pueden establecer las siguientes conclusiones:

- A partir de los resultados obtenidos con la aplicación de la encuesta, se logran conocer aspectos importantes sobre el proceso de requisitos en las empresas desarrolladoras de software de la ciudad de Pereira, lo cual, junto con el análisis de las técnicas obtenidas del referente contextual, sirvió de base para el desarrollo del procedimiento, pero también podría servir para otros estudios y trabajos futuros.
- El procedimiento se logró validar mediante la aplicación de dos pilotos de aplicación en dos Mipymes desarrolladoras de software de la ciudad

de Pereira, y se consiguió que ambas refinaran su proceso de definición y validación de requisitos.

- La aplicación del procedimiento en las empresas participantes de la validación permitió que los equipos responsables del desarrollo de los requisitos alcanzaran niveles de satisfacción que dan cuenta de los buenos resultados obtenidos en sus proyectos.
- Para los dos pilotos de aplicación llevados a cabo, la aplicación del procedimiento logró encontrar errores en etapas tempranas mediante el uso de las técnicas que él mismo propuso, lo cual contribuyó a mejorar la calidad del producto y que fuera más cercano a los requisitos.
- Uno de los factores de éxito en la aplicación del procedimiento fue el uso de las técnicas, guías y formatos que apoyaron la especificación y validación de los requisitos, de tal manera que el esfuerzo de los miembros del equipo se centró en la ejecución de los diferentes pasos que propone cada técnica definida en el procedimiento.
- Al encontrar la mayor cantidad de defectos en el desarrollo de los requisitos se reduce notablemente el número de defectos encontrados en la etapa de producción atribuibles a los requisitos, aunque se impacte el tiempo empleado en el desarrollo.
- En los proyectos intervenidos mediante los pilotos de aplicación siguen existiendo defectos en la etapa de producción atribuibles a los requisitos, por lo que resulta de importancia que las Mipymes puedan refinar o madurar el procedimiento para la validación de requisitos.
- Finalmente, y de acuerdo a la discusión de resultados presentada, se considera que los objetivos propuestos por el proyecto se alcanzaron satisfactoriamente y contribuyeron a mejorar procesos en las empresas en las cuales se aplicó el procedimiento, gracias a su alineación con cada una de las etapas de la metodología.

## REFERENCIAS

- [1] IEEE, SWEBOK Guide V3.0, Piscataway: IEEE, 2014.
- [2] K. Wiegers y J. Beaty, Software Requirements, 3.<sup>a</sup> ed., Redmon, Washington: Microsoft Press, 2013.
- [3] H. G., Salazar Osorio, H. J. Rengifo Romero, L. E. Machuca Villegas y J. A. Aranda Bueno, “Una guía general para la especificación y verificación formal de requerimientos usando Event-B<sup>TM</sup> y Rodin<sup>TM</sup>”, Revista Educación en Ingeniería, vol. 7, N.º 14, pp. 82-91, 2012.
- [4] F. García Ramírez y P. Puello Marrugo, “Gestión de Requisitos en la Ingeniería del Software”, Ingeniator | Revista Virtual de los Programas de Ingeniería|Universidad de San Buenaventura, vol. 1, N.º 1, pp. 57-65, 2010.
- [5] A. Toro Lazo y J. G. Gálvez Botero, Procedimiento para especificar y validar requisitos de software en MiPymes desarrolladoras de software de la ciudad de Pereira, basado en estudios previos de la región. Manizales: Universidad Autónoma de Manizales, 2017.
- [6] A. Toro Lazo y J. G. Gálvez Botero, Especificación de requisitos de software: una mirada desde la revisión teórica de antecedentes. Entre Ciencia e Ingeniería, vol. 10, N.º 19, pp. 108-113, 2016.
- [7] M. Cristiá, Introducción a la Ingeniería de Requerimientos, Universidad Nacional de Rosario, 2011. Disponible en: <http://www.fceia.unr.edu.ar/~mcristia/publicaciones/ingreq-a.pdf> [Acceso el 20 de junio de 2014].
- [8] The Standish Group, CHAOS Report, Project Smart, 2014.
- [9] L. E. Peláez Valencia, Propuesta metodológica para estandarizar el proceso de construcción y evaluación del producto software que permita a las PyMES medir la calidad del software, Pereira, 2011.
- [10] Departamento Administrativo Nacional de Estadística (DANE), “Resolución N.º 066 del 31 de Enero de 2012”, Bogotá: DANE, 2012. Disponible en [https://www.dane.gov.co/files/sen/nomenclatura/ciiu/normatividad/Resolucion066\\_2012.pdf](https://www.dane.gov.co/files/sen/nomenclatura/ciiu/normatividad/Resolucion066_2012.pdf)
- [11] V. H. Menéndez Domínguez y M. E. Castellanos Bolaños, “SPEM: Software Process Engineering Metamodel”, Revista Latinoamericana de Ingeniería de Software, pp. 92-100, 2015.
- [12] Object Management Group (OMG), Software Process Engineering Metamodel (SPEM) Specification version 1.0, Object Management Group, 2008. Disponible en <http://www.omg.org> [Acceso el 26 de diciembre de 2016].
- [13] Bioestadístico, Alfa de Cronbach, índice de consistencia interna, Bioestadístico, dic. 2014. Disponible en <http://bioestadistico.com/> [Acceso 15 de dic. de 2016].
- [14] <http://www.pdcahome.com/5202/ciclo-pdca/>





## Capítulo XIV

# MANEJO DE INFORMACIÓN GEOGRÁFICA A TRAVÉS DE HERRAMIENTAS INTEGRADAS. CASO DE ESTUDIO: CENSO ARBÓREO MUNICIPIO DE TULUÁ

*Beatriz Eugenia Marín - bmarin@admon.uniajc.edu.co*

Docente, Universidad Antonio José Camacho de Cali

*Antonio Pantoja - Antonio.pantoja@team.unigis.net*

Docente,

*Edwin Núñez Ortiz - enunez@admon.uniajc.edu.co*

Docente, Universidad Antonio José Camacho de Cali

*Daniel Ernesto Osorio - danielosorio@geoprocess.com.co*

Docente, Geoprocess

*Miguel Andrés Idrobo - miguelidrobo@geoprocess.com.co*

Docente, Geoprocess

## I. INTRODUCCIÓN

Anteriormente, el tratamiento de la información geográfica (IG) estaba reservado para expertos en el área de la geografía y su tratamiento se enfocaba en necesidades particulares, por lo regular para uso de organismos públicos de agricultura, ayuntamientos y, en general, el uso de suelos [1]. Sin embargo, hoy en día, con los avances de la tecnología y el amplio acceso a la conectividad, la información, y en especial la IG son de ocupación común, y se hace necesario garantizar su proceso de recolección de datos, tratamiento, análisis y exposición de una manera eficiente. Los métodos convencionales para la gestión IG implican altos costos porque necesitan convertir información análoga en datos digitales e inversión en equipos especializados que garantizan alta precisión en sus resultados, pero son de difícil acceso para proyectos que cuentan con bajos recursos. En este documento se comparan ventajas y desventajas de utilizar métodos convencionales con herramientas integradas frente a las que usualmente se utilizan como GPS de precisión submétrica, formularios en papel, etc., y se muestra que de acuerdo al proyecto, sus características y requerimientos es viable seguir cualquiera de las dos opciones. Se ha tomado como referencia un caso práctico ambiental.

El documento está organizado de la siguiente manera: en primer lugar, se introducen los conceptos generales de los sistemas de información geográfica (SIG), en la segunda parte se describe el caso del censo arbóreo realizado por la empresa GeoProcess en el municipio de Tuluá utilizando una herramienta integrada, y como tercero y último elemento se discuten los resultados frente a las herramientas convencionales y se concluye su viabilidad en cada caso.

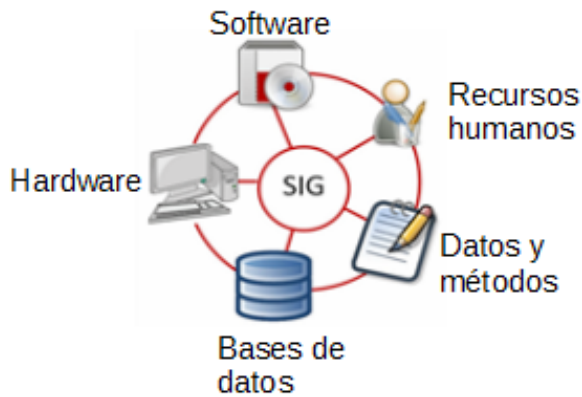
## II. CONCEPTOS PRINCIPALES SOBRE SIG

A continuación se presentan los principales conceptos que enmarcan el desarrollo y tratamiento de la información y los procesos en un sistema de información geográfico:

### A. Sistemas de Información geográfica (SIG)

Según [2] un sistema de información geográfica (SIG) es un sistema de información donde los datos se encuentran georreferenciados con una posición en un sistema de coordenadas. Estos datos pueden ser almacenados, consultados y analizados y apoyar la toma de decisiones.

**Componentes de un SIG:** Los componentes principales de un SIG son hardware, software, datos espaciales, métodos y recursos humanos (ver Figura 1).



**Fig. 1.** Componentes de un SIG [3]

**Datos espaciales:** Por lo general los datos de entrada están referenciados geográficamente (aunque algunos pueden no estarlo necesariamente). Es ideal que un SIG sea alimentado o actualizado con frecuencia, también debe permitir convertir los datos en información suficientemente clara para que los usuarios puedan tomar decisiones pertinentes, lo cual es posible solamente si la información es veraz, precisa y actual. El Comité Técnico Geomatics (ISO TC/211) fue el encargado de estandarizar la información geográfica, asimismo, el Open Geospatial Consortium (OGC) realiza la promoción y divulgación de estándares y tecnologías en un esfuerzo por mantener la homogeneidad de la información. Los métodos se encargan de guiar la construcción de los elementos de diseño, planeación y ejecución del SIG que van acompañados de las prácticas propias de la organización propietaria del sistema, y se determinan de acuerdo a las características de los datos.

**Hardware:** Son los equipos necesarios para ejecutar el SIG, incluyen desde servidores y equipos de escritorio hasta herramientas de visualización como plotter, impresoras, etc.

**Software:** son aplicaciones computacionales como:

- Una base de datos espacial, como Oracle Spatial o PostgreSQL + PostGIS (software libre), entre otros.

- Un servidor de mapas como Mapserver, GeoServer, GeoNetwork o ArcGIS Server.
- Servidores de aplicaciones, por ejemplo JBOSS.
- Librerías para el visor geográfico: Leaflet (anteriormente Open Layers).
- Motores para gestión de bases de datos geográficas: por ejemplo ARCGIS SDE, y lenguajes como Java, JavaScript.

**Bases de datos:** Contienen datos espaciales y no espaciales, así como las relaciones entre ellos. Pueden ser de diferentes fuentes internas o externas de la organización.

**Recursos humanos:** Es el personal que opera y administra el sistema y lo lleva al mundo real. También se incluye en este grupo a los usuarios finales que interactúan con los resultados finales del SIG [4].

**Ciclo de vida de un SIG:** El desarrollo de un SIG se enmarca en los procesos generales para la producción de software tales como análisis, planeación y diseño, implementación y despliegue y se adecua a las metodologías existentes, entre ellas RUP, XP, Scrum, Iconix, etc.

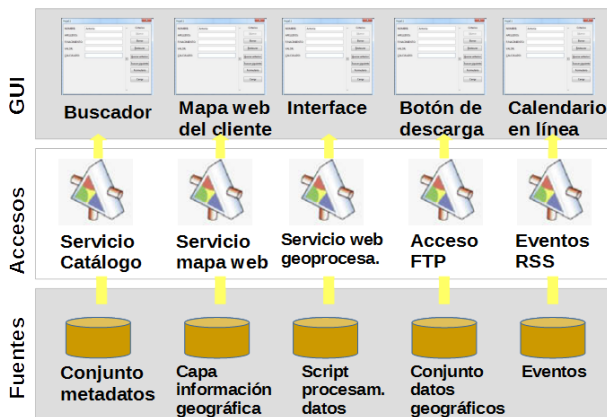
En las fases de análisis, planeación y diseño se incluyen las tareas de modelado, procesos de integración de datos de fuentes externas y su conversión en datos espaciales. En este punto se define el cálculo de la precisión y veracidad de los datos dependiendo de su tipo y del análisis que se quiere realizar. Cabe destacar que el análisis de los datos está ligado a los modelos de procesamiento, que pueden ser operaciones entre tablas o funciones de geoprocamiento. Estas funciones pueden estar disponibles en software SIG de escritorio o en SIG en la nube. Las funciones están relacionadas con operaciones numéricas, estadísticas, estadísticas espaciales, operaciones espaciales y consultas entre tablas.

En la implementación se desarrollan, integran y prueban los componentes de software. En el despliegue se prepara la publicación de la información, la cual está relacionada con la visualización de los datos, por lo regular a través de geoportales o Infraestructura de datos espacial (IDE). También en esta etapa se capacita el personal y se realizan los mantenimientos que sean necesarios.

## B. Geoportales

Un geoportal es una plataforma web donde se combinan los SIG y las tecnologías web para brindar soluciones personalizadas a problemas de naturaleza espacial. Con la posibilidad de brindar acceso en diferentes niveles: administrativo, de agregación y modificación de la información, o solamente de consulta. Los geoportales son las puertas de enlace en el www a la información geográfica (Figura 2) [6].

**Infraestructura de Datos Espaciales (IDE);** Una IDE se puede definir como “una colección relevante de tecnologías, políticas y arreglos institucionales que facilitan la disponibilidad y el acceso a los datos espaciales” (GSDI-Cookbook), lo cual se realiza a través de la web con información proveniente de diferentes fuentes, a fin de proveer al usuario información estandarizada para consultar, descargar o realizar superposición y análisis visual o avanzado con información propia o de otros portales web (que pueden ser de otras IDE o no) [3], [6].



**Fig. 2.** Ejemplo de recursos provistos a través de una arquitectura de un geoportal orientado al servicio [6].

**Herramientas para disponer información a través de una IDE;** Los geoportales para la IDE, a diferencia de otras aplicaciones de mapas en línea (geoportales), aseguran que la información sea de calidad y esté actualizada, según las políticas de aseguramiento de calidad y mantenimiento de información geográfica (IG) que, idealmente, debieran establecer las instituciones integradoras de la IDE.

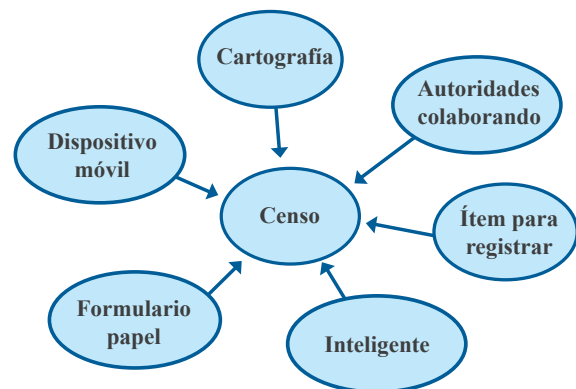
En lo que respecta a los elementos técnicos, uno de los componentes con los que debe contar la IDE es la

posibilidad de utilizar servicios de datos, los cuales deben tener estándares de calidad y ajustarse a normas de calidad a nivel institucional y de gobierno. En los últimos años la toma de datos geográficos e IG se ha popularizado gracias a la conectividad que permite internet, y que ha dado origen a la información geográfica voluntaria (IGV) que se distribuye en servicios como Open Street Map y Wikimapia [7].

**Las aplicaciones en la nube:** La computación en la nube ha permitido el acceso de organizaciones y particulares a datos y servicios con bajo costo, los geoportales unifican en un solo lugar esta información y herramientas para diferentes áreas. Por ejemplo, sobre el geomarketing los geoportales permiten desplegar información de ubicación de clientes, de la competencia, de rutas de distribución, etc. [8]. En el sector ambiental y climático los geovisores permiten a los usuarios descargar datos espaciales y alfanuméricos de distintos satélites y estaciones meteorológicas, y descargar además pronósticos del tiempo [9].

Existen también aplicaciones que cuentan con doble función: recolectar información georreferenciada y disponerla en tiempo real para los mismos usuarios, a diferencia de los geovisores convencionales que muestran información procesada previamente.

Los SIG en los procesos de censo: El registro en un censo puede ser respaldado por un mapa presentando los resultados de manera cartográfica para facilitar su divulgación y análisis. Adicionalmente, los mapas permiten la reunión de datos y su supervisión para poder garantizar la consistencia y exactitud de la información [5].



**Fig. 3.** Componentes de gestión de un censo [10].

La utilización de un SIG dentro de una empresa implica una adecuación de la infraestructura tecnológica y en algunos casos también de tipo organizacional. En la Tabla 1 se listan algunas de las tareas sugeridas en el manual de cartografía digital de la ONU que se enfoca en proyectos de censo:

**Tabla 1.** Tareas Necesarias para Empezar a Usar los SIG [5]

Tarea	Descripción
Planificación y diseño de sistemas, servicios de consultoría, tiempo del personal administrativo, incluye también: Integración o adquisición de equipos informáticos. Evaluación y selección de los SIG/programas de elaboración de mapas.	Es la planificación general del proyecto que quiere hacer uso de un SIG, incluyendo la definición de computadores, periféricos y programas de tratamiento de datos y visualización.
Desarrollo de prototipos.	Programas piloto con población reducida que permita validar cualquier procedimiento del SIG que deba ser ajustado antes de llevarlo a la población total.
Configuración/adaptación de los sistemas de equipos y programas informáticos.	Establecer las interfaces y redes que se necesiten para interactuar con el SIG.
Planificación de los recursos humanos.	Estimar el recurso humano requerido para hacer uso del SIG.
Capacitación, calificación profesional, readiestramiento.	Preparación del personal existente.
Diseño de bases de datos, preparación de manuales.	Modelos conceptuales, lógicos y físicos de la BD que recibirá los datos y se integrará con el SIG.
Compra/adquisición de datos	Identificar los datos requeridos que corresponden a fuentes comerciales.
Control/garantía de calidad, validación	Determinar los procedimientos rigurosos de comprobación de los datos reales que se obtendrán.

### III. CASO: CENSO ARBÓREO DEL MUNICIPIO DE TULUÁ

Los censos arbóreos urbanos se establecen en los decretos de silvicultura urbana a partir del año 2013 como mecanismos de control y protección de la vegetación dentro de las ciudades, y deben ser ejecutados por los municipios o las empresas prestadoras de servicios de aseo y podas. A causa de esta obligatoriedad, las municipalidades se enfrentan al reto de poder realizar esta actividad con recursos limitados.

El método convencional para realizar los censos arbóreos es el siguiente:

- Formularios largos en papel, lápiz, cámara fotográfica, maleta para llevar todos los implementos descritos.

- Comisiones de campo, normalmente conformadas por cuatro personas:

- Ingeniero forestal (encargado de caracterizar las especies arbóreas).
- Un operario (diligencia el formulario en papel, toma fotografías y carga equipaje)
- Topógrafo (opera el equipo GPS donde tomará registros durante mínimo 20 minutos).
- Dos operarios (auxiliares de topografía) que se encargan de caracterizar cada especie, y consignar sus atributos en un formulario de papel y fotografía en archivo digital.

- La georreferenciación de cada árbol debe ser submétrica, es decir, con una precisión máxima. Para ello se requieren equipos GPS robustos para obtener precisiones de (+/-) 50 cm por cada comisión; una base GPS que tome datos en placas establecidas por los institutos geográficos nacionales y un postprocesamiento llevado a cabo por ingenieros topográficos o geodésicos que garanticen la exactitud posicional por cada árbol.
- Conformar un gabinete de técnicos que digite los datos de formularios en papel empleando hojas de cálculo para posteriormente generar bases de datos. Otro personal debe construir y administrar la base de datos y enlazar dentro de cada id-árbol las fotografías y las coordenadas geográficas adquiridas con dispositivos independientes.

El municipio de Tuluá, dando cumplimiento a las leyes gubernamentales, contrató en el año 2016 a la empresa Geoprocess ubicada en la ciudad de Cali, para llevar a cabo el censo arbóreo de la ciudad.

### IV. MATERIALES Y MÉTODOS

Para la construcción de la propuesta inicialmente se validó la información geográfica disponible y se verificó la viabilidad de soportar el censo con un sistema geográfico. Posteriormente se plantearon las siguientes hipótesis orientadoras del proyecto:

- Disminuir el tiempo de recolección de datos desenlaza en menos tiempo en campo, menos riesgo y menos recursos financieros destinados a este ítem.



- Utilizar herramientas de posicionamiento GPS, junto con los registros por árbol y la fotografía permitirá reducir el personal, disminuir la probabilidad de error en los datos y la probabilidad de riesgos, accidentes y costos de operación en campo.

Con el propósito de desarrollar cada hipótesis se utilizaron las siguientes estrategias:

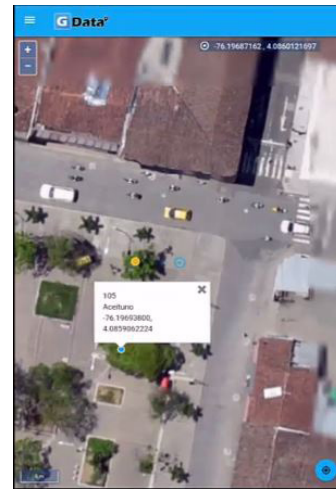
**Disminuir el tiempo de recolección de datos:**

Para ello se empleó la herramienta GisData[11] de propiedad de la empresa Geoprocess, que permite, de acuerdo a un formulario preestablecido, diseñar un formulario digital que agilice el trabajo de campo, por ejemplo:

- Nombre de barrios y comunas preestablecidos y evitar errores en la base de datos.
- Condiciones fitosanitarias y de edad del árbol empleando campos de selección múltiple, selección única y menús desplegables.
- Para el caso de seleccionar las especies, familias y géneros de cada árbol, estas superan las 200 opciones, por lo que se construyó el autocompletable, que permite realizar una búsqueda entre un diccionario de datos preestablecido.

**Disminuir el personal:** Al no requerirse equipos GPS submétricos y base GPS, el personal se resumió en el técnico forestal y un auxiliar (opcional). No se necesitó personal en oficina para digitalización de datos, ni organización de base de datos.

Articular el posicionamiento GPS con los registros por árbol y la fotografía: GisData incorporó la ortoimagen de muy alta resolución de precisión submétrica, lo cual permitió visualizar la copa de árboles y posicionar una marca para obtener una coordenada con la exactitud posicional requerida para estos proyectos. Además permitió incorporar una imagen al ID del árbol censado con un registro fotográfico (Figura 4).



**Fig. 4.** Ortoimagen con etiqueta. Geoprocess [11]

**Disminuir la probabilidad de error de los datos:**

No fue necesario digitalizar la información análoga, por tanto se disminuyó el error por transcripción de datos. Asimismo, al no ser necesario enlazar el nombre de una fotografía y el id de la base de datos se disminuyó la probabilidad de error de que un id-árbol registre una fotografía que no corresponda.

Igualmente, no fue necesario emplear GPS submétrico ni métodos de postprocesamiento, lo cual ahorró tiempo en obtener la coordenada geográfica en campo, en menos de un minuto, y no fue necesario procesamiento en oficina. Además se disminuyó la probabilidad de obtener georreferenciación errónea.

**Disminuir los costos de materiales:** Para tomar datos no se requirió comprar:

- Lápiz y papel
- Maleta
- Cámara fotográfica
- Equipo GPS
- Tabla

**Disminuir la probabilidad de riesgos, accidentes**

**y costos de operación en campo:** Al disminuir el personal de campo y los tiempos de recolección de datos, se disminuyó la probabilidad de accidentes y el número de seguros para los operadores, y se simplificó además la logística de campo y los costos de operación



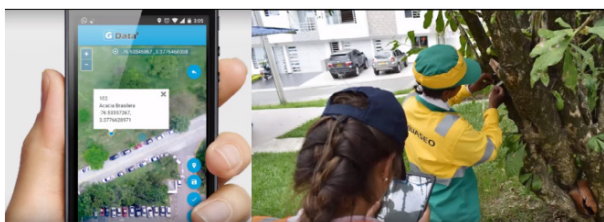


Fig. 5. Censo arbóreo en Tuluá – Geoprocess [11]

**A. Procedimiento**

El procedimiento consistió en realizar un vuelo fotogramétrico con aviones pequeños o drones, que además contó con puntos de control en tierra para garantizar una exactitud posicional entre 30 y 50 centímetros.

Se emplearon dispositivos móviles inteligentes con sistema operativo Android, el cual involucró las siguientes funcionalidades:

- Formulario digital.
- Geovisor que despliega la ortoimagen para posicionar coordenada geográfica submétrica.
- Geovisor que despliega los puntos ya tomados y que pueda consultar la información ya recogida.
- Tomar registro fotográfico y anidarlos al registro por árbol.
- Sincronización a la nube de los datos recogidos.

Se utilizó una plataforma web en la cual se pueden visualizar mediante un geovisor los datos recogidos en campo en tiempo real. La plataforma permitió consultar tablas, donde se pudo observar el rendimiento por operador, el estado fitosanitario de las especies arbóreas y su distribución espacial, los árboles que requieren podas, entre otras consultas geográficas.

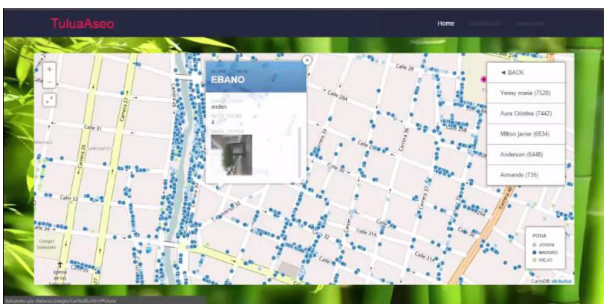


Fig. 6. Visualización de datos del censo. Geoprocess [11]

Además, la plataforma web permitió generar estadísticas, gráficos y tablas cruzadas para agilizar el proceso de reportes e informes (Figura 7).

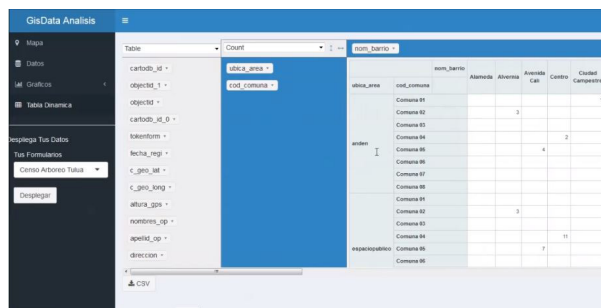


Fig. 7. Muestra de tabla dinámica generada con los datos del censo. Geoprocess [11]

**V. RESULTADOS**

En el proyecto de censado arbóreo de Tuluá se levantaron 28.000 registros sobre un área de 29 kilómetros cuadrados, con seis operarios en campo y cuatro tabletas, logrando registrar 110 árboles por día.

**Tabla 2.** Método Convencional frente a una Herramienta Integrada para el Caso del Censo Arbóreo de Tuluá

Ítem	Convencional	GisData
Número de operarios en campo	20	6
Número de equipos GPS submétricos.	5	0
Orto-imagen con exactitudes posicionales submétricas.	0	1
Papelería-maletín	20	0
Número de digitadores en oficina	20	0
Tablets	0	4
Software licenciado GIS y de procesamiento GPS	2	0
Número de árboles censados por día	80	110

Fue evidente la reducción en el costo del proyecto así como el rendimiento del trabajo de campo y procesamiento de la información.

**VI. DISCUSIÓN**

En los procesos convencionales de censos que utilizan SIG os problemas más comunes que se presentan son los siguientes:

- El tiempo que se toma un operador en diligenciar formularios en papel.
- El costo de contar con personal amplio en campo, ya que se requieren seguros, logística, indumentaria, etc.
- La desarticulación entre los formularios de papel, la coordenada geográfica y la fotografía.
- Los errores en la exactitud posicional de la coordenada, debido a que la señal GPS se ve obstruida por la misma copa de los árboles y edificios adyacentes, que ocasionan que el valor de la coordenada quede posicionado en la mitad de la vía o en una manzana que no corresponde.
- Los GPS de precisión submétrica requieren al menos 20 minutos por árbol para obtener la precisión adecuada, eso significa mayor tiempo en campo y un incremento en los costos del proyecto.
- Los errores en la digitalización de la información, ya que el digitador en ocasiones interpreta con dificultad lo plasmado por el operador de campo sobre el papel.
- La alta probabilidad de error en la integración en la base de datos de la fotografía exacta de la especie arbórea.
- El alto costo del personal digitalizador de datos. Las probabilidades de inconvenientes en campo son mayores que los procesos en oficina, por la inseguridad, el clima y los accidentes.

Los problemas al utilizar una herramienta integrada para el censo son los siguientes:

- Se pierde autonomía y control sobre cada uno de los procesos de recolección, procesamiento y despliegue, ya que están integrados y definidos previamente en la herramienta.
- En algunos casos, según la herramienta, los costos de soporte son muy elevados.
- Las evaluaciones y validaciones de datos algunas veces son más complicadas porque no pueden realizarse por módulos.

Bajo este panorama, las herramientas integradas permiten reducir los costos invertidos en el proyecto con resultados en corto tiempo, y para proyectos que requieren módulos específicos ajustados a las características de la organización, las herramientas integradas limitan el acceso y generan dependencia de las casas desarrolladoras.

Cada proyecto en particular debe evaluar la viabilidad económica, en tiempo y en personal requerido para tomar una decisión adecuada que le permita obtener resultados efectivos.

## VII. TRABAJOS FUTUROS

Algunos autores como [12] definen una nueva línea en las TIC denominada tecnologías de información geográfica (TIG), las cuales se especializan en recolección, análisis, procesamiento y visualización de información geográfica en entornos naturales y sociales, incluyendo herramientas como los SIG, los sistemas de posicionamiento global, la teledetección y las herramientas de internet.

Entre las tendencias que se encuentran en esta línea está el uso de redes neuronales para permitir la generación de información que no se encuentra explícita [13].

Otro aspecto importante que impulsa el desarrollo en TIG es la web 2.0 que ha convertido a cada persona en sensores humanos con dispositivos móviles, lo que permite obtener información en tiempo real y desde cualquier parte del planeta así como tecnologías de interacción avanzadas, como la realidad aumentada.[14]

## VIII. CONCLUSIONES

Las herramientas tecnológicas integradas permiten disminuir costos en procesos propios del desarrollo del SIG, ahorran tiempo y recursos humanos, pero su aplicación está sujeta a las características propias del proyecto.

Cuando son proyectos a largo plazo, cada tarea que se desarrolle requerirá actualización e intervención periódica, lo cual genera dependencia de la casa desarrolladora de la herramienta, por lo tanto, se debe contar con los recursos suficientes, o si se decide utilizar un método desintegrado se debe contar con los expertos en cada tema dentro del equipo de trabajo.

Cuando son proyectos a corto plazo cuyos resultados se utilizarán para apoyar la toma de decisiones, una herramienta integrada es una buena opción teniendo en cuenta que se contaría con los aspectos ya definidos

en cada área importante del proyecto. La herramienta provee lo necesario para generación de formularios, análisis de la información, entornos de visualización, etc.

Es importante, además, para el desarrollo de los proyectos, que el manejo de la información se realice en formatos compatibles que garanticen su comunicación, donde las herramientas integradas brindan el soporte necesario para evitar inconvenientes y pérdidas de información y precisión.

La vinculación de un equipo interdisciplinar de expertos para el desarrollo de proyectos para la utilización de SIG es también sinónimo de garantía en la calidad y profundidad con la que se emplean herramientas de análisis y desarrollo, pero como se comentó anteriormente, requiere el tiempo suficiente para resolver las inconsistencias e incompatibilidades que se presenten en el desarrollo de las tareas y procesos al utilizar herramientas desintegradas.

## REFERENCIAS

- [1] E. López, C. Posada y J. Moreno, “Los sistemas de información geográfica”, en I Congreso de Ciencia Regional de Andalucía, Andalucía, 2004.
- [2] F. Sarria, “Sistemas de información geográfica”, Universidad de Murcia, 2004.
- [3] UNIGIS, “Infraestructura de datos espaciales”, 2016.
- [4] A. Carmona y J. Monsalve, “Sistemas de información geográficos», en Congreso de Ingeniería de Sistemas, 2004.
- [5] Organización de las Naciones Unidas, “Manual de sistemas y cartografía digital”, Naciones Unidas, Departamento de Asuntos Económicos y Sociales, 2000.
- [6] B. de Longueville, “Community-based geoportals: the next generation? Concepts and methods for the geospatial Web 2.0”, Elsevier, Computers, environment and urban System, vol. 34, pp. 299-308, 2010.
- [7] M. Bernabé y C. López, Fundamentos de las Infraestructuras de Datos Espaciales (IDE), Universidad Politécnica de Madrid: UPM, 2012.
- [8] C. Chasco, “El geomarketing y la distribución comercial”, Investigación y marketing, vol. 79.
- [9] J. Ojeda, P. Díaz, J. Álvarez, J. Pérez y A. Prieto, Geoportales y geovisores web: un nuevo entorno colaborativo para la producción, acceso y difusión de la información geográfica, Universidad de Zaragoza.
- [10] DANE, “Metodología censo general 2005”, Colombia, 2005.
- [11] A. Osorio y M. Idrobo, GisData, Geoprocess. Disponible en <http://www.geoprocess.com.co>
- [12] “Tecnologías de información geográfica para el manejo de los recursos naturales”, InfoResources, vol. 3, 2007.
- [13] L. González, R. C. Macía Arce, José C. Armas Quintá, F. J. Cabalar Fuentes, Coords., Procesos de Desarrollo en un Mundo Globalizado, Santiago de Compostela: Universidad de Santiago de Compostela, 2013. Disponible en: <https://www.agaliasociacion.org/AGALI/PUBLICACIONES/PDMG.pdf>
- [14] D. Ballari, L. Vilches, D. Randolph, D. Pacheco y V. Fernández, “Tendencias en infraestructuras de datos espaciales en el contexto latinoamericano”, Revista Científica Maskana, vol. 8.

## Capítulo XV

# REVISIÓN SISTEMÁTICA DE LITERATURA: FACTORES QUE AFECTAN EL PROCESO DE PRUEBAS DE SOFTWARE

*Darío Enrique Soto Durán - dsoto@tdea.edu.co*

Facultad de Ingeniería, Tecnológico de Antioquia - I. U., Medellín, Colombia.

*Katherine Villamizar Suaza - kvillamizars@gmail.com*

Facultad de Ingeniería, Tecnológico de Antioquia - I. U., Medellín, Colombia.

*Juan Camilo Giraldo Mejía - jgiraldo1@tdea.edu.co*

Facultad de Ingeniería, Tecnológico de Antioquia - I. U., Medellín, Colombia.

*Fabio Alberto Vargas Agudelo - fvargas@tdea.edu.co*

Facultad de Ingeniería, Tecnológico de Antioquia - I. U., Medellín, Colombia.

*Jovani Alberto Jiménez Builes - jajimen1@unal.edu.co*

Facultad de Minas, Universidad Nacional de Colombia (Unal), Medellín, Colombia

*Adriana Xiomara Reyes Gamboa - axreyes@elpoli.edu.co*

Facultad de Ingeniería, Politécnico Jaime Isaza Cadavid (PJIC), Medellín, Colombia

## I. INTRODUCCIÓN

El objetivo de las pruebas de software (PS) es garantizar que el producto cumpla con las funcionalidades requeridas y satisfaga con un nivel de calidad aceptable [1]. Las PS son el proceso que asegura la calidad del producto (grado en el que un conjunto de características cumple con los requisitos) [2], [3]. En consecuencia, las PS representan una revisión final de las especificaciones, el diseño y la codificación [4]. Por lo tanto, se consideran como un proceso paralelo al proceso de desarrollo que permite evaluar, desde diferentes aspectos y momentos, el comportamiento de un sistema o componente [5].

Para alcanzar la calidad de un producto existen variables que inciden en el proceso: el tiempo, los recursos, la infraestructura, la cualificación de los profesionales, la metodología empleada, los requisitos del software, etc. [6]. El propósito de esta revisión bibliográfica es identificar los factores que inciden en la efectividad y la eficiencia de las pruebas [7] y realizar una revisión crítica al respecto, que posibilite

la síntesis de la información disponible y sea la base para evidenciar posibles mejoras a los problemas o factores que se presenten en la revisión de literatura, teniendo en cuenta diferentes aspectos [8].

Diversos estudios han presentado factores y riesgos que afectan los procesos en el ciclo de vida del software. Un ejemplo es el del “Top 10 de factores que obstaculizan la mejora de los procesos de verificación y validación en organizaciones intensivas en software” [9]. Otro ejemplo es la exposición de los riesgos en proyectos de software [10]. Sin embargo, estos estudios se centran en el proceso del software mas no en los factores que lo afectan.

El presente artículo se encuentra organizado en cuatro secciones: la primera describe el protocolo que orienta la revisión de la literatura, la segunda corresponde al análisis y, la tercera, a los resultados, donde se relaciona el protocolo con el análisis crítico de las investigaciones seleccionadas. Por último, se presentan las conclusiones.

## II. METODOLOGÍA

La revisión de literatura se realizó bajo el protocolo de revisión [11], el cual permite, en el contexto de la ingeniería de software, apoyar la labor de los investigadores con el objetivo de evitar sesgos en la realización de este proceso. El protocolo plantea tres fases:

**Planificación:** Se refiere a las actividades previas a la revisión, para definir las preguntas de investigación, los criterios de inclusión y exclusión, las fuentes de información, la cadena de búsqueda y los procedimientos de categorización.

**Ejecución:** Se refiere a la búsqueda y la selección de los estudios, con el fin de extraer de ellos los datos y sintetizarlos.

**Resultados:** Es la fase final, y tiene por objeto la redacción de los resultados de acuerdo con las preguntas de investigación planteadas. Los resultados derivados de la revisión presentan un análisis crítico de los autores.

## III. ANÁLISIS

### A. Planificación

Preguntas de investigación y justificación: En la Tabla 1 se presentan las preguntas que se propone resolver con la revisión de la literatura, así como la justificación del porqué se plantean.

**Tabla 1.** Preguntas de Investigación y Justificación

N.º	Pregunta de investigación	Justificación de la pregunta
PI1	¿Cuándo y dónde fue publicado el estudio?	El tema de investigación es amplio y pertinente en la disciplina de ingeniería de software. Esta pregunta de investigación tiene como propósito conocer y comprender las fuentes de las publicaciones específicas para la temática y cuándo han sido publicadas.

N.º	Pregunta de investigación	Justificación de la pregunta
PI2	¿Cuál es el estado de los estudios que existen en el proceso de pruebas?	Existen diferentes niveles de madurez que alcanza un estudio en el ámbito científico. Esta pregunta de investigación se enfoca en identificar el estado en que se encuentra la investigación.
PI3	¿Cuáles son los factores que inciden en las pruebas de software?	De forma similar al interrogante anterior, se formula esta pregunta de investigación para establecer las categorías del proceso de pruebas donde se identifican factores que impactan el proceso.
PI4	¿Qué iniciativas se han elaborado para mitigar las problemáticas detectadas?	Identificar las principales soluciones que potencian el proceso de las pruebas de software. Esta información es útil para evaluar los beneficios reales de los estudios actuales en las pruebas de software, así como para señalar los problemas pendientes y, por tanto, la necesidad de profundizar en la temática.

**Criterios de inclusión y exclusión:** La selección de criterios está organizada en un criterio de inclusión (CI) y cuatro criterios de exclusión (CE). El criterio de inclusión es: (CI1) El estudio analiza los factores que dificultan el proceso de pruebas. Los criterios de exclusión son: (CE1): El estudio no puede ser un artículo corto. (CE2): El estudio tiene un tiempo de publicación mayor a 15 años. (CE3): La publicación es una versión anterior de otro estudio publicado. (CE4): La publicación es un artículo producto de talleres y tutoriales.



**Fuentes de búsqueda:** La revisión se enfocó en tres bases de datos electrónicas consideradas las más relevantes por [12]. Son ellas: IEEE Xplore (<http://ieeexplore.ieee.org>), SpringerLink (<http://www.springerlink.com>) y Science Direct (<http://www.sciencedirect.com>).

**Palabras clave y cadena de búsqueda:** La búsqueda de los estudios se realizó teniendo en cuenta el área de interés de la revisión y construyendo la cadena de búsqueda con las palabras claves. La cadena de búsqueda definida fue: (“Problem” OR “risk” OR “barrier”) AND (“software testing” OR “software test” OR “Testing” OR “Test”).

**Validación:** Antes de realizar la búsqueda de la información, el protocolo de revisión fue validado evaluando la pertinencia de las palabras claves y la cadena de búsqueda, con una pequeña muestra de publicaciones.

## B. Ejecución

Extracción de la información y síntesis: El proceso de búsqueda consideró publicaciones anteriores al mes de junio de 2016. La última búsqueda se realizó en agosto de 2016. Como resultado de la búsqueda se obtuvo un total de 202 publicaciones detalladas así: IEEE XPLORE (65), SpringerLink (39) y Science Direct (98). En primera instancia, se eliminaron las publicaciones duplicadas que aparecen en más de una fuente, y se obtuvo una reducción aproximada del 20 % (quedaron 162 publicaciones). En segundo lugar, se aplicaron los criterios de selección (criterios de inclusión y exclusión) sobre título, resumen y palabras clave, lo cual resultó en 49 documentos (reducción de aproximadamente el 69 %); en el tercer momento, se aplicaron los criterios de selección teniendo en cuenta el texto completo, y el resultado fue un conjunto de 39 publicaciones (reducción de aproximadamente del 20 %). Finalmente se obtuvieron 39 artículos discriminados así:

**Tabla 2.** Artículos Seleccionados

N.º de referencia	Título	
[13]	Optimal allocation and control problems for software-testing resources.	IEEE
[14]	Analysis of equipment’s software test’s problems and the countermeasures.	IEEE
[15]	Literature survey of Ant Colony Optimization in software testing.	IEEE
[16]	Analysis of Problems in Testing Practices.	IEEE
[17]	Optimal testing resource allocation, and sensitivity analysis in software development.	IEEE
[18]	On effectiveness of pairwise methodology for testing network-centric software.	IEEE
[19]	When to migrate software testing to the cloud?	IEEE
[20]	A model of third-party integration testing process management for foundation software platform.	IEEE
[21]	Optimal testing resource allocation, and sensitivity analysis for modular software testing.	IEEE
[22]	Reflective Architecture Based Software Testing Management Model.	IEEE
[23]	Comparative analysis of classical multi-objective evolutionary algorithms and seeding strategies for pairwise testing of Software Product Lines.	IEEE
[24]	Reliable code coverage technique in software testing.	IEEE
[25]	Research and implementation of knowledge management methods in software testing process.	IEEE
[26]	Optimal budget spending for software testing under the condition of nonlinear constraint.	IEEE

N.º de referencia	Título	
[27]	An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use.	SCIENCE DIRECT
[28]	Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the Cuckoo Search algorithm.	SCIENCE DIRECT
[29]	Design and analysis of different alternating variable searches for search-based software testing.	SCIENCE DIRECT
[30]	Investigating software testing and maintenance reports: Case study.	SCIENCE DIRECT
[31]	Quality assurance through rigorous software specification and testing: A case study.	SCIENCE DIRECT
[32]	Software safety testing based on STPA.	SCIENCE DIRECT
[33]	Software test process improvement approaches: A systematic literature review and an industrial case study.	SCIENCE DIRECT
[34]	Combinatorial testing for software: An adaptation of design of experiments.	SCIENCE DIRECT
[35]	Achieving quality on software design through test-driven development.	SCIENCE DIRECT
[36]	Constraint-based testing: An emerging trend in software testing.	SCIENCE DIRECT
[37]	Global software testing under deadline pressure: Vendor-side experiences.	SCIENCE DIRECT
[38]	Testing techniques selection based on ODC fault types and software metrics.	SCIENCE DIRECT
[39]	A survey of software testing practices in Canada.	SCIENCE DIRECT

N.º de referencia	Título	
[40]	Analyzing the relationships between inspections and testing to provide a software testing focus.	SCIENCE DIRECT
[41]	A Model-Driven approach for functional test case generation.	SCIENCE DIRECT
[42]	How are software defects found?	SCIENCE DIRECT
[43]	A characterization schema for software testing techniques.	SPRINGER
[44]	Cooperative software testing and analysis: Advances and challenges.	SPRINGER
[45]	Who tested my software? Testing as an organizationally cross-cutting activity.	SPRINGER
[46]	Risk orientation in software testing processes of small and medium enterprises: An exploratory and comparative study.	SPRINGER
[47]	Advances in test generation for testing software and systems.	SPRINGER
[48]	Mining software defect data to support software testing management.	SPRINGER
[49]	Observations on software testing and its optimization.	SPRINGER
[50]	Design and implementation of software testing cases management system based on J2EE.	SPRINGER
[51]	A model-based framework for classifying and diagnosing usability problems.	SPRINGER

Fuente: Elaboración propia

#### IV. RESULTADOS

¿Cuándo y dónde fue publicado el estudio? (PI1)  
Para dar respuesta a este interrogante, las publicaciones se caracterizaron teniendo en cuenta los siguientes criterios:

- Año de publicación (Fig. 1).
- País de la publicación (Fig. 2).
- Tipo de fuentes donde se encuentra la publicación.

Las 39 publicaciones se pueden visualizar en las gráficas. En la Figura 1 se puede observar que el tema en referencia ha tomado mayor relevancia en los últimos tres años, periodo en el que se presenta el 83% (25 de 39) de las publicaciones.



**Fig. 1** Cantidad de revisiones por año.

En la Figura 2 se puede concluir que frente a la temática los países con mayor número de publicaciones son EE. UU. y China, con aproximadamente el 51% (20 de 39) del total de estudios seleccionados.



**Fig. 2.** Cantidad de publicaciones por país

## V. DISCUSIÓN

La gestión de conocimiento. Estas categorías clasificadas por año y país agrupan las publicaciones que hacen referencia a la necesidad de incorporar el conocimiento en el proceso de pruebas de software. Diferentes autores plantean como premisa la necesidad de reusar el conocimiento técnico y el conocimiento del dominio de aplicación en los diferentes momentos del proceso con el propósito de optimizar las pruebas. Se retoman soluciones a experiencias previas por ser un soporte para el personal en la toma de decisiones aplicables a diferentes momentos de las pruebas de

software. No se pudo comprobar la validación de los estudios analizados.

**Gestión del proceso.** Esta categoría hace referencia a los diferentes procesos que cubren las actividades de verificación y validación. Se estandariza el marco de procesos en tres niveles: los procesos de organizaciones, los procesos de gestión y los procesos fundamentales.

Diferentes publicaciones identifican problemáticas en el contexto organizacional, específicamente en la estrategia que orienta el proceso, y proponen alternativas que plantean la eficiencia del proceso de pruebas desde la combinación de las técnicas estáticas y dinámicas.

En cuanto a los procesos fundamentales, las publicaciones revisadas analizan los procesos de pruebas desde el enfoque técnico y centran sus esfuerzos en optimizar técnicas de pruebas de caja negra y caja blanca.

Algunas publicaciones hacen un análisis de los criterios que implica la gestión de pruebas en un equipo de pruebas de forma deslocalizada. Este estudio plantea los desafíos y las estrategias que se deben incorporar en una metodología para equipos distribuidos.

**Automatización de pruebas.** La automatización es una de las alternativas para garantizar la efectividad frente al aumento de la complejidad del software, uno de los desafíos es lograr la automatización del proceso de pruebas al 100 %, y se evidencia que la mayor tendencia de esta clasificación es la generación automática de los casos de prueba.

En consecuencia, se han definido dos subcategorías: la primera denominada “Automatización de procesos”, que pretende garantizar el marco metodológico que orienta las organizaciones en las pruebas de software. En segundo lugar se propone la subcategoría “Automatización de pruebas dinámicas”. Estos trabajos están enfocados en la automatización de las pruebas funcionales, rendimiento, desempeño y carga.

En el análisis de la revisión se identifican iniciativas que plantean actividades de mitigación a las problemáticas que inciden en las PS y contribuyen a dos aspectos importantes: la reducción de costos y el incremento de la efectividad del proceso.

En la reducción de costos, algunos artículos proponen modelos para optimizar la asignación de recursos a las actividades de pruebas, parten del concepto de confiabilidad para apoyar a los gerentes de proyectos en la toma de decisiones respecto de la definición de recursos del proceso de pruebas.

Alineados con el propósito de incremento de la efectividad del proceso de pruebas, se identifican los siguientes trabajos: La efectividad se plantea a partir de la optimización de los procesos técnicos o fundamentales de las pruebas de software, algunos trabajos presentan investigaciones validadas desde el enfoque de la optimización de las pruebas combinatorias, usadas para detectar con eficacia fallos en el software en las pruebas de caja blanca y caja negra. Los estudios se centran en la generación de casos y datos de pruebas, y usan diferentes enfoques, como: optimización basada en el algoritmo de optimización por colonia de hormigas; lógica difusa, y algoritmos de búsqueda con el propósito de optimizar la eficiencia y el rendimiento de los conjuntos combinatorios de las técnicas de pruebas. En la misma línea se encuentran las investigaciones orientadas a la automatización de las PS focalizadas en los procesos técnicos, con el objeto de aportar a la generación automática de datos de entrada y así validar los casos de prueba, con mayor relevancia para las pruebas de caja negra.

## VI. CONCLUSIONES

De acuerdo al análisis de los estudios realizados, se identifican tres factores fundamentales que impactan en un alto nivel el proceso de pruebas como son: el conocimiento, el proceso y la automatización de las pruebas. Siendo el conocimiento un insumo esencial que incide en la efectividad del proceso y la eficiencia de la automatización de las pruebas.

En la dimensión de la gestión de conocimiento (GC), los principales factores identificados que inciden en el proceso de pruebas son:

- Barreras en la transferencia del conocimiento
- Pérdida de conocimiento
- Bajo índice de reúso del conocimiento
- Distribución y uso del conocimiento
- No se consideran de forma adecuada las experiencias y conocimientos adquiridos en la etapa de planificación.

El principal problema de la GC es el reúso del conocimiento, dado que las personas son reticentes para compartir el conocimiento, pues puede convertirse en un elemento competitivo en su entorno laboral, y llega a ser una causa común de varios autores para que la estrategia de GC no logre el objetivo del aprendizaje organizacional.

Dentro de las iniciativas en GC, la reutilización de casos de prueba es la estrategia de mayor prevalencia. La gran mayoría de investigaciones se enfocan en aspectos relacionados con la implementación de los procesos de GC en su conjunto o específicamente en una de sus fases. Por otra parte, los estudios se centran en presentar el diseño de sistemas basados en conocimiento como alternativa de implementación de la GC, y dan solución sólo a un aspecto del proceso de pruebas sin lograr la sinergia entre los diferentes elementos que conforman el sistema, como son las personas, la tecnología y la estrategia organizacional. Las técnicas de GC más utilizadas y con mejores resultados son las páginas amarillas y los mapas de conocimiento. Otras líneas de investigación sin explorar son: la gestión de conocimiento asociado a cada tipo de prueba y el proceso para obtener de manera fiable y estructurada las lecciones aprendidas para su posterior reúso.

En cuanto a la gestión del proceso, los factores de mayor relevancia que impactan el proceso de pruebas son:

- Falta de compromiso de la alta dirección.
- Modelo y metodologías existentes rígidos y complejos de implementar en las organizaciones de software.
- Ausencia o baja calidad de los artefactos de desarrollo.

El compromiso de la alta dirección es un elemento esencial en cualquier iniciativa empresarial de mejora de procesos, sin embargo, el tamaño de la organización y la estructura del equipo de pruebas inciden en la efectividad del proceso. Por ejemplo, en las organizaciones pequeñas, para minimizar los costos de los proyectos, asignan las responsabilidades de pruebas a los desarrolladores de software y soslayan en gran medida los criterios de pruebas. Una alternativa para este tipo de escenarios son los desarrollos basados en metodologías ágiles, porque permiten minimizar riesgos a medida que avanza el desarrollo.

La metodología o el modelo para el desarrollo del proceso de pruebas son un factor que asegura en gran medida el éxito del proyecto, alineados a la política y estrategia de pruebas establecidas en los proyectos de pruebas. En consecuencia, se han construido iniciativas que adoptan metodologías de desarrollo y mejora de procesos acorde a los contextos organizacionales. Con el estándar ISO 29119 recientemente liberado, se unifican conceptos y criterios de la aplicación de pruebas que resuelven el interrogante ¿qué se debe hacer?, pero sin resolver ¿el cómo se hace?, y es una oportunidad para un dominio común que permita establecer procesos para soportar investigaciones en el campo de la mejora de procesos y la gestión de conocimiento.

Otro elemento en esta dimensión es la inadecuada planeación del proceso, focalizada en la dificultad para identificar los riesgos que genera una selección inadecuada de la estrategia. Los activos del proceso de construcción son insumos primordiales para el proceso de pruebas de software, pero en algunos casos resultan incompletos y defectuosos y afectan la ejecución de las pruebas. Esta situación es reiterativa en la tercerización de pruebas y diferentes autores proponen alternativas como: trabajo cooperativo entre desarrolladores y probadores, estudios en análisis de riesgos, reúso de experiencias y optimización de pruebas exploratorias. Sin embargo, se deben explorar procesos de mejora basados en conocimiento y experiencias que permitan estimar y planear con información histórica.

En esta dimensión, las automatizaciones de las pruebas tienen varios factores que se repiten en la causa expresada por las investigaciones analizadas. Veamos:

- Alto nivel de complejidad y bajo nivel de usabilidad en las herramientas de automatización.
- Desconocimiento del uso de las herramientas por parte del personal.
- Falta de estandarización del diseño arquitectónico.
- Recursos ineficientes.

En la actualidad, el proceso de pruebas en los proyectos de desarrollo de software es una etapa sacrificada en tiempo, costos y personal frente a retrasos de desarrollo, que convierte la automatización en un proceso manual sin impacto en la calidad del producto.

Las investigaciones en esta área se enfocan en resolver la automatización de procesos técnicos para mitigar sesgos derivados de las operaciones manuales.

En esta área se pueden identificar nuevas líneas de investigación: la automatización de pruebas en el contexto SOA, líneas de productos y proyectos de explotación de datos.

## REFERENCIAS

- [1] C. Kaner, B. Johnson, J. Falk y H. Q. Nguyen, "This is near-final, B. L. Recruiting Software Testers", *Software Development*, N.º 7, pp. 62-64, 1999, San Francisco,.
- [2] ISO, I. IEEE, *Systems and Software Engineering-Vocabulary*. ISO/IEC/IEEE 24765: (E) Piscataway, NJ: IEEE Computer Society, Tech. Rep, 2010.
- [3] C. García, A. Dávila y M. Pessoa, "Test process models: Systematic literature review", en *International Conference on Software Process Improvement and Capability Determination*, Springer International Publishing, 2014, pp. 84-93.
- [4] F. A. Amo, L. M. Normand, y F. J. S. Pérez, *Introducción a la ingeniería del software*, Delta Publicaciones, 2005.
- [5] G. J. Myers, C. Sandler y T. Badgett, *The art of software testing*, John Wiley & Sons, 2011.
- [6] P. Kruchten, *The rational unified process: an introduction*, Addison-Wesley Professional, 2004
- [7] A. Bertolino, "Software testing research: Achievements, challenges, dreams", en 2007



- Future of Software Engineering, IEEE Computer Society, 2007, pp. 85-103.
- [8] M. J. Harrold, "Testing: a roadmap", en Proceedings of the Conference on the Future of Software Engineering, ACM, 2000, pp. 61-72.
- [9] J. García, A. de Amescua y M. Velasco, "Top 10 de factores que obstaculizan la mejora de los procesos de verificación y validación en organizaciones intensivas en software", REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software, vol. 2, N.º 2, pp. 18-28, 2006.
- [10] L. M. I. Infante, D. E. A. Rabí y M. M. Díaz, "Exposición a los riesgos en un proyecto de software: aplicación del modelo Mogerí", Serie Científica de la Universidad de las Ciencias Informáticas, vol. 6, N.º 6, 2013.
- [11] A. Kitchenham, Guidelines for performing systematic literature reviews in software engineering, Technical report, EBSE Technical Report. EBSE, Durham: University of Durham, 2007.
- [12] D. H. Ham, "A model-based framework for classifying and diagnosing usability problems", Cognition, technology & work, vol. 16, N.º 3, pp. 373-388, 2014.
- [13] H. Ohtera y S. Yamada, "Optimal allocation and control problems for software-testing resources", IEEE Transactions on Reliability, vol. 39, N.º 2, pp. 171-176, 1990.
- [14] J. P. Zhang, W. F. Li, J. Yang y Z. Y. Ding, "Notice of Retraction Analysis of equipment's software test's problems and the countermeasures", en International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering, (QR2MSE), IEEE, 2013, pp. 1143-1146.
- [15] B. Suri y S. Singhal, "Literature survey of ant colony optimization in software testing", en CSI Sixth International Conference on Software Engineering (CONSEG), IEEE, 2012, pp. 1-7.
- [16] J. Kasurinen, O. Taipale y K. Smolander, "Analysis of problems in testing practices", en Software Engineering Conference. APSEC'09, Asia-Pacific, IEEE, 2009, pp. 309-315.
- [17] C. Y. Huang y M. R. Lyu, "Optimal testing resource allocation, and sensitivity analysis in software development", IEEE Transactions on Reliability, vol. 54, N.º 4, pp. 592-603, 2005.
- [18] K. Z. Bell y M. A. Vouk, "On effectiveness of pairwise methodology for testing network-centric software", en Third International Conference on Information and Communications Technology ITI. Enabling Technologies for the New Knowledge Society, IEEE, December 2005, pp. 221-235.
- [19] T. Parveen y S. Tilley, "When to migrate software testing to the cloud?", En Third International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, 2010, pp. 424-427.
- [20] J. Gao, Y. Lan y M. Jin, "A Model of Third-Party Integration Testing Process Management for Foundation Software Platform", en Fourth International Conference on Wireless Communications, Networking and Mobile Computing. WiCOM'08, IEEE, 2008, pp. 1-4.
- [21] C. Y. Huang, J. H. Lo, J. W. Lin, C. C. Sue y C. T. Lin, "Optimal resource allocation and sensitivity analysis for modular software testing", en Fifth International Symposium on Multimedia Software Engineering. Proceedings, IEEE, 2003, pp. 231-238.
- [22] Y. A. O. Jun-feng, Y. I. N. G. Shi, L. U. O. Ju-bo, X. Dan y J. Xiang-yang, "Reflective architecture based software testing management model", en IEEE International Conference on Management of Innovation and Technology, vol. 2, IEEE, 2006, pp. 821-825.
- [23] R. E. Lopez-Herrejon, J. Ferrer, F. Chicano, A. Egyed y E. Alba, "Comparative analysis of classical multi-objective evolutionary algorithms and seeding strategies for pairwise testing of software product lines", en IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 387-396.
- [24] D. N. Rao, M. V. Srinath y P. H. Bala, "Reliable code coverage technique in software testing", en International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, 2013, pp. 157-163.
- [25] L. Xue-Mei, G. Guochang, L. Yong-Po y W. Ji, "Research and implementation of knowledge management methods in software testing process", en WRI World Congress on Computer Science and Information Engineering, vol. 7, IEEE, 2009, pp. 739-743.
- [26] H. Yongming, W. Xianglin y Y. Chaoyuan, "Optimal budget spending for software testing under the condition of nonlinear constraint", Journal of Systems Engineering and Electronics, vol. 14, N.º 3, pp. 92-97, 2003.
- [27] T. Mahmoud y B. S. Ahmed, "An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use", Expert Systems with Applications,

- vol. 42, N.º 22, pp. 8753-8765, 2015.
- [28] B. S. Ahmed, T. S. Abdulsamad y M. Y Potrus, "Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm", *Information and Software Technology*, vol. 66, pp. 13-29, 2015.
- [29] J. Kempka, P. McMinn y D. Sudholt, "Design and analysis of different alternating variable searches for search-based software testing", *Theoretical Computer Science*, vol. 605, pp. 1-20, 2015.
- [30] P. Janczarek y J. Sosnowski, "Investigating software testing and maintenance reports: Case study", *Information and Software Technology*, N.º 58, pp. 272-288, 2015.
- [31] L. Lin, J. He, Y. Zhang y F. Song, "Quality assurance through rigorous software specification and testing: A case study", *Procedia Computer Science*, N.º 62, pp. 257-265, 2015.
- [32] C. Yang, "Software Safety Testing based on STPA", *Procedia Engineering*, N.º 80, pp. 399-406, 2014.
- [33] W. Afzal, S. Alone, K. Glocksien y R. Torkar, "Software test process improvement approaches: A systematic literature review and an industrial case study", *Journal of Systems and Software*, N.º 111, pp. 1-33, 2016.
- [34] R. N. Kacker, D. R. Kuhn, Y. Lei y J. F. Lawrence, "Combinatorial testing for software: An adaptation of design of experiments", *Measurement*, vol. 46, N.º 9, pp. 3745-3752, 2013.
- [35] E. Guerra y M. Aniche, "Achieving quality on software design through test-driven development", en *Software Quality Assurance: Large Scale and Complex Software-intensive Systems*, Chapter 9, 2015.
- [36] A. Gotlieb, "Constraint-based testing: An emerging trend in software testing", *Advances in Computers*, N.º 99, pp. 67-101, Chapter Two, 2015.
- [37] H. Shah, M. J. Harrold y S. Sinha, "Global software testing under deadline pressure: Vendor-side experiences", *Information and Software Technology*, vol. 56, N.º 1, pp. 6-19, 2014.
- [38] D. Cotroneo, R. Pietrantuono y S. Russo, "Testing techniques selection based on ODC fault types and software metrics", *Journal of Systems and Software*, vol. 86, N.º 6, pp. 1613-1637, 2013.
- [39] V. Garousi y J. A. Zhi, "A survey of software testing practices in Canada", *Journal of Systems and Software*, vol. 86 N.º 5, pp. 1354-1376, 2013.
- [40] F. Elberzhager, J. Münch, y D. Assmann, "Analyzing the relationships between inspections and testing to provide a software testing focus", *Information and Software Technology*, vol. 56, N.º 7, pp. 793-806, 2014.
- [41] J. J. Gutiérrez, M. J., Escalona y M. Mejías, "A Model-Driven approach for functional test case generation", *Journal of Systems and Software*, N.º 109, pp. 214-228, 2015.
- [42] M. V., Mäntylä y J. Itkonen, "How are software defects found? The role of implicit defect detection, individual responsibility, documents, and knowledge", *Information and Software Technology*, vol. 56, N.º 12, pp. 1597-1612, 2014.
- [43] S. Vegas y V. Basili, "A characterization schema for software testing techniques", *Empirical Software Engineering*, vol. 10, N.º 4, pp. 437-466, 2005.
- [44] T. Xie, L. Zhang, X. Xiao, Y. F. Xiong y D. Hao, "Cooperative software testing and analysis: Advances and challenges", *Journal of Computer Science and Technology*, vol. 29, N.º 4, pp. 713-723, 2014.
- [45] M. V. Mäntylä, J. Itkonen y J. Iivonen, "Who tested my software? Testing as an organizationally cross-cutting activity", *Software Quality Journal*, vol. 20, N.º 1, pp. 145-172, 2012.
- [46] M. Felderer y R. Ramler, "Risk orientation in software testing processes of small and medium enterprises: An exploratory and comparative study", *Software Quality Journal*, vol. 24, N.º 3, pp. 519-548, 2016.
- [47] H. Yenigün, C. Yilmaz y A. Ulrich, "Advances in test generation for testing software and systems", *International Journal on Software Tools for Technology Transfer*, vol. 18, N.º 3, pp. 245-249, 2016.
- [48] R. Hewett, "Mining software defect data to support software testing management", *Applied Intelligence*, vol. 34, N.º 2, pp. 245-257, 2011.
- [49] D. Liu, S. Xu y H. Liu, "Observations on Software Testing and Its Optimization", en *Computer and Information Science*, R. Lee, ed., Switzerland: Springer International Publishing, 2013, pp. 33-49.
- [50] Lu, J., Su, N., & Zhao, A. "Design and implementation of software testing cases management system based on J2EE", en *Information Engineering and Applications*, Springer London, 2012, pp. 1268-1274.
- [51] D. H. Ham, "A model-based framework for classifying and diagnosing usability problems", *Cognition, Technology & Work*, vol. 16, N.º 3, pp. 373-388, 2014.

