



resumen

En este artículo se presentan los resultados del proyecto de grado “Modelo Evolutivo para Procesos Software, fase 2 (MEPS 2)”. MEPS contempla cuatro niveles y pretende guiar a los desarrolladores de *software* en su proceso de formación. Tiene el soporte de un ambiente *software* que le permitirá al desarrollador controlar su evolución y un mapa conceptual que le dará claridad sobre los conceptos básicos de la ingeniería de *software*. El proyecto se centró en obtener un conocimiento claro del marco que encierra los procesos *software* y modelos de calidad de proceso para incorporarlo a MEPS.

# Modelo evolutivo para procesos software, fase 2

Andrés Mauricio Bernal P. • Alejandro Alfonso Perea L.

introducción

**Durante** su ejercicio profesional, el ingeniero de *software* tiene la responsabilidad de desarrollar productos que cumplan con los requerimientos determinados por el cliente y los criterios básicos de calidad que debe tener cualquier producto *software* [1]. El incumplimiento de estos aspectos se evidencia en los siguientes problemas: el retraso en los plazos de entrega; requerimientos no cumplidos; el abandono del desarrollador al cliente después de haber entregado el producto; el *software* no arroja los resultados esperados; y, los errores presentados después de entregar el producto [2]. Estos problemas generan consecuencias que han llevado a varias empresas a la pérdida de grandes sumas de dinero; a problemas legales tanto para el desarrollador como para el cliente por el incumplimiento en el plazo de tiempo de los contratos; y, a la inconformidad del cliente con el producto entregado, los cuales hacen que el desarrollo de software se vea afectado por la reputación que lo precede.

Estos problemas se presentan porque no se tiene en cuenta la verdadera dimensión del trabajo; no se sabe estimar y organizar el tiempo que requiere un proyecto; no existe un entendimiento claro porque no se





siete  
91912

establecen reglas y parámetros en la comunicación entre el desarrollador y el cliente; y, se hacen promesas que el desarrollador no puede cumplir [2].

En razón de que las causas de los problemas mencionados anteriormente se podrían reducir durante el proceso de formación del desarrollador de *software*, se creó MEPS, una propuesta de mejoramiento continuo durante la formación de un desarrollador de *software*, con el propósito de ir moldeando al estudiante para que al llegar al mundo laboral cometa el menor número de los errores y tenga un conocimiento claro de los conceptos básicos de la ingeniería de *software*.

Existen modelos de calidad de proceso, tales como CMM<sup>1</sup>, CMMI<sup>2</sup> y SPICE<sup>3</sup>, orientados a las organizaciones, mas no al desarrollador como tal; y, modelos como TSP<sup>4</sup> y PSP<sup>5</sup> encaminados al mejoramiento del trabajo en equipo y personal del desarrollador, respectivamente. Sin embargo, estos modelos buscan orientar a los desarrolladores a mejorar aspectos que fueron débiles en su proceso de formación [3].



Hasta el momento se ha trabajado en dos fases del proyecto. En la primera (PGMEPS 1<sup>6</sup>) se desarrollaron tres productos. Un mapa conceptual en donde se plasmó el proceso investigativo realizado para obtener conocimiento de conceptos como procesos de *software*, ciclos de vida, modelos de calidad, métricas y estimación. Un ambiente *software* cuyo propósito es apoyar los objetivos de formación propuestos por MEPS, presentando información a los usuarios que permita visualizar el mejoramiento evolutivo en cada uno de los niveles de MEPS. Por último, se desarrolló la herramienta de gestión de tiempos que tiene como objetivo presentarle al estudiante, gráficamente, la información para visualizar el tiempo que invierte en actividades cotidianas y en su proceso formativo.



El trabajo realizado en la fase 2 pretende complementar el adelantado en la fase anterior, haciendo énfasis en los procesos. Por esta razón, se centró en dos conceptos: procesos y modelo de calidad de proceso. Con el conocimiento adquirido se refinó el modelo MEPS existente y se inició el desarrollo de la herramienta de planificación.

En este artículo se hace una descripción de la metodología utilizada durante todo el proyecto de grado y se explican los productos ob-





tenidos: el marco teórico; el modelo MEPS propuesto con cada uno de sus niveles y líneas; el mapa conceptual con los conceptos más relevantes de la ingeniería de *software*; el ambiente *software* y la herramienta de planificación. Después se presenta el horizonte que podría seguir el grupo que continúe con el proyecto y, por último, se muestran las conclusiones.

## Desarrollo

Para la realización del trabajo de la segunda fase de este proyecto, se definieron dos etapas. En la primera se enfatizó en la parte teórica y conceptual del proyecto, y en la segunda se trabajó en el ambiente *software* y en el desarrollo de la herramienta de planificación.

La primera etapa estuvo compuesta por tres actividades generales: la primera consistió en adueñarse del marco teórico, el mapa conceptual y el modelo MEPS, desarrollados en la primera fase del proyecto [3]; en la segunda se inició la investigación de procesos y modelos de calidad de proceso; estos se utilizaron en la tercera actividad, puesto que a partir del conocimiento adquirido se refinó el modelo MEPS definiendo su arquitectura y detallando alguno de sus elementos.

El plan de trabajo para la segunda etapa se centró en el ambiente *software* y el desarrollo de la herramienta de planificación. Para esto se definieron dos actividades: la primera consistió en apropiarse del ambiente software y de la herramienta de gestión de tiempos (MyTimeTool<sup>7</sup>) desarrollada en la primera fase del proyecto (PGMEPS 1), y en la segunda actividad se hizo la construcción de la herramienta de planificación, considerando dos ciclos de desarrollo. En el primero se realizó el caso de uso de gestionar proyecto y en el segundo se desarrolló el caso de uso de gestionar proceso y mantener interfaz.

## Resultados

Como ya se mencionó, el equipo responsable de la segunda fase del proyecto enfocó su trabajo en los procesos y modelos de calidad de proceso para complementar el trabajo investigativo y proponer una mejora al modelo evolutivo desarrollado en la primera fase. Al finalizar la segunda etapa del proyecto, se presentan como resultados el mapa conceptual derivado de la investigación, el modelo evolutivo propuesto, la arquitectura del ambiente *software* que apoya el modelo MEPS y la herramienta de planificación.

## Mapa conceptual

El proceso investigativo para la extensión del mapa conceptual de MEPS se centró en dos de los conceptos más importantes de la ingeniería de *software*: procesos de desarrollo y modelos de calidad de proceso. Uno de los resultados de esta investigación fue un mapa conceptual mucho más completo en cada uno de los conceptos investigados.



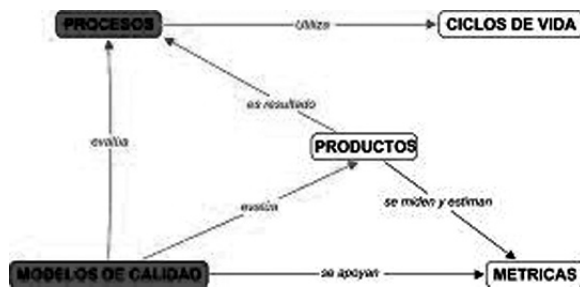


Figura 1. Mapa conceptual.

En la figura 1 se destacan los dos conceptos que se trabajaron (procesos y modelos de calidad), así como las relaciones que se consideraron en la investigación. De los dos conceptos se desarrollaron mapas conceptuales internos que complementan el trabajo realizado por el equipo que efectuó la primera fase del proyecto.

El aporte que se hizo para el concepto de procesos se centró en la definición de proceso y en la investigación de metodologías ágiles [4] [5], clásicas y dirigidas por modelos. Para cada uno de los tipos de metodologías mencionadas anteriormente se desarrolló un mapa conceptual que representa su definición. En la figura 2 se pueden observar los tres tipos de metodologías considerados [6].

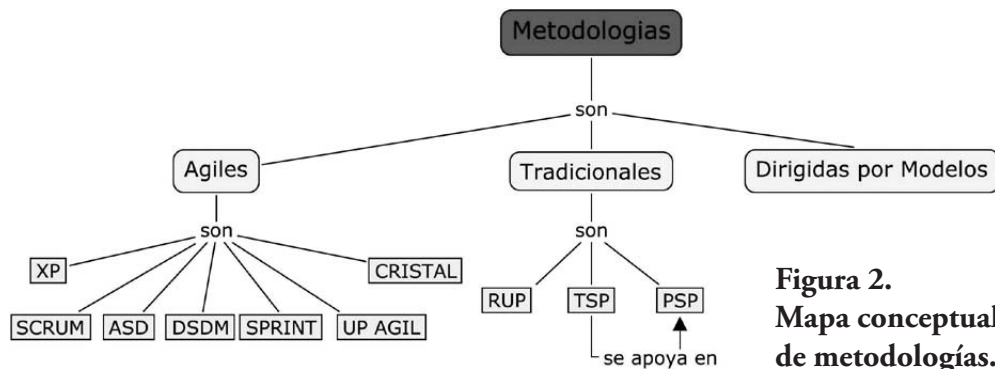


Figura 2. Mapa conceptual de metodologías.

En cuanto a las metodologías dirigidas por modelos, se desarrolló un mapa conceptual en el que se explican el proceso y cada una de las tecnologías utilizadas en este tipo de metodologías. En la figura 3 se puede ver el mapa conceptual que representa esta clase de metodologías (ver Figura 3 página 72).

El aporte que se hizo con respecto a los modelos de calidad de proceso fue la profundización de los principales modelos de calidad que apoyan las organizaciones (CMM [7], CMMI, SPICE) [4], así como los de mejoramiento personal y de equipo (PSP [9], TSP [8] [10]). Cada uno de los modelos de calidad tiene su mapa conceptual asociado. En la figura 4 se puede observar el mapa conceptual principal para los modelos de calidad de proceso, donde se relacionan todos los modelos mencionados anteriormente (ver Figura 4 página 72).



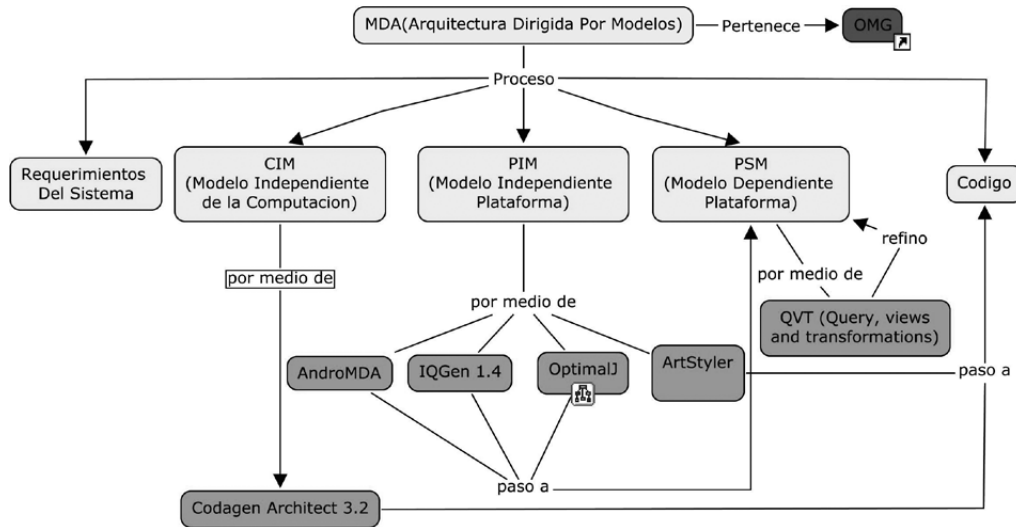


Figura 3. Mapa conceptual de metodologías dirigidas por modelos.

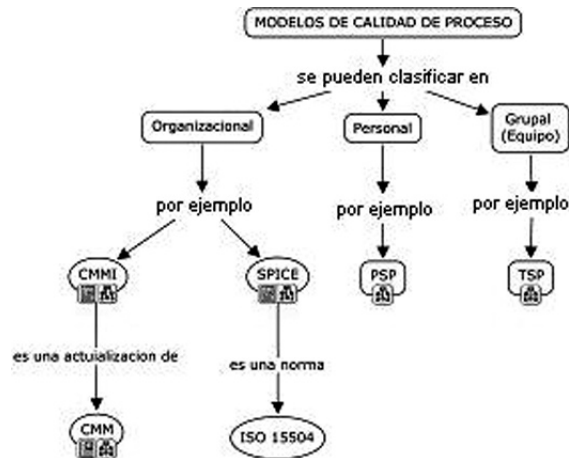


Figura 4. Mapa conceptual de modelos de calidad de proceso.

## Modelo evolutivo para procesos software

MEPS tiene como objetivo ser una guía que pueda implementarse durante el proceso de formación en el área de desarrollo de *software* de un estudiante de ingeniería de sistemas, donde este podrá conocer el estado y evolución de su proceso, a medida que logra los objetivos de formación planteados en el modelo MEPS. Con la investigación realizada y el objetivo del proyecto en mente, se trabajó en el diseño de la segunda versión del modelo. Se plantearon propuestas basadas en las experiencias vividas por los integrantes





del grupo durante su proceso de formación y en una revisión del proceso formativo en otras universidades [11] [12] [13].

El modelo MEPS propone cuatro niveles consecutivos (MEPS0, MEPS1, MEPS2, MEPS3); cada uno de ellos tiene esquematizado su objetivo de formación en el cruce con cada una de las líneas propuestas. A medida que el desarrollador vaya logrando el objetivo de formación por nivel, irá evolucionando al siguiente nivel del modelo. MEPS propone seis líneas: características del proyecto, tamaño del producto, ciclos de vida, criterios de calidad, hábitos generales y principios. Para cada línea se sugiere un lineamiento en cada nivel, que ayudará a cumplir el objetivo de formación de cada uno de ellos.

En la segunda fase se modificó el modelo MEPS refinando las cinco líneas ya establecidas, durante la primera etapa del proyecto, con los dos conceptos investigados (procesos y modelos de calidad de proceso). Básicamente, se hicieron los siguientes cambios en las líneas: se dividió la línea llamada “Tamaño y tiempo” en dos nuevas líneas “Características del proyecto” y “Tamaño del producto”, puesto que se estaban mezclando dos conceptos muy diferentes, el de proyecto y el tamaño del producto. El otro cambio que se hizo fue eliminar la línea de “Hábitos específicos” porque se decidió incluirla en la línea de “Hábitos generales”. Y, por último, se agregó una nueva línea llamada “Principios”, debido a que en la investigación de metodologías se encontró que es importante tener algunos principios básicos para orientar el desarrollo de *software*. En cuanto a los cambios que se hicieron a los niveles, se agregó uno nuevo, “MEPS3”, en la medida en que se consideró que hacía falta un nivel para que el desarrollador terminara de formarse y desarrollara proyectos un poco más grandes de los establecidos en MEPS2. En la figura 5 se puede observar una matriz que representa la estructura del modelo propuesto, con sus respectivos niveles y líneas.

**Figura 5.  
Modelo  
evolutivo  
MEPS.**

		NIVELES			
		Meps0	Meps1	Meps2	Meps3
LÍNEAS	Características del proyecto	1-2 personas 1-4 semanas 15 h/semana	2-3 personas 4-6 semanas 20 h/semana	3-4 personas 6-8 semanas 25 h/semana	4-5 personas 8-10 semanas 30 h/semana
	Tamaño del producto	10-500 líneas	501-1.000 líneas	1.001-3.000 líneas	3.001-... líneas
	Ciclo de vida	Cascada	Iterativo e incremental	Espiral	Desarrollo evolutivo
	Criterios de calidad	Funcionalidad	Usabilidad	Mantenibilidad	Confiabilidad
	Hábitos generales	Gestión de tiempos	Gestión de defectos	Gestión de riesgos	Gestión de proyectos
	Principios	La simplicidad ante todo	El cliente siempre debe quedar satisfecho	La comunicación interna del equipo y con el cliente debe ser primordial	Cualquier proyecto podría ser posible





## Ambiente software

En el modelo MEPS se definen diferentes objetivos de formación que el desarrollador debe ir logrando paulatinamente. A medida que un desarrollador evoluciona en su proceso de formación, la información obtenida crece bastante rápido y por tal motivo visualizar su avance es difícil. Por esto se debe tener un ambiente *software* que permita almacenar y manejar la información, presentando al desarrollador cifras o parámetros que le indiquen cómo va evolucionando en cada una de las líneas y niveles del modelo.

La estructura del ambiente consta de ocho herramientas, entre estas la que se encargará de integrar a las demás, denominada herramienta de monitoreo MEPS.

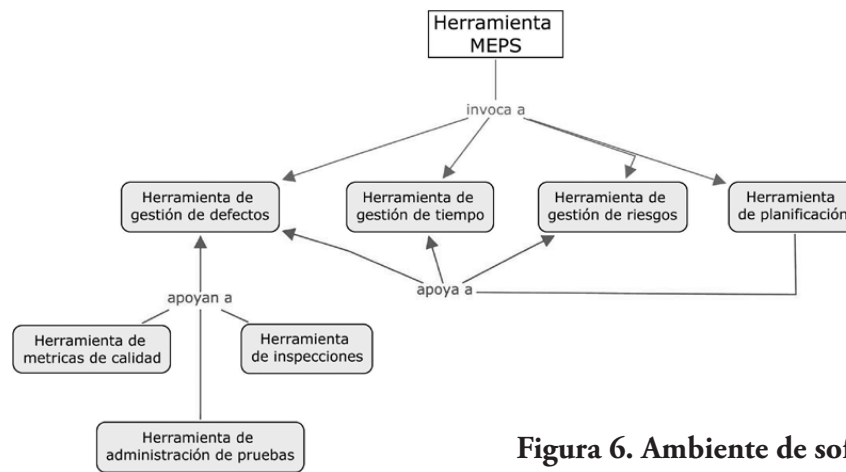


Figura 6. Ambiente de software.

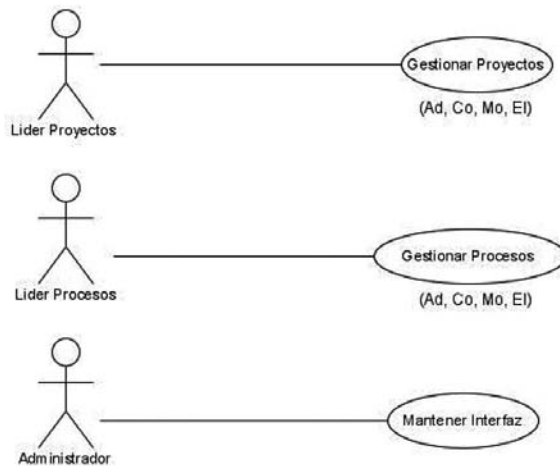
En la figura 6 se puede observar la manera como está conformado el ambiente *software* y la forma como las herramientas interactúan y se apoyan entre sí. Las herramientas de gestión de tiempos, defectos y riesgos, que apoyan el objetivo de la línea de hábitos generales en los niveles MEPS0, MEPS1 y MEPS2, respectivamente, y la herramienta de planificación que apoya a MEPS3.

Por otro lado, la herramienta de métricas de calidad apoya la línea de criterios de calidad durante todos los niveles, y junto con las herramientas de manejo de defectos y la de administración de pruebas apoya a la herramienta de gestión de defectos.

## Herramienta de planificación

En esta etapa del proyecto se desarrolló la primera versión de la herramienta de planificación, cuyo objetivo es permitirles a los líderes de proyecto la planeación de los mismos. Además de la funcionalidad mencionada anteriormente, el administrador de la herramienta tendrá la posibilidad de personalizar la interfaz a su gusto.





**Figura 7.**  
**Diagrama de casos de uso.**

El caso de uso de gestionar proyectos tiene el propósito de permitirle al usuario ingresar sus proyectos, la estructura de descomposición de trabajo (WBS<sup>8</sup>) y las actividades; además de esto, el líder de proyectos podrá asignarle los recursos humanos y el tiempo que necesita cada una de las actividades que conforman su proyecto.

El caso de uso de gestionar procesos le permitirá al líder de procesos administrar los procesos estándares, que proponen algunas metodologías conocidas (XP<sup>9</sup>, RUP<sup>10</sup>) en el desarrollo de software. El líder de proyectos podrá tener una guía para poder crear sus propios proyectos, incluyendo las estructuras de descomposición de trabajo que proponen estos procesos estándares. En cuanto al caso de uso de mantener interfaz, lo que se quiere lograr es que el administrador de la herramienta pueda modificar la apariencia de la herramienta y personalizarla a su gusto; cabe anotar que esta funcionalidad no forma parte esencial de la herramienta de planificación, sino es un caso de uso de soporte.

## Trabajo futuro

MEPS sólo está en su segunda versión; debe evolucionar en la misma forma en que nació: de las experiencias de un grupo de estudiantes y personas involucradas en el proceso de formación.

Se debe continuar complementando el mapa conceptual de la misma manera que en esta segunda fase del proyecto, enfocándose en uno o más conceptos para hacer la respectiva investigación y, después con ese conocimiento adquirido, complementar el mapa conceptual para que este producto sea cada vez más robusto y de gran ayuda tanto en la evolución de MEPS como para el desarrollador que aplique el modelo.

Infortunadamente, no se alcanzó a cumplir con uno de los compromisos del equipo, el cual era dejar la herramienta de gestión de tiempos en producción. Es importante que el equipo que continúe con el proyecto se encargue de dejar en producción las dos herra-







mientas creadas hasta el momento, puesto que si se siguen construyendo herramientas y el estudiante no usa ninguna, se va a perder el objetivo del proyecto que es construir un ambiente *software* que apoye al desarrollador en su proceso de formación.

## Conclusiones

Al lograr los objetivos de formación en cada uno de los niveles del modelo MEPS, el estudiante aprenderá a corregir sus fallas, conocer sus virtudes y tener control sobre su evolución.

Actualmente algunos de los estudiantes, por no decir todos, no se interesan mucho por conocer cómo van mejorando en su proceso de formación con el paso del tiempo. Tal vez el desarrollador durante su proceso de formación no vea la gran importancia que esto tiene; por tal razón debe aprender que al controlar su evolución y conocer sus fallas y virtudes como desarrollador, podrá realizar proyectos de mejor calidad y usar su tiempo en una forma eficiente.

Además de esto, los estudiantes que apliquen el modelo se podrán apoyar y consultar el mapa conceptual para obtener un conocimiento claro de los conceptos básicos de la ingeniería de *software*. Dicho modelo nos proporcionó información mucho más estructurada de lo investigado y nos ayudó a ver los resultados del proceso de investigación.

En cuanto al ambiente *software* es importante que el desarrollador lo use para que pueda tener control sobre su evolución y sea consciente de los errores que cometió en el pasado para que los evite en el futuro.

## Referencias

- [1] Pressman, R.S. (1997), "Ingeniería de software. Un enfoque práctico", cuarta edición, McGraw-Hill.
- [2] Wikipedia, "Crisis del software" [en línea], disponible en [http://es.wikipedia.org/wiki/Crisis\\_del\\_Software](http://es.wikipedia.org/wiki/Crisis_del_Software)
- [3] Becerra Melo, R. et all. (2006) "Libro de proyecto MEPS" [trabajo de grado]. Colombia, Escuela Colombiana de Ingeniería Julio Garavito.
- [4] Hurtado, A. y Bastarrica, (2005) "Hacia una línea de procesos ágiles:Agile SPsL" [en línea], Universidad del Cauca, disponible en [www.dcc.uchile.cl/~cecilia/papers/AgileSPsL.pdf](http://www.dcc.uchile.cl/~cecilia/papers/AgileSPsL.pdf)
- [5] Ronald, J. (2000), "What is XP?" [en línea], disponible en: [www.xprogramming.com/xpmag/whatisxp.htm](http://www.xprogramming.com/xpmag/whatisxp.htm).
- [6] Becerra Melo, R. et all. (2005) "Artículo de revisión técnica, MyPSP", Bogotá, Escuela Colombiana de Ingeniería Julio Garavito.
- [7] Software Engineering Institute, "Capability Maturity Model for Software (CMM)" [en línea], disponible en: <http://www.sei.cmu.edu/cmm/>





- [8] Watts S, H. (1999) "Introduction to the Team Software Process". The SEI Series in Software Engineering
- [9] Disney, A. y Johnson, P. (1998) "Investigating Data Quality Problems in the PSP" [en línea] . Sixth International Symposium on the Foundations of Software Engineering (SIGSOFT'98), Orlando, FL., November, 1998, disponible en: <http://www2.umassd.edu/SWPI/PersonalSoftwareProcess/PSP.html>
- [10] Centro de Investigación en Matemáticas, "TSP" [en línea], disponible en: <http://www.cimat.mx/ingsoft/seminario/TSPInfo.html>
- [11] Universidad Nacional de Colombia, "Plan de Estudios para la Carrera de Ingeniería de Sistemas" [en línea], disponible en: [http://www.unal.edu.co/dis/informacion/programas/plan\\_estudios.html](http://www.unal.edu.co/dis/informacion/programas/plan_estudios.html).  
Pontificia Universidad Javeriana. "Plan de estudios Ingeniería de sistemas" [en línea], disponible en: [http://fing.javeriana.edu.co/ingenieria/ing\\_sistemas/planAnterior.htm](http://fing.javeriana.edu.co/ingenieria/ing_sistemas/planAnterior.htm).
- [13] Universidad Libre. "Ingeniería de sistemas y computación" [en línea], disponible en: [http://www.unilibre.edu.co/facultades/Ingenieria/fac\\_ingenieria.htm](http://www.unilibre.edu.co/facultades/Ingenieria/fac_ingenieria.htm).

## Notas de pie de página

<sup>1</sup> Capability Maturity Model.

<sup>2</sup> Capability Maturity Model Integration.

<sup>3</sup> Software Process Improvement Capability Determination.

<sup>4</sup> Team Software Process

<sup>5</sup> Personal Software Process.

<sup>6</sup> Proyecto de grado Modelo Evolutivo para Procesos Software, fase 1, 2005-2006.

<sup>7</sup> MyTimeTool: Herramienta para la gestión de tiempo desarrollada por PGMEPS 1 en la primera etapa del proyecto.

<sup>8</sup> Work Breakdown Structure (estructura de descomposición de trabajo o estructura desglosada de trabajo).

<sup>9</sup> Extreme Programming.

<sup>10</sup> Rational Unified Process

**Andrés Mauricio Bernal P.** Estudiante de último año del Programa de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito. Conocimiento e interés en investigación de nuevas tendencias en ingeniería de software y en desarrollo de productos software. Responsable, disciplinado y alegre.

**Alejandro Alfonso Perea L.** Estudiante de último año del Programa de Ingeniería de Sistemas de la Escuela Colombiana de Ingeniería Julio Garavito. Capacidad de realizar labores relacionadas con el desarrollo de aplicaciones web y con proyectos de ingeniería de software, en general. Habilidades en investigación y comunicación.

