

# Marco conceptual para el metaproceso de desarrollo de software

**Sandro Javier Bolaños Castro**

Universidad Distrital “Francisco José de Caldas”, Bogotá D.C., Colombia  
sbolanos@udistrital.edu.co

**Víctor Hugo Medina García**

Universidad Distrital “Francisco José de Caldas”, Bogotá D.C., Colombia  
vmedina@udistrital.edu.co

**Jesús Soto Carrión**

Universidad Pontificia de Salamanca, Madrid, España  
jesus.soto.carrion@acm.org

## RESUMEN

En este artículo se presenta el marco conceptual propuesto para el metaproceso de software, donde se detalla la gestión, estructuración e innovación como los procesos que configuran a su vez el metaproceso. Se destaca, no solo el flujo de trabajo en el que clásicamente se enfocan los procesos de desarrollo de software, sino que también se hace énfasis en variables clave que afectan los procesos y que permiten definir el mismo proceso de desarrollo de software como objeto de estudio. Se ven los procesos desde una categoría superior, crea roles y conocimiento emergente entorno a los procesos, constituyendo una herramienta más fuerte para afrontar el desarrollo de software.

**Palabras clave:** Metaproceso de desarrollo, proceso de desarrollo, gestión, estructuración e innovación.

## ABSTRACT

This article presents the conceptual framework proposed for the software metaprocess, where the management, structuring and innovation are detailed as processes that in turn shape the metaprocess. It highlights not only the work flow where software development processes are traditionally focused on, but also emphasizes the key variables that affect the processes and define the software development process itself as a case study. The observation of processes from a higher level creates roles and emerging knowledge around them, forming a stronger tool to address software development.

**Key words:** development metaprocess, development process, management, structuring, innovation.

## 1. INTRODUCCIÓN

El metaproceso enfrenta ángulos más allá de los abordados por los procesos o metodologías. Clásicamente modelos de procesos como: cascada, espiral, fases, evolutivo, prototipos (Sommerville, 2011), (Pressman 2009) o procesos como RUP (Jacobson, et al. 2001), enfatizan en el flujo de trabajo; mientras las metodologías centran su interés en prácticas y como estas se articulan finalmente, también al igual que los procesos en un flujo de trabajo. Aparentemente el bloque fundamental para los procesos y metodologías es la actividad y las formas de interacción entre ellas. De lado se dejan entre otros aspectos fundamentales, sus ejecutores, sus productos, el conocimiento generado, la información asociada y las pautas de los diferentes mecanismos empleados y que se pueden ver como categorías o subcategorías de las mismas actividades. El metaproceso de software no entra en conflicto con el proceso o metodología de software, más bien amplía su espectro fortaleciendo aún más una buena aplicación de la ingeniería de software.

## 2. ASPECTOS QUE DESTACA EL METAPROCESO DE SOFTWARE

“Los problemas significativos que enfrentamos no pueden resolverse al mismo nivel de pensamiento en el que estábamos cuando los creamos” Albert Einstein (Gómez, 2002).

El metaproceso de software propone caracterizar un proceso de software o metodología desde la *gestión*, *estructuración* e *innovación*, estos procesos corresponden respectivamente a la administración, arquitectura y creación de rasgos para el proceso de software (ver Figura 1).

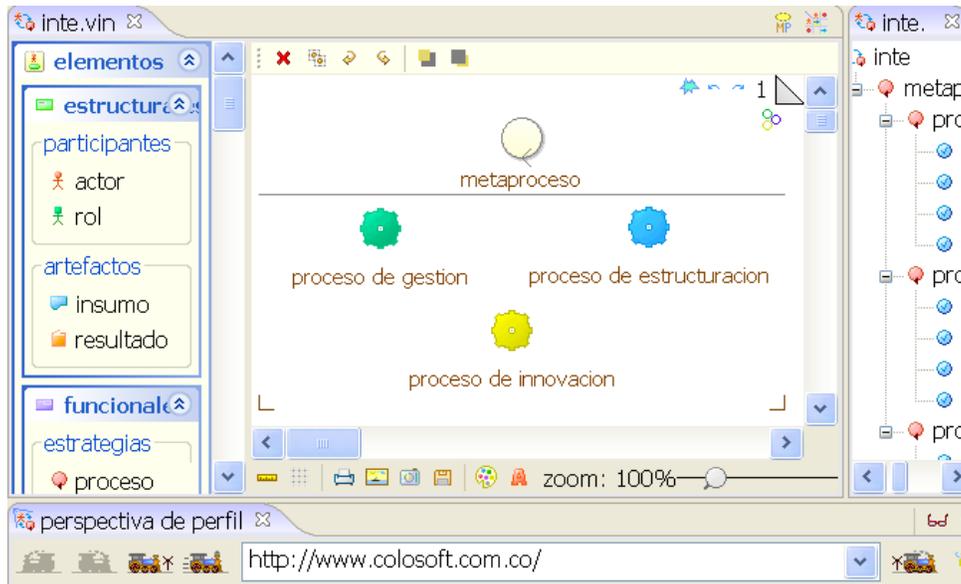


Figura 1: Procesos del Metaproceso. Fuente: Autores, extraído del software Coloso<sup>1</sup>

### 2.1 GESTIÓN

La gestión de software centra su interés en los participantes del proceso, estos asumen posturas de actores y roles, es decir el metaproceso promueve definir claramente la organización y responsabilidades para el proceso (ver Figura 2).



Figura 2: Gestión. Fuente: autores, extraído del software Coloso.

Definir los participantes permite identificar con claridad las fuentes de información, “personas” involucradas en el proceso. Al igual que es importante identificar quienes participan, es importante definir quienes no deben ser

<sup>1</sup> www.colosoft.com.co Coloso es el ambiente desarrollado para soportar el metaproceso de software, implementado por uno de los autores, Sandro Bolaños.

considerados en el proceso y con ello eliminar el ruido que estos introducen. Cuando se identifica un actor es posible proponer el papel que jugará en el proceso y asociado a él sus responsabilidades. Con las responsabilidades claramente se desprende el hecho de tener unos subproductos, los que configuraran el producto final. La gestión dentro del metaproceto por sí misma genera el rol de gestor del metaproceto, quien se enfocará en definir los participantes, las funciones realizadas por estos y que deben producir.

Otra de las características de la gestión es *la trazabilidad*; para el metaproceto es importante definir como los artefactos que se originan en el proceso van evolucionando convirtiéndose con cada *fase, actividad, tarea o instrucción* en un artefacto más elaborado. Por ejemplo, un requerimiento se convierte en una funcionalidad, esta a su vez, en una pieza de software, etc. La trazabilidad promueve la visibilidad del trabajo que se va realizando en el proceso, así como en el rastro que se debe seguir en el proceso no sola hacía delante; si no también hacía tras, es decir, de principio a fin como ha progresado algo que inicia siendo un concepto y se convierte en un producto, como también definir desde el producto que fue lo que lo originó; esta última característica “trazabilidad” permite verificar una pieza de software, promover actividades de mantenimiento y desarrollo dentro del proceso.

## 2.2 ESTRUCTURACIÓN

La estructura permite dentro del proceso definir los bloques de construcción del proceso y como estos interactúan. Definir la arquitectura del proceso es importante para tener el mapa del proceso (ver Figura 3). Poder estructurar un proceso permite resolver dos preguntas fundamentales. *¿Cómo* actúa el proceso? y *¿Por qué* un proceso actúa de una forma determinada? Los procesos convencionales resuelven el *cómo*, es decir, se dirigen los esfuerzos a definir como seguir un conjunto de actividades pero no promueve la indagación del *¿Por qué* las actividades son dispuestas de una u otra forma? o *¿Por qué* es clave definir una actividad, junto con sus insumos y resultados?.



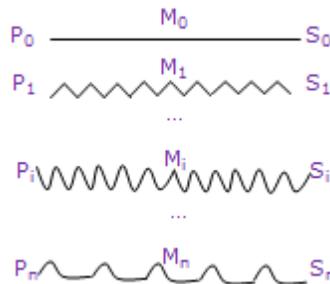
**Figura 3. Estructuración. Fuente: autores, extraída del software Coloso.**

La estructuración permite la personalización del proceso, no debería aceptarse como ley del proceso, el que sea *definido*. Los procesos no deben ser, ni convertirse en guiones estáticos; más bien deben ajustarse a las condiciones del problema, de los participantes, de los recursos, en fin de todas las variables que lo afectan. Puede ser que en la selección de un proceso se opte por realizar un modelo conocido, por ejemplo un modelo en espiral, un modelo por prototipos, uno por fases, en fin un modelo de los ya establecidos; aunque estos son ya definidos, el proceso que condujo a ellos no lo es, por lo tanto, se está haciendo metaproceto. Usar un mismo proceso una y otra vez puede ser nocivo, no buscar nuevas formas de resolver un problema a la postre creara limitaciones cuando el problema cambie y se vuelva más complejo o sea simplemente un nuevo problema. En la Figura 4 se esquematiza la solución a un problema P a través de un método M obteniéndose como solución S.



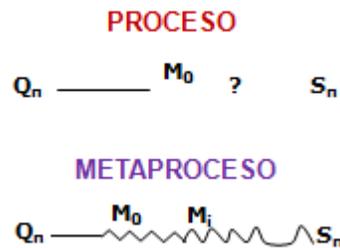
**Figura 4: Solución usando proceso. Fuente Autores.**

En la figura 5 se esquematiza la solución a un problema P a través de un método M que varía obteniéndose como Solución S.



**Figura 5: Solución usando metaprocreso. Fuente autores.**

Qué sucede cuando el problema cambia de naturaleza P a Q ? La respuesta es contundente; ya no es aplicables un método M definido por tanto solo el método que se ha experimentado en variedad de formas es la fuente de solución. Es aquí a donde apunta el metaprocreso, a establecer diferentes formas del proceso de tal manera que se acomode a nuevas consideraciones de problemas (ver Figura 6).



**Figura 6: Solución de Proceso Vs Metaprocreso. Fuente: Autores.**

Reflexionar a cerca de la estructura del proceso es fundamental, es posible que no se esté en disposición de hacer del proceso parte del objeto de estudio; pues ya se tiene suficiente trabaja con el software mismo, pero es la única manera de obtener la herramienta adecuada para construir la solución. La estructuración de un proceso significa construir una estrategia, definir unos mecanismos, unos artefactos y propiciar una contribución.

El metaprocreso propone dos estrategias posibles el proceso y/o la metodología. Escoger una opción u otra hace parte del problema de ingeniería de software y no debe asumirse como una postura de facto, algún proceso de moda, la metodología top o algún antecedente previo. Si bien es cierto se debe apostar a lo conocido también es

cierto que es necesario acondicionar ya sea el proceso o la metodología a cada problema, existe una extensa gama de procesos y metodologías con perspectivas que pueden ser favorables de un problema a otro, ese estudio es el propuesto por el metaproceso. La estrategia para el metaproceso se encarga entonces del estudio, como ya lo hacen los procesos y metodologías, del flujo de trabajo, necesario para desarrollar el *problema-solución* pero desde una óptica superior, desde el bosque de los procesos y metodologías y no desde los árboles. Insistiendo en la necesidad de considerar el proceso como objeto de estudio dentro del proyecto de software, dicho estudio puede verse como el proceso de selección o creación del proceso. De este enfoque surge por supuesto un nuevo rol este es el de arquitecto de proceso, el cual se encarga entonces de definir la estructura del proceso o metodología del software siendo el proceso y la metodología el bloque de construcción de estrategia para el metaproceso.

Una vez definida la estrategia es necesario definir los bloques para construirla. Estos son los que van a definir el flujo de trabajo, que a su vez está formado por las actividades, estas estarán formadas por tareas y finalmente estas a su vez estarán formadas por instrucciones. Cada uno de estos conceptos, permiten configurar la estrategia. Convencionalmente hablamos de actividades como los bloques fundamentales sin embargo el metaproceso propone una categoría superior en el flujo de trabajo y dos subcategorías las tareas e instrucciones. Con esta ampliación es posible modelar mucho mejor un proceso o metodología. Los mecanismos *flujo de trabajo - actividad - tarea - instrucción*, facilitan al arquitecto definir con más precisión un proceso o metodología, además jugar a las construcciones de propuestas que realmente se ajusten a las condiciones del problema de software que se debe resolver.

Es importante dentro de la estructuración definir un *mapa de artefactos* que se constituyen en la guía de los elementos a desarrollar, tener hitos visibles desde un inicio facilita el seguimiento del proceso, los artefactos, además de visibilidad, aportan los elementos necesarios para considerar los recursos que se deben invertir en el proceso, por lo menos desde la perspectiva de los artefactos. Los artefactos también posibilitan establecer, métodos para el proceso desde la perspectiva de los resultados obtenidos, una vez se han invertido unos insumos, los artefactos le permiten al metaproceso articularse con puntos de vista, diagramas, esquemas, entre otras perspectivas que son típicamente, realizadas usando lenguajes de programación, diseño o arquitectura. Por ejemplo un diagrama de casos de uso es para el metaproceso un artefacto, un diagrama de organización de archimate (The Open Group. 2009), un cronograma de actividades, en fin cualquier grafo, documento o recurso visible puede ser un artefacto valioso de resaltar en el metaproceso.

Finalmente al metaproceso propone la necesidad de establecer cuál es la contribución que se tiene después de la ejecución de una estrategia y uso de unos mecanismos. Esta contribución es referida fundamentalmente al servicio o producto que arroja el metaproceso. Definir de antemano el fin al que debe conducir una estrategia favorece el desarrollo controlado dentro de un proceso. Clásicamente los procesos suponen una salida al final de la realización de flujo de trabajo, pero no la explicitan: Para el metaproceso es fundamental proponer unos resultados visibles, ya sean estos servicios y/o productos. Para el metaproceso el software puede considerarse a la vez un producto o un servicio. Además como resultado de un proceso de desarrollo se obtiene productos alternos como manuales, documentación, instaladores fuentes entre otros.

### 2.3 INNOVACIÓN

El otro enfoque del metaproceso esta dado en la innovación, es necesario que los procesos establezcan formas en las que constantemente se reinventen, incluso se establezcan nuevas formas de estrategias, mecanismos y pautas. Para fortalecer los procesos de innovación es necesario proponer formas de fortalecer la *comunicación, intercambio de conocimiento, verificación de la mejora* y establecimiento de los *incidentes* típicos dentro de los procesos (ver Figura 7).



**Figura 7. Innovación. Fuente autor. Extraída del software coloso.**

La comunicación ha sido una herramienta clave desde los inicios del hombre, fortalecer formas de comunicación le ha posibilitado el desarrollo de la cultura, el arte y la ciencia. Hacer énfasis en la comunicación como medio de innovación promueve en los procesos de software hacer claras las estrategias, mecanismos y pautas, además hacerlas visibles, entendibles y por ello controlables. Muchos procesos y metodología dan algunas alternativas de comunicación como por ejemplo las entrevistas, los cuestionarios, la programación por pares, las reuniones diarias de 15 minutos. Infortunadamente estos métodos no se han acentuado como definitorios para los procesos, la comunicación puede fortalecer los procesos o entorpecerlos si se carece de esta. Ya la torre de Babel (Brooks, 1995), enseñó como un proyecto con todos los recursos y condiciones favorables puede fracasar si se carece de comunicación. El metaproceto plantea la necesidad de desarrollar los métodos de comunicación, como elemento fundamental a la hora de desarrollar software, no propone que un método sea mejor que otro, propone la necesidad e importancia de estudiar y ajustar las formas de comunicación más convenientes. En ese sentido se debe como mínimo definir el rol de *emisor*, el rol de *receptor*, el *canal* y por supuesto el *mensaje* a transmitir (Shannon, 1948). En un modelo de reunión por ejemplo el emisor los constituye el líder de la reunión, los receptores lo forman los participantes de la reunión, la reunión es el medio de transmisión en tanto que el mensaje lo conforma la información que el líder comunica en forma oral. De un modelo de reunión se extrae unas conclusiones y otras consideraciones importantes para el metaproceto. La comunicación facilita el progreso del proceso alertando tempranamente sobre problemas y ajustes necesarios.

El otro énfasis de la innovación esta en el conocimiento. No hay progreso alguno en el área que sea, si no hay una acumulación de conocimiento. El conocimiento es el catalizador fundamental en el metaproceto, permitiendo mejorarlo por las experiencias que se van obteniendo. Los procesos convencionales asumen de facto que ya está todo conocido y que lo único necesario es aplicar una formula ya lograda sin reconocer la necesidad de mejorar; el metaproceto es perfectible así mismo. El metaproceto propone gestionar el conocimiento, adoptando planeamientos de gestión de conocimiento. Para el metaproceto hay dos tipos de conocimiento: el conocimiento blando y el conocimiento duro. El conocimiento blando reside en los individuos por tanto solo es transferible a través de los procesos propuestos por Nonaka (Nonaka and Ikujiro, 1995). El conocimiento duro es el que reside fuera del individuo en sitios como la organización e incluso los procesos mismos. El metaproceto plantea la necesidad de identificar las categorías del conocimiento y describirlas de tal manera que pueda hacerse un seguimiento y así lograr identificar su progreso.

La innovación también propone dentro del metaproceto identificar los incidentes que se cometieron, se cometen y se podrán cometer, esta forma de ver el metaproceto permite atender los riesgos y fortalecer el metaproceto. Lo que se plantea con identificar los incidentes, es permitir que posibles errores, debilidades, defectos, en fin cualquier contra desfavorable pueda ser tratada sin dejar espacio a fallas, los incidentes son una muy buena forma de ver el otro lado de la moneda, fortalece la práctica del proceso evitando los posibles problemas. El metaproceto anima a listar los incidentes, junto con sus contras que eviten tales incidentes.

Finalmente, el metaproceto plantea enumerar las mejoras que se han logrado al aplicarlo, estas deben residir en un banco de mejoras que se conviertan en insumos de nuevas pasadas, ya sea que estas sean de dominio público o privado. Las mejoras se pueden identificar como todos los rasgos que han contribuido a que el metaproceto sea mejor, es decir, sea entendible, aplicable y seguible. Mientras los incidentes son la cara negativa de aplicar el

metaproceto, las mejoras son la cara positiva, ambas caras deben ser administradas para sacar el mejor provecho del metaproceto.

### 3. FLUJO DE TRABAJO DEL METAPROCETO

El metaproceto se centra en la gesti3n, estructuraci3n e innovaci3n y cada una de estas perspectivas tiene sus puntos de vista, definir qu3 se debe hacer o por d3nde empezar, no es el gui3n que pretende plantear el metaproceto. Un ingeniero de metaproceto puede decidir hacer arquitectura, luego gesti3n y despu3s innovaci3n; puede definir hacer solo arquitectura, o solo gesti3n, o solo innovaci3n, cualquier combinaci3n posible; se tendr3n resultados m3s completos si se logran recorrer todas las perspectivas, pero a3n as3 no necesariamente se debe tener un orden est3tico. El metaproceto anima a tener el propio marco de trabajo, con el ensamble de sus recomendaciones, lo contrario desemboca en crear un proceso m3s de los ya existentes.

### 4. DIFERENCIAS ENTRE EL METAPROCETO Y EL PROCESO

Por definici3n el metaproceto es una categor3a superior del proceso por lo tanto centra su estudio en el proceso y no en la aplicaci3n del proceso en particular, y aunque uno de los objetivos es llegar a la aplicaci3n de un proceso no es el 3nico, de hecho el principal es llegar a la decisi3n de definir el proceso, entre las diferencias m3s importantes que el metaproceto tiene con respecto al proceso est3n:

- En tanto el proceso es una soluci3n limitada, el metaproceto es una categor3a y por lo tanto completa. Dado un problema P y una soluci3n S al problema P siendo la funci3n soluci3n M. En donde M es el proceso para todo P en adelante se tendr3 siempre como soluci3n a S.

$$M(P) = S \equiv M : P \rightarrow S$$

A diferencia del proceso el metaproceto propone como funci3n soluci3n una composici3n.

$$M M' M''(P) = S$$

Seg3n la composici3n planteada se obtienen una soluci3n S diferente gracias a que el metaproceto propicia la composici3n. Las ventajas de esta son claras es posible crear un espectro de soluci3n y escoger lo m3s conveniente.

- Los problemas de software son ajustadas a procesos en tanto que el metaproceto se ajusta al problema. Encajar todos los problemas a la visi3n parcializada de un proceso en particular es como tener como 3nica herramienta de construcci3n un martillo y tener que ver todos los artefactos para esa construcci3n como clavos.
- El proceso de software es est3tico, en tanto que el metaproceto es din3mico, dado que los procesos se han centrado en definir una estructura dotada de actividades establecidas, los hace poco flexibles en su aplicaci3n. Aparentemente esto es una ventaja, quien no desea un gui3n al que se ajuste todo, esto es cierto si los fen3menos tratados fueran inertes, est3ticos y constantes pero al contrario los problemas de software son tan vol3tiles y din3micos que la herramienta usada para resolverlo tambi3n deben ser consideradas en la misma forma. El proceso es una suerte de mapa que nos ubica con precisi3n en un lugar. Por supuesto tan solo en miles, quiz3s millones de a3os un mapa podr3a verse modificado por el movimiento en las masas continentales, y ya no ser muy 3til. Infortunadamente en ingenier3a de software el tiempo se reduce a d3as, semanas, meses en el mejor de los casos, por tanto es demasiado optimista pensar que un mapa fijo propuesto por un proceso nos sirva en todas circunstancias y tiempo. El metaproceto plantea estudiar el

mapa y adaptarlo al problema, no es lo mismo tratar de entender la hidrografía de una región con un mapa económico o político. Los mapas en el metaproceso son objeto de estudio.

- El proceso de software se centra en si mismo en tanto que el metaproceso se centra en las variables que afectan el proceso. Los procesos de software no han ido más allá de las consideraciones con respecto al flujo de trabajo, el metaproceso analiza otras variables como son el individuo, el conocimiento que se genera, las estrategias que se utilizan entre otras variables que afectan el proceso. El metaproceso viabiliza el proceso al estudiar cuál es el pertinente para la solución de un problema determinado.
- El proceso utiliza el conocimiento existente en tanto que el metaproceso promueve la generación de conocimiento. Es demasiado optimista considerar a un proceso como la formula salvificadora cuando se desarrollo un proyecto de software, esto en consideración que el proceso parte de unos supuestos que espera sean suficientes para resolver el problema. El metaproceso no da por sentado el uso de un proceso, busca a través del conocimiento de las variables si existe algo que pueda ser usado de lo contrario promueve la generación de un proceso apropiado basado en un conocimiento emergente.
- El proceso puede ser visto como un conjunto de actividades planeadas para resolver un problema, el metaproceso puede ser visto como un conjunto de procesos y las variables que los afectan no solo limitándose a las actividades sino extendiéndolas a categorías y subcategorías, además se consideran los participantes sus roles, los artefactos e incluso operaciones entre los elementos que constituyen el metaproceso.
- El proceso propone un marco conceptual particular en tanto que el metaproceso propone un marco conceptual general y un vocabulario que permite expresar consistentemente, ese marco, el vocabulario establece un lenguaje que debe basarse en gramáticas (Chomsky, 2004), (Rosen, 2004).

$$G = (V, T, S, P)$$

## 5. APORTACIONES DEL METAPROCESO

El metaproceso es una propuesta consistente, robusta y completa acerca del proceso de software. Provee los elementos necesarios, para su realización, señalando los conceptos clave que hacen parte del problema.

El metaproceso no es proceso más de los existentes, es una propuesta dotada de consideraciones que promueven el estudio del proceso y las variables que lo afectan tendientes a gestionar, estructurar e innovar sobre él.

El metaproceso propone un vocabulario simplificado, dirigido de manera directa al tratamiento adecuado del proceso de desarrollo de software, este vocabulario es la base para la constitución de un lenguaje de modelamiento dirigido específicamente a los proceso de desarrollo.

El metaproceso origina roles específicos para los procesos de desarrollo. El rol de gestor de proceso se enfoca en la administración del proceso enfatizando en el manejo del recurso humano; el rol de arquitecto de proceso está dedicado a la estructuración y personalización del proceso; el rol de innovador del proceso está dedicado a la indagación de nuevas formas y variables para el proceso.

El metaproceso definiría la guerra de los procesos y metodologías de software a favor del estudio y mejoramiento de los procesos y metodologías existentes, proporcionado además una base conceptual para la personalización de los procesos, incluso la creación de ellos.

## 6. CONCLUSIONES

- Las fortalezas en ingeniería de software, se han originado con aportaciones que apuntan al desarrollo de una base teórica que soporta el trabajo disciplinado de esta área de conocimiento, y una de las áreas con mayor impacto es la referida a los procesos de desarrollo de software.
- Hay una gama muy extensa de procesos y metodologías de desarrollo, es evidente que no existe un proceso o metodología especial que sea reconocido(a) como el grial para el desarrollo de software, mas bien cada proceso o metodología reclama para sí la supremacía.
- El metaproceso no es un proceso más que intenta reclamar el puesto de honor de los procesos y metodologías, este propone dos aspectos fundamentales, el primero es ver desde una perspectiva superior los procesos y el segundo es tratar las variables no vistas por los procesos pero de suma importancia. Ver los procesos implica reflexionar sobre ellos incluyendo los aspectos relevantes que entorno a los procesos los impactan. Esto es lo que define el *meta* del proceso.
- El metaproceso propone estudiar el hecho mismo de escoger un proceso o metodología, lo que puede conducir a uno ya conocido, pero también puede conducir a una reflexión particular de u proceso personalizado, situación más adecuada para la solución de un problema.

## 7. REFERENCIAS

- Brooks, F. (1995). *The mythical Man - Month*. Addison Wesley.
- Chomsky, N. (2004). *Estructuras Sintácticas*, Siglo XXI editores.
- Gómez, G. (2002). *Optimicemos la educación con PNL, Programación Neurolingüística*. Trillas.
- Jacobson, I. Booch, G. Rumbaugh, J. (2001). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- Pressman, R. (2009). *Ingeniería del Software*. Mc Graw Hill. Edición 7th.
- The Open Group. (2009). *Archimate 1.0 Specification*. [www.archimate.org](http://www.archimate.org).
- Nonaka, Ikujiro. (1995). *The knowledge creating company: how Japanese companies create the dynamics of innovation*. New York: Oxford University Press.
- Rosen, K. (2004). *Matemática Discreta*, Mc Graw Hill.
- Shannon, C. (1948). *A mathematical theory of communication*. Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656.
- Sommerville, I. (2011). *Software Engineering*. Addison Wesley. Edition 8th.

### ***Autorización y renuncia***

*Los autores autorizan a LACCEI para publicar el escrito en los procedimientos de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito.*

### ***Authorization and disclaimer***

*Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.*