



Tecnura

ISSN: 0123-921X

tecnura@udistrital.edu.co

Universidad Distrital Francisco José de Caldas  
Colombia

BOLAÑOS CASTRO, SANDRO JAVIER; LÓPEZ CHAPARRO, SERGIO ANDRÉS  
LMPS como propuesta alterna a BPMN para el modelado de procesos de software

Tecnura, vol. 16, núm. 34, octubre-diciembre, 2012, pp. 157-170

Universidad Distrital Francisco José de Caldas

Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257024712013>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en [redalyc.org](http://redalyc.org)

 redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# LMPS como propuesta alterna a BPMN para el modelado de procesos de software

*SPML as an alternative proposal to BPMN for modeling development processes*

## **SANDRO JAVIER BOLAÑOS CASTRO**

Ingeniero de sistemas, magister en Teleinformática. Docente de la Universidad Distrital Francisco José De Caldas. Bogotá, Colombia.

Contacto: [bsandrojavier@gmail.com](mailto:bsandrojavier@gmail.com).

## **SERGIO ANDRÉS LÓPEZ CHAPARRO**

Ingeniero de sistemas. Investigador de la Universidad Distrital Francisco José De Caldas. Bogotá, Colombia. Contacto: [checholopez6@gmail.com](mailto:checholopez6@gmail.com)

**Fecha de recepción:** 31 de agosto de 2011

**Clasificación del artículo:** Revisión de Tema

**Fecha de aceptación:** 26 de junio de 2012

**Financiamiento:** Pontificia Universidad de Salamanca

**Palabras clave:** lenguaje, marco de trabajo, modelo de proceso, notación.

**Key words:** language, framework, process model, notation.

## **RESUMEN**

En este artículo se presentará una descripción del nuevo lenguaje LMPS (Lenguaje de Modelado de Procesos de Software), para el modelado de procesos de software y se realizará una comparación con el estándar BPMN (Business Process Management Notation), estableciendo diferencias y similitudes en su notación, perspectiva, herramientas y propuesta, destacando las ventajas y

las desventajas del uso de cada notación; generando inquietudes al lector acerca de una alternativa para el modelado de procesos de software de una manera fácil y personalizada.

## **ABSTRACT**

This paper presents an overview of the recently-released LMPS (Lenguaje de Modelado de Proce-

sos de Software) language, intended for modeling software processes. LMPS is compared with the BPMN (Business Process Management Notation) standard, establishing differences and similarities in their notation, perspective, tools and propo-

sal. The idea is to highlight the advantages and disadvantages of using each of the notations and encourage the reader's reflections about an alternative approach that allows modeling software processes in an easy and customized way.

\* \* \*

## 1. INTRODUCCIÓN

La ingeniería de software es una disciplina de la ingeniería cuya meta es el desarrollo costeable de sistemas de software [1], además de generar y mantener sistemas de software dentro de las restricciones de tiempo, funcionalidad y costos acordados con el cliente [2], pero para lograr que se cumplan estas condiciones, es necesario tener claro que, el proyecto de software que se esté desarrollando, debe estar bajo control de quien lo dirige. Los proyectos varían de acuerdo a la dimensión y a la formalidad del proyecto y conforme aumenta la complejidad y el tamaño del mismo, la coordinación de sus elementos se dificulta debido al incremento en las relaciones entre los ingenieros de software, administradores y clientes [3].

En la actualidad, casi todos los países dependen de complejos sistemas informáticos, infraestructuras nacionales y utilidades, dependen de dichos sistemas, la fabricación industrial y la distribución está completamente informatizada, como es el ejemplo del sistema financiero [1]. Esto ha obligado a los expertos en software al estudio de nuevas propuestas de herramientas, lenguajes, estándares y notaciones que permitan tener una visión clara del proceso de software en términos de tiempo, presupuesto, avance, funcionalidad y recursos en un punto determinado del proyecto, obteniendo como resultado algo llamado: modelado de procesos.

El modelado de procesos es una técnica para la organización y la documentación de los procesos de un sistema, sus entradas, sus salidas y sus for-

mas de almacenamiento de datos. Una rama particular del modelado de procesos es el modelado de procesos de software, vale la pena aclarar que este no se centra únicamente en la descripción del proceso de software sino que va mucho más lejos [4], esta técnica permite captar las características principales del proceso, identificando su flujo y sus componentes para poder así comunicarlo con otros procesos, comprender las relaciones entre sus partes y conocer sus actores, responsables y demás artefactos que intervienen en su ejecución.

Estos procesos pueden ser representados de diversas formas y cada una de ellas suple necesidades diferentes debido a características inherentes a las herramientas de modelado usadas y a las notaciones definidas para el entendimiento por parte de todos los participantes del proyecto.

Es así como aparecen herramientas que se usan para el modelado, como los diagramas de flujo de datos (Gane & sarson) una de las más utilizadas en el mundo [4], propuestas como BPMN (Business Process Management Notation), EPC (Event-driven Process Chain) [5], que son estandarizadas por organizaciones internacionales como OMG (Object Management Group) y lenguajes como UML [6] (Unified Modeling Language) y BPEL (Business Process Execution Language) [7].

## 2. BPMN

De las propuestas de la OMG, cabe resaltar a BPMN, es un estándar del cual son miembros la mayoría de los proveedores más importantes de

**Tabla 1.** Objetivos BPMN vs Objetivos LMPS

OBJETIVOS BPMN [9]	OBJETIVOS LMPS [10]
Proporcionar una notación fácilmente comprensible por todos los usuarios del negocio, desde los analistas, los desarrolladores técnicos, hasta aquellos que monitorizaran y gestionaran los procesos.	Unificar un vocabulario común para los arquitectos, gestores e innovadores de procesos de software que facilita la comunicación de los participantes dentro del proceso de software.
Crear puentes entre el diseño de los procesos de negocio y la implementación del proceso.	Modelar los procesos y metodologías de desarrollo de software
Proporcionar una notación gráfica para los lenguajes basados en xml que describen procesos (BPEL4WS).	Registrar los fenómenos más importantes que se presentan en la aplicación de los procesos de software conducentes al control del proceso que se desarrolla y construir una base de mejora y madurez para nuevas pasadas del proceso o procesos futuros.
	Integrar los lenguajes de programación, diseño, arquitectura y demás que participan en el proceso de desarrollo.
	Estructurar los procesos de software basado en patrones del proceso de software ya definidos y que pueden ser expresados y documentados en el vocabulario LMPS.
	Promover el desarrollo de nuevas formas de proceso de software, al definir el proceso desarrollo de software como el objeto mismo de estudio.

Fuente: elaboración propia

tecnologías de información. BPMN se difundió rápidamente a nivel mundial y casi todos los proveedores, sean grandes o pequeños, académicos o consultores, empezaron a adoptar el standard [8], que tiene como principal objetivo proveer una notación que es legible y entendible por todos los usuarios del negocio, desde los analistas que crean los borradores iniciales de los procesos, los desarrolladores técnicos, responsables por la implementación de la tecnología qué llevará a cabo los procesos, y finalmente a las personas del negocio quienes van a administrar y monitorear los procesos [9].

### 3. LMPS

Es un lenguaje de modelado de procesos de software que nace gracias a la necesidad de proponer lenguajes que conviertan ideas del lenguaje natural LN (Dominio del problema), a expresiones de lenguaje de máquina LM (Dominio de solución),

como estrategia para solucionar un problema usando un sistema computacional.

Es un lenguaje cuyo propósito fundamental es el establecimiento de un vocabulario apropiado que permita plasmar las principales ideas que se dan en torno a los procesos de software, los participantes junto con sus roles, la comunicación, la producción, los artefactos, la información y el conocimiento, entre otros conceptos [10].

### 4. LMPS vs BPMN

Aunque el lenguaje LMPS, como la notación BPMN, son opciones para el modelado de procesos dentro de un proyecto, BPMN se enfoca en el modelado de procesos de negocio, mientras LMPS se enfoca en el modelado de procesos de software. Cada propuesta tiene características propias que hace que contengan, al mismo tiempo, similitudes y diferencias, por lo que vale

la pena realizar una comparación en puntos relevantes para aclarar al lector el ¿por qué tener en cuenta esta nueva propuesta LMPS? o decidir continuar con un estándar como BPMN.

## 4.1 Objetivos

Para empezar con este paralelo se estudiarán LMPS y BPMN por medio del análisis de los objetivos que plantean, para definir el propósito de su uso y los contextos en los que se convierten en una buena elección. Ver tabla 1.

### 4.1.1 Análisis

Aunque algunos de sus objetivos son similares, como el grupo de actores a quienes va dirigido (arquitectos, gestores, usuarios, analistas), se empiezan a marcar diferencias importantes en cuanto a los logros que quieren alcanzar. Por ejemplo, BPMN pretende proporcionar una notación para el modelado de procesos de negocio; mientras que LMPS lo hace para procesos de software, aunque no sólo se enfoca en los procesos, también aclara que apoya estos procesos por medio del modelado de metodologías. Adicionalmente, LMPS propone algo que vale la pena resaltar y que no se ve en BPMN, y es la capacidad de registrar fenómenos en el proceso con el objetivo de mejora y madurez del proceso.

Finalmente, se observa otro punto a favor de LMPS debido al objetivo de integración de los lenguajes de programación, diseño y arquitectura para manejar el proceso de software de una manera holística.

## 4.2 Teoría de grafos

Una de las finalidades comunes de las dos propuestas para modelado, es el uso de la teoría de grafos para traducir la gramática de los procesos a grafos, permitiendo definir, por medio de nodos y

aristas, diferentes semánticas de acuerdo a la forma de los elementos.

BPMN define un Business Process Diagram (BPD), que se basa en una técnica de grafos de flujo para crear modelos gráficos de operaciones de procesos de negocio. Un modelo de procesos de negocio es una red de objetos gráficos, que conforman actividades y controles de flujo que definen su orden de rendimiento [11].

LMPS está construido con una *gramática con estructura de frases* (que permite expresión formal del lenguaje y una representación gráfica de esa gramática) [10], denotada por  $G = (V, T, S, P)$  que consiste en un vocabulario  $V$ , un subconjunto  $T$  de  $V$  formado por los elementos terminales y un símbolo inicial  $S$  de  $V - T$  y un conjunto  $P$  de producciones. El conjunto  $V - T$  se denota por  $N$ . Los elementos de  $N$  se llaman elementos no terminales. Toda producción debe contener al menos un elemento no terminal en su lado izquierdo.

La gramática con estructura de frases es aplicada en LMPS de la siguiente manera: el vocabulario ( $V$ ) está expresado en un *nivel meta* y un *nivel objeto*, el nivel meta, es una base fija e inmodificable y configura el LMPS, se mantiene este vocabulario como fijo para generar una guía de categorización del vocabulario en términos de gramática. El nivel objeto configura el vocabulario formado por las palabras objeto, y como instancias pueden ser extendidas con mecanismos como el estereotipo.

$T$  = conocimiento blando, conocimiento duro, documento, anotación, texto operador estructural, operador funcional, actor, rol, entrada, salida, proceso, metodología, fase, efectividad, tarea, instrucción, indicación, contraindicación.

$N$  = elemento, funcional, estructural, cognitivo, informacional, operacional, participantes, artefactos, mecanismos, pautas.

$S$  = elementos. [12].

#### 4.2.1 Análisis

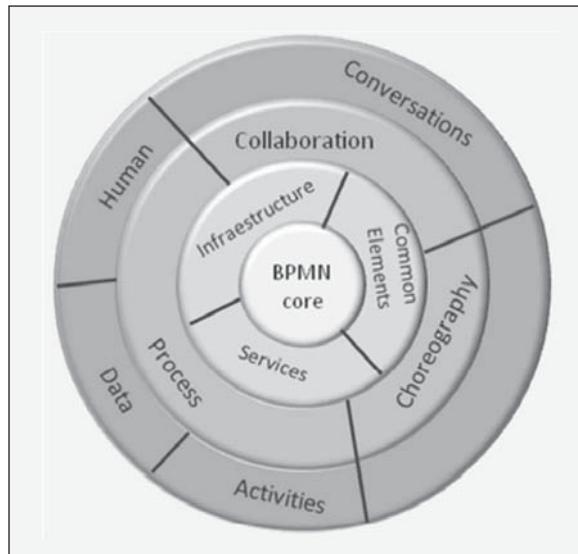
Ambos lenguajes están basados en teoría de grafos, con BPMN se define un BPD de grafos de flujo, que a su vez tienen la ventaja de transformarse en otros lenguajes como BPEL por medio de su representación XML, algo con lo que aún no cuenta LMPS, pero este último permite al usuario diseñar su proceso no solo desde un modelo (nivel objeto) sino también desde un meta modelo (nivel meta) permitiéndole al usuario definir los modelos que se van a crear, especificando los componentes de sus modelos de proceso de software y no creando simplemente los modelos de proceso.

#### 4.3 Estructura

En BPMN el “Process Modeling Conformance” se ve representado en el diagrama del núcleo de BPMN (figura 1). En donde se observan los tipos de modelos que se pueden generar conservando el principio de extensibilidad y los principios básicos de trabajo por capas que pueden ser compuestos en un camino bien definido. El enfoque usa constructos de formalización para la extensibilidad que son aplicados consistentemente a la definición.

El efecto adicional del trabajo por capas es que la compatibilidad de las capas pueda ser construida, permitiendo diferentes niveles de cumplimiento entre los proveedores, y también permitiendo a los vendedores adicionar sus propias capas en soporte a diferentes industrias verticales.

La especificación BPMN es estructurada en capas, donde cada capa se construye en la parte superior de unas capas más bajas, incluido el núcleo, que tiene los elementos más fundamentales de BPMN, los cuales son requeridos para la construcción de diagramas BPMN: procesos,



**Figura 1.** Representación del núcleo BPMN y su estructura por capas.

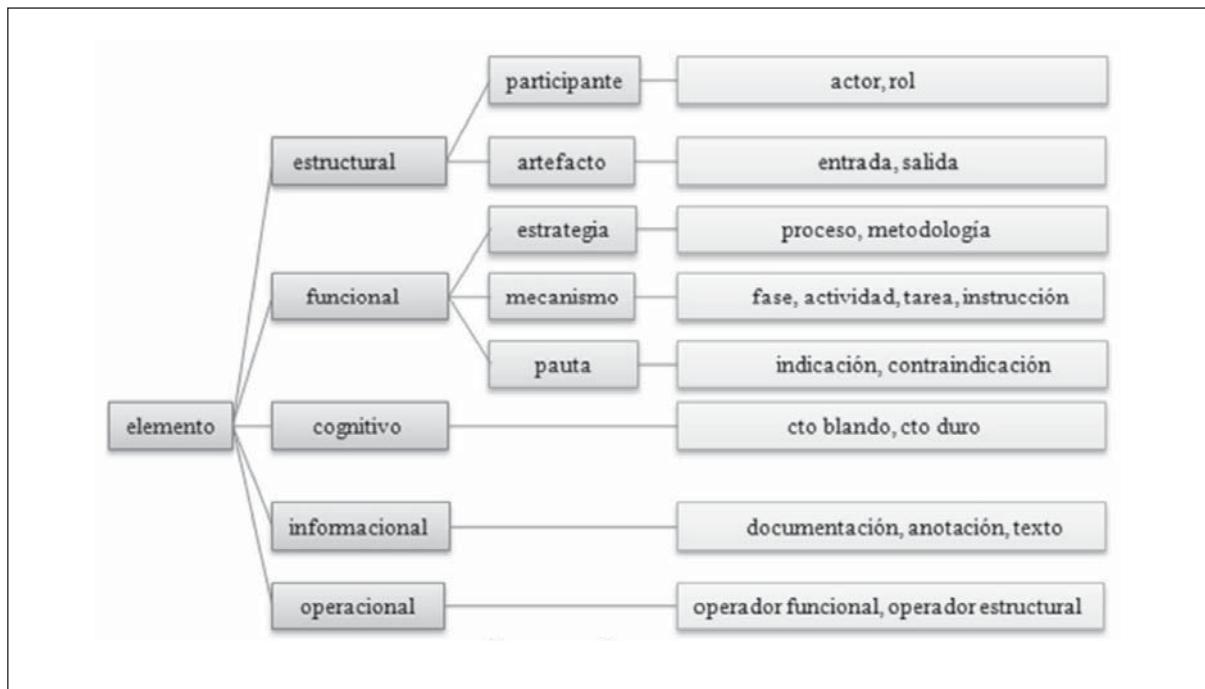
Fuente: tomado de [13]

coreografía y colaboración. El núcleo intenta ser simple, conciso, y extensible con un comportamiento bien definido [13].

En LMPS los elementos están constituidos de la misma manera pero con la forma de árbol (ver figura 2) en donde el elemento es la raíz y a partir de éste se van desplegando hijos que van siendo cada vez más específicos, dependiendo de la necesidad, se van generando elementos estructurales, funcionales, cognitivos, operacionales o informacionales, y estos a su vez tienen sus propias subdivisiones hasta llegar a las hojas del árbol [10].

#### 4.3.1 Análisis

Tanto LMPS como BPMN, manejan su estructura desde un núcleo y van expandiéndose de acuerdo a las necesidades de modelado, con un camino de extensibilidad (BPMN) y categorización (LMPS) definidos, que permiten dar claridad al usuario



**Figura 2.** Producciones LMPS.

Fuente: elaboración propia

acerca de la notación que desea usar para representar sus ideas de proceso, mientras que BPMN puede trabajar sus fases de manera personalizada y separada con un camino definido, en LMPS se conserva el punto de vista holístico de los sistemas, es decir, a pesar de que se tiene claro a qué categoría pertenece cada elemento, para que el modelo de proceso tenga una representación clara y entendible, es necesario que sus partes interactúen y formen un todo.

#### 4.4 Notaciones

Cada lenguaje presenta un conjunto de elementos gráficos que tienen representación semántica, a continuación se mostrarán algunos de ellos para realizar una comparación de algunos comunes y otros con características propias de cada propuesta. Ver tabla 2.

#### 4.4.1 Análisis

En las notaciones, para cada lenguaje se pueden ver algunas ventajas y desventajas que se explicarán a continuación, ya que, mientras en BPMN se cuenta con diversidad de elementos definidos (por ejemplo las diferentes notaciones que existen para representar una tarea) y son reconocidos mundialmente por la estandarización realizada por OMG y probablemente una persona que conozca BPMN al ver alguno de sus grafos reconozca inmediatamente de qué elemento se trata; en LMPS se cuenta con la posibilidad de personalización de los elementos, ya que, por ejemplo, hay un sólo ícono que representa una tarea pero éste puede ser reconocido como una tarea personalizada en el proceso de desarrollo de software (debug, compile, develop, document, deploy and test) por medio de un identificador gráfico, lo que permite al usuario representar su lenguaje natural en uno de los

**Tabla 2.** Cuadro comparativo notaciones BPMN vs LMPS

BPMN		LMPS	
	Actividad		Actividad
	Evento		Proceso
	Gateway		Operadores
	Conector		Conectores
X	X		Metodología
X	X		Actor Rol
	Contenedores		Cada elemento es un contenedor
	Entrada y Salida		Entrada y Salida
			Fase
	Tarea y Tipos de Tarea		Tarea
X	X		Conocimiento Blando
X	X		Conocimiento Duro
	Datos		Documentos
X	X	<ul style="list-style-type: none"> <li>⊕ contraindicacion</li> <li>↓ limitacion</li> <li>✗ restriccion</li> <li>✗ riesgo</li> <li>⊕ indicacion</li> <li>● funcion</li> <li>* objetivo</li> <li>✓ practica</li> </ul>	Indicaciones y Contraindicaciones
	Coreografía		Interfaces

Fuente: elaboración propia

elementos gráficos, sin limitaciones y permitiendo abstraer sus modelos con unos gráficos guía.

## 4.5 Patrones y antipatrones

Debido a la cantidad de elementos con los que cuenta BPMN, es posible que los usuarios nuevos que decidan incursionar en esta notación, cometan errores en cuanto a la conexión entre los elementos, el uso de gráficos adecuados y planteamiento de procesos de una manera estándar. Es por esto que los expertos en BPMN han trabajado generando una lista de patrones y anti patrones al momento de diseñar los modelos de procesos, entre los que se destacan *Uso De Tareas Y Eventos* (analistas a menudo modelan erróneamente eventos y tareas. Por ejemplo, eventos son modelados equivocadamente como tareas y los estados de las tareas como nuevas tareas), *Mal Uso De Flujos*

*Entre Pools* (cuando se modelan eventos de inicio y fin, pools y flujos de secuencia son a menudo perdidos, porque erróneamente se cree que los flujos de mensaje sustituyen los flujos de secuencia; adicionalmente, los flujos de secuencia son mal usados al conectar pools), *Flujos Dentro De Lanes* (lanes son, a menudo, erróneamente usadas en forma similar a un pool. Estos últimos de manera equivocada contienen más procesos de negocio o contienen flujos de mensajes entre diferentes lanes), entre otros [14].

A pesar de que LMPS es una propuesta reciente, viene dotada de un conjunto de buenas prácticas en la aplicación de procesos de software, compiladas en un conjunto de patrones, y un conjunto de errores comunes presentes en la aplicación de procesos de software compilados en un conjunto de antipatrones. Ver tabla 3.

**Tabla 3.** Cuadro comparativo patrones y antipatrones BPMN vs LMPS.

BPMN	LMPS
<p><b>Patrones:</b></p> <ul style="list-style-type: none"> <li>• No redundancia de tareas.</li> <li>• Eventos no producen flujo de mensajes</li> <li>• Un proceso por lane</li> <li>• Diferenciación proceso y subproceso</li> <li>• Uso de compuertas</li> <li>• Patrones de workflow</li> </ul>	<p><b>Patrones:</b></p> <ul style="list-style-type: none"> <li>• Patrón flujo de trabajo líneas</li> <li>• Patrón flujo de trabajo paralelo</li> <li>• Patrón flujo de trabajo cíclico</li> <li>• Patrón rotación de rol</li> <li>• Patrón de comunicación</li> <li>• Patrón de ejecutores</li> <li>• Patrón insumo – producto</li> <li>• Patrón gestión documental</li> <li>• Patrón de trazabilidad</li> <li>• Patrón generación de conocimiento</li> <li>• Patrón estados del conocimiento</li> <li>• Patrón Recipiente de Conocimiento</li> </ul>
<p><b>Antipatrones:</b></p> <ul style="list-style-type: none"> <li>• Mal uso de flujos entre pools.</li> <li>• Confusión entre eventos y tareas.</li> <li>• Eventos de tiempo usados incorrectamente.</li> <li>• Uso de eventos de mensaje y flujos de mensaje.</li> <li>• Flujos dentro de lanes.</li> <li>• Uso de mecanismo de secuencia de flujo.</li> </ul>	<p><b>Antipatrones</b></p> <ul style="list-style-type: none"> <li>• Proceso Top TOP</li> <li>• Súper proceso</li> <li>• Proceso extremo</li> <li>• Proceso casual</li> <li>• Proceso tobogán</li> <li>• Proceso inmutable</li> <li>• Proceso sin evidencia</li> <li>• Proceso sin ritmo</li> <li>• Proceso dominó</li> <li>• Proceso perpetuo</li> <li>• Proceso sin cabeza</li> <li>• Proceso sin comunicación</li> </ul>

Fuente: elaboración propia

#### 4.5.1 Análisis

BPMN y LMPS muestran un arduo trabajo para conseguir dos objetivos fundamentales que son: la detección de antipatrones en el modelado de procesos para la puesta en común de errores recurrentes en el diseño de procesos, y la generación de patrones que den una guía del camino correcto al momento de modelar un proceso de software, para que el usuario pueda, por medio de esta guía, definir un camino personalizando pero siguiendo estándares que harán que su actividad de modelado se realice de manera óptima.

#### 4.6 Tipos de diagramas

BPMN presenta elementos para diseñar diagramas de modelo de procesos, pero estos procesos no necesariamente tienen que ser procesos de software, sin embargo, existen variaciones de un solo tipo de diagrama, dichas variaciones se deben a la adición u omisión de elementos en el diagrama; pero, finalmente, se obtiene como resultado un sólo tipo de diagrama desde la misma perspectiva.

LMPS presenta una propuesta novedosa e interesante, debido a que, a pesar de que el resultado también es un diagrama de procesos de software, introduce algo llamado puntos de vista o *viewpoints*, que configuran los grafos bien formados que el lenguaje recomienda como ángulos clave a tener en cuenta dentro del modelado de procesos. Los puntos de vista son agrupados en tres categorías: la categoría de gestión, estructuración e innovación.

La categoría de gestión se enfoca en los puntos de vista a partir de la administración del proceso de software, destacando puntos de vista como participantes, responsabilidades, documentación y producción. La categoría estructuración hace especial énfasis en la arquitectura del proceso de

software, para ello propone los puntos de vista: estrategia, mecanismo, artefactos y contribución. Y como tercer grupo de puntos de vista, está la capa de innovación, enfocada a la madurez y mejora del proceso, en los que se plantean los puntos de vista mejora, conocimiento, comunicación e incidentes, estos son descritos en la tabla 4 [10].

#### 4.6.1 Análisis

BPMN presenta un solo tipo de diagrama, en el que se pretende especificar los diversos tipos de modelos de procesos sin importar el punto de vista de quién los realice; de manera alterna, LMPS propone algo novedoso que incluye el modelado de procesos de software a partir de diversos puntos de vista, en los que se persigue un objetivo distinto y que probablemente pueden ser realizados por actores diferentes. Este puede ser uno de los puntos más relevantes de diferenciación entre la nueva propuesta (LMPS) y la tradicional (BPMN), debido a que, cuando el gestor del proyecto quiera observar los modelos de proceso de software, no tendrá que observar modelos quizás más técnicos (categoría estructuración), sino que irá directamente a la categoría de gestión, caso similar ocurrirá con los arquitectos, que encontrarán una categoría específica en la cual observar el modelo de proceso de software y para los encargados de madurez y mejora de procesos que se referirán a la categoría de innovación.

#### 4.7 Retroalimentación y madurez del proceso

La dirección de proyectos de software se enfoca en gestionar exitosamente los procesos de trabajo asociados con el desarrollo, mantenimiento y soporte de productos y sistemas intensivos. Por gestión exitosa se entiende que los productos y servicios generados por los procesos cumplen completamente con los requisitos del cliente interno y externo, y que ellos satisfacen los objeti-

**Tabla 4.** View Points LMPS.

VIEWPOINT	DESCRIPCIÓN
Participantes	Permite modelar jerarquía de actores y roles, ubicar las fuentes de información claras que participan en el proceso de desarrollo además y mostrar la configuración del recurso humano con el que se contará.
Responsabilidades	Permite asignar a los roles que participan en el proceso de desarrollo (fases – actividades – tareas – interacciones) así como indicar funciones, objetivos y prácticas.
Documentación	Está dirigido al registro documental del proceso de desarrollo y la manera en la cual los participantes lo desarrollan.
Producción	Permite asociar los productos que los participantes deben generar. Esta configuración no solo le da visibilidad al proceso de desarrollo, sino que permite materializar el trabajo de los participantes.
Estrategia	Plantea modelar desde la perspectiva del proceso a la de la metodología. Con los mecanismos o pautas asociadas.
Mecanismos	Se centra a detallar el workflow del proceso o metodología, los mecanismos empiezan desde una categoría superior en la fase, pasan por las actividades y tareas hasta llegar a la categoría atómica en las instrucciones.
Artefactos	Enfatiza en la trazabilidad que deben tener los artefactos y que ayudan a estructurar y dar visibilidad y control al proceso o metodología de desarrollo.
Contribución	Integra el proceso, los mecanismos y los artefactos posibilitando hacer notorio el ciclo de vida de los artefactos, procesos, y/o metodología.
Mejora	Apunta al establecimiento de las pautas que favorece con una metodología y en vía de ella el proceso de desarrollo de software.
Conocimiento	Este punto de vista propone describir el conocimiento que se puede evidenciar en el proceso. Este conocimiento representa experticia, habilidades, destreza, actitud en general condiciones inherentes a los participantes.
Comunicación	Apunta a estrechar la relación de los participantes vía la interacción entre estos, manejadas explícitamente a través de una interfaz oportuna que pueda pasar desde una reunión formal, hasta una conversación casual.
Incidentes	Propone controlar limitaciones, restricciones, riesgos y contraindicaciones que debilitan la estrategia definida.

Fuente: elaboración propia

vos de negocio de la organización responsable de desarrollar los productos [15].

Por esto, es importante, para cualquier líder o gestor de proyecto de software, saber qué procesos son los que están funcionando de la manera esperada y cuáles aún tienen dificultades debido a situaciones internas o externas. Es necesario entonces, llevar un control de los procesos, para esto se recomienda que se detecten los riesgos y se gestionen, posibilitando una mejor toma de decisiones; se destaqueen los conocimientos acumu-

lados de las personas encargadas de los procesos y se generen buenas prácticas a partir de estos.

BPMN no ofrece como notación esta posibilidad, ya que para esto hace uso de BAM (Business Activity Monitoring) un software externo que se usa para monitorear las actividades del negocio y así lograr la mejora continua de los procesos. Por el contrario, LMPS está concebido de manera tal que los procesos siempre vayan en pro de la mejora continua y de la madurez, esto lo logra por medio de sus puntos de vista de innovación (me-

jora, conocimientos, comunicación e incidentes) ya que es posible detectar los aspectos positivos y los negativos de un proceso, así como definir sus riesgos y gestionar el conocimiento duro (se puede almacenar) o blando (adquirido a través de la experiencia) [16], que está presente en los procesos a través de los actores.

#### 4.7.1 Análisis

Se podría decir que BPMN no cuenta con elementos propios para la mejora de los procesos, ni para determinar la madurez de los mismos, debido a que depende de un software externo que se encarga de monitorizarlos. Por el contrario, LMPS introduce un punto de vista exclusivo para la mejora de procesos, otro para la detección de incidentes y otro para la gestión de conocimiento; bien sea el que está almacenado o aquel adquirido con la experiencia de los participantes en el modelado del proceso.

#### 4.8 Frameworks o herramientas para modelado

El último factor a evaluar es la cantidad y facilidad de herramientas para el modelado de estas propuestas. BPMN cuenta con un amplio número de frameworks, unos libres, otros licenciados, para realizar modelos de procesos de negocio con notación BPMN, por lo que los usuarios podrían adaptarse fácilmente a su proveedor preferido o a las comodidades de diseño que ofrecen estos frameworks. Por el contrario, LMPS cuenta con un solo framework para realizar este modelado, este es llamado “Coloso Enterprise” [17] y es un proyecto desarrollado en la Universidad Distrital Francisco José De Caldas que está en su versión Teseo [10], dando la posibilidad de, además de diagramas LMPS, realizar diagramas UML [6] y ARCHIMATE [18], y permitir conectarse con fuentes en JAVA entre otros [19]. A continuación se presenta un *screenshot* con los diagramas propuestos en el lenguaje LMPS. Ver figura 3.

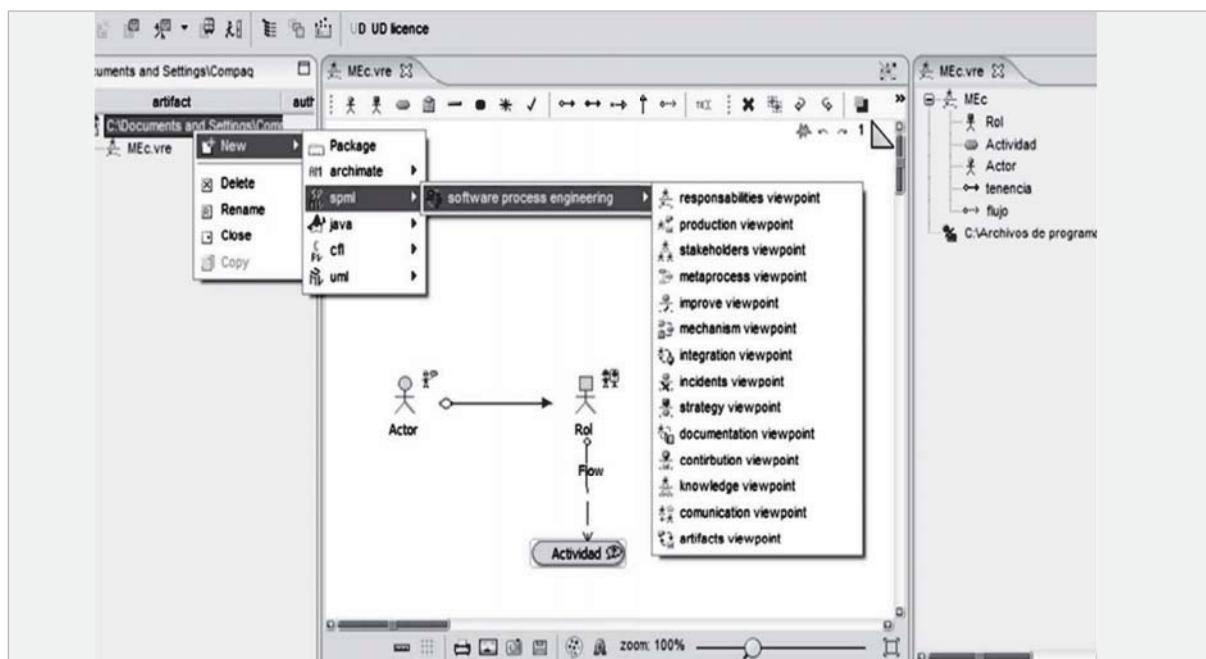


Figura 3. Screenshot coloso.

Fuente: elaboración propia

## 4.8.1 Análisis

LMPS cuenta con un sólo framework para el modelado, pero da posibilidades adicionales como diagramar en otros lenguajes o conectar con fuentes de lenguajes de programación como java; el software tiene una interfaz amigable para el usuario y está en sus primeras versiones. Por el contrario, BPMN cuenta con una extensa cantidad de frameworks, por lo que es posible para el usuario decidir y escoger el que le parezca más adecuado para sus necesidades y gustos.

## 5. TABLA COMPARATIVA

Finalmente, se ha realizado una tabla comparativa (ver tabla 5), con los factores analizados a través de este artículo para brindar, a manera de

resumen, los pros y contras de cada uno de los lenguajes.

## 6. CONCLUSIONES

BPMN y LMPS se presentan como alternativas para modelar procesos de software en proyectos donde es indispensable tener control de los mismos. Esto lo realizan por medio del uso de la teoría de grafos, transformando las ideas plasmadas en lenguaje natural a modelos gráficos que representen de manera estándar los flujos y todos los artefactos relacionados con el proceso de software. A través del presente artículo, se han descrito LMPS y BPMN por medio de puntos de comparación diferentes, mostrando sus principales características, fortalezas y debilidades para una mejor visión de la propuesta que ofrecen.

**Tabla 5.** Comparación LMPS vs BPMN.

FACTOR	LMPS	BPMN
Dirigido a los arquitectos, diseñadores, desarrolladores, stakeholders y usuarios	SI	SI
Notación para modelado de procesos	SI	SI
Notación para el modelado de metodologías	SI	NO
Interoperabilidad con otros lenguajes existentes como BPEL gracias a el lenguaje XML	NO	SI
Reconocimiento apoyado por una organización internacional	NO	SI
Los lenguajes de programación, diseño, arquitectura para manejar el proceso de software de una manera holística.	SI	NO
Documentación suficiente para autoestudio y práctica	NO	SI
Posibilidad de diseñar modelos y metamodelos	SI	NO
Categorización o extensibilidad bien definidos para dar claridad al usuario acerca del tipo de elementos que debe usar para el modelado	SI	SI
Gráficos mundialmente reconocidos, debido a estandarización.	NO	SI
Personalización en los gráficos en colores y semánticas.	SI	NO
Generación de patrones en la construcción de modelos de procesos	SI	SI
Varios tipos de diagrama dependiendo del punto de vista del usuario que ve el proceso	SI	NO
Elementos para la mejora y la madurez de los procesos	SI	NO
Acumulación y detección de conocimiento duro y blando en el proceso	SI	NO
Cuenta con framework para diseñar el lenguaje o notación	SI	SI
Da la posibilidad de escoger que framework usar para su uso	NO	SI

Fuente: elaboración propia

BPMN es una notación dirigida a arquitectos, diseñadores, desarrolladores, usuarios y actores para modelado de procesos de negocio, pero no de metodologías, cuenta con interoperabilidad con otros lenguajes como BPEL por medio del lenguaje XML, tiene reconocimiento internacional gracias al OMG, es por esto que sus gráficos son fácilmente identificados, pero no da la posibilidad de relacionarse con lenguajes de programación, diseño y arquitectura vitales para el proceso de software. Cuenta con documentación y frameworks suficientes para el autoestudio y diseño de modelos, aunque no da la opción de diseñar metamodelos. Estos modelos pueden tener errores en su forma, por lo que cuenta con un conjunto de patrones y antipatrones detectados gracias a la experiencia de los usuarios. No presenta diferentes tipos de diagrama en cuanto al punto de vista del usuario, no tiene elementos propios para la mejora y madurez de los procesos ni para plasmar el conocimiento adquirido empíricamente por los actores del proceso, pero es una notación usada mundialmente gracias al consenso de expertos en el modelado de procesos.

LMPS es un lenguaje dirigido a arquitectos, diseñadores, desarrolladores, usuarios y actores para modelado de procesos y metodologías de desarrollo de software, no cuenta con interoperabilidad con otros lenguajes como BPEL pero da la posibilidad de interactuar con lenguajes de programación, de diseño y de arquitectura que son impor-

tales para un desarrollo holístico del proceso de software. Es un proyecto que está en sus etapas iniciales, por lo que no cuenta con reconocimiento por una entidad internacional y sus gráficos aún no son estándar, aunque esta debilidad se trata de superar dando la posibilidad al usuario de personalizar sus iconos con semántica y descriptores adicionales. LMPS no cuenta aún con la suficiente documentación para el autoestudio, aunque ha trabajado fuertemente en patrones de diseño generados para ayudar a sus usuarios futuros en el modelado de procesos. Cuenta con un framework para su uso llamado “Coloso Enterprise” que ofrece posibilidades de interacción con otro tipo de lenguajes (UML, ARCHIMATE, JAVA). Presenta un factor importante en cuanto a los posibles diagramas que se pueden diseñar de acuerdo al punto de vista del proceso, ofreciendo doce viewpoints, especializando las actividades y así mismo su representación, además de una fortaleza importante debido a la característica de promover la detección de riesgos e incidentes y el manejo de conocimiento para la mejora y madurez de los procesos.

Queda a elección del lector cuál de estos dos lenguajes o notaciones decide usar para el modelado de procesos de software de sus proyectos, (cabe la posibilidad que elija un tercero), pero una de las finalidades del artículo es crear inquietudes acerca de este nuevo lenguaje LMPS, experimentar con sus propuestas y compararlo con una notación tradicional robusta como BPMN.

---

## REFERENCIAS

---

- [1] I. Sommerville, *Ingeniería del Software*, 7th ed., Madrid: Addison-Wesley, 2005, pp. 4- 59.
- [2] M. García and J. Rodríguez, “Aplicación del modelado de procesos en un curso de ingeniería de software”, *Revista Electrónica de Investigación Educativa*, Vol. 3, No. 2, p. 2, 2001.
- [3] R. Kraut and L. Streeter, “Coordination in software development”, *Communications of the ACM*, pp. 63-80, 1995.
- [4] V. Fernández, *Desarrollo de Sistemas de Información*, Barcelona: UPC, 2006, pp. 175-176.

- [5] M. Dumas, W. Van der Aalst, and A. Ter-Hofstede, *Process-Aware Information*, New Jersey: John Wiley & Sons, Inc, 2005.
- [6] *UML Unified Modeling Language: Infrastructure version 2.0*, OMG Standard, 2005
- [7] *Web Services Business Process Execution Language Version 2.0*, OASIS Standard, 2007.
- [8] J. Freund, B. Rucker y B. Hitpass, *BPMN 2.0 Manual de Referencia y Guía Práctica*, Santiago de Chile: BenhardHitpass, 2011, p. 4.
- [9] J. Perez, “Notaciones y lenguajes de procesos. Una visión global”, *Departamento de sistemas y lenguajes informáticos, Universidad De Sevilla*, [en línea]. Disponible: <http://www.lsi.us.es/docs/doctorado/memorias/Perez,%20Juan%20D.pdf>
- [10] S. Bolaños, “COLOSO SPML”, *Colombian Patent 120118080*, 2009.
- [11] S. A. White, “Introduction to BPMN”, *IBM Corporation*, [en línea]. Disponible: [http://www.bpmn.org/Documents/OMG\\_BPMN\\_Tutorial.pdf](http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf)
- [12] K. Rosen, *Matemática Discreta y sus Aplicaciones*, 5th ed., Mc Graw Hill, 2004.
- [13] *BPMN Business Process Model and Notation*, OMG Standard, 2011.
- [14] G. Polančič y T. Rozman “Notación para el modelado de procesos de negocio (BPMN) Poster”, *ITPoster*, [en línea]. Disponible: [http://upload.wikimedia.org/wikipedia/commons/a/ac/BPMN\\_Poster.pdf](http://upload.wikimedia.org/wikipedia/commons/a/ac/BPMN_Poster.pdf)
- [15] W. Florac, R. E. Park and A.D. Carleton, *Practical Software Measurement: Measuring for Process Management and Improvement*. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1997, pp.1-12.
- [16] I. Nonaka, “A dynamic Theory of Organizational Knowledge Creation”, in *Organization science*, 1994, pp. 14 – 37.
- [17] *Coloso Enterpriseversiónteseo 2011*, COLOSOFT E.U. [Online]. Available: <http://www.colosoft.com.co/>
- [18] *ARCHIMATE version 1.0*, Opengroup Standard, 2008-2009
- [19] *JAVA Standard Edition version 6.0*, Sun Standard, 2006.