

# Alternativa para nuevas problemáticas

Carlos Alberto Cortés Carrillo

Juan Pablo Giraldo Rendón

*Estrategia de apoyo a revisiones técnicas formales en el proceso de desarrollo de software.*

El advenimiento de otras formas de aplicación de los procesos de la ingeniería de *software* a nuevos medios como el desarrollo web, aplicaciones multimediales, aplicaciones geográficas entre otros, nos obligan a estar preparados con estrategias de apoyo a las labores de verificación y validación en el proceso de desarrollo. Por ello, este texto propone una alternativa para esta problemática.

El ejercicio del desarrollo de *software*, es una de las actividades asociadas a la vasta disciplina de la informática, en ella surgen múltiples dimensiones que absorben conocimiento y se fortalecen como parte de nuevos paradigmas en construcción, uno de los más actuales y con mayores aportes es la ingeniería de *software*.

De los aportes realizados a la Ingeniería de Software, por incontables personalidades, a través de su recorrido histórico se develan herramientas, técnicas, métodos, metodologías, estrategias de medición, modelos, entre otros medios, que aportan a nuevos y experimentados practicantes, de elementos que les permiten obtener resultados acordes con políticas y lineamientos de calidad, asociados a los productos *software*.

Estos nuevos aportes inician su avatar ante la necesidad de construir *software* de manera adecuada para las organizaciones, recordemos los primeros desarrollos realizados de manera estructurada con herramientas de programación que no permitían ejercer estrategias de modularidad, reutilización, entre otras caracterís-

ticas de calidad, asociadas a los productos *software*.

La refinación realizada desde la programación estructurada, pasando a las metodologías de análisis y diseño estructuradas, la aplicación de ciclos de vida, el cambio al paradigma del objeto, tanto en sus metodologías como en los lenguajes de programación, y llegando hoy a los modelos y propuestas asociadas a la ingeniería web, permite crear alternativas variadas que acompañen los procesos de desarrollo, para dar garantía de calidad a los productos finales.

Garantía esta que se encuentra mediada por los participantes en los procesos de desarrollo, cualquiera que sea el ciclo de vida, metodología y herramientas aplicadas por el grupo de trabajo, se requiere de condiciones mínimas de gestión al proyecto y a los procesos, por ello toma fuerza la generación de estrategias que permitan apoyar los procesos de Verificación y Validación (V&V), de los resultados en cada una de las etapas del desarrollo de *software*.

### **Validación y verificación en el proceso de desarrollo de software**

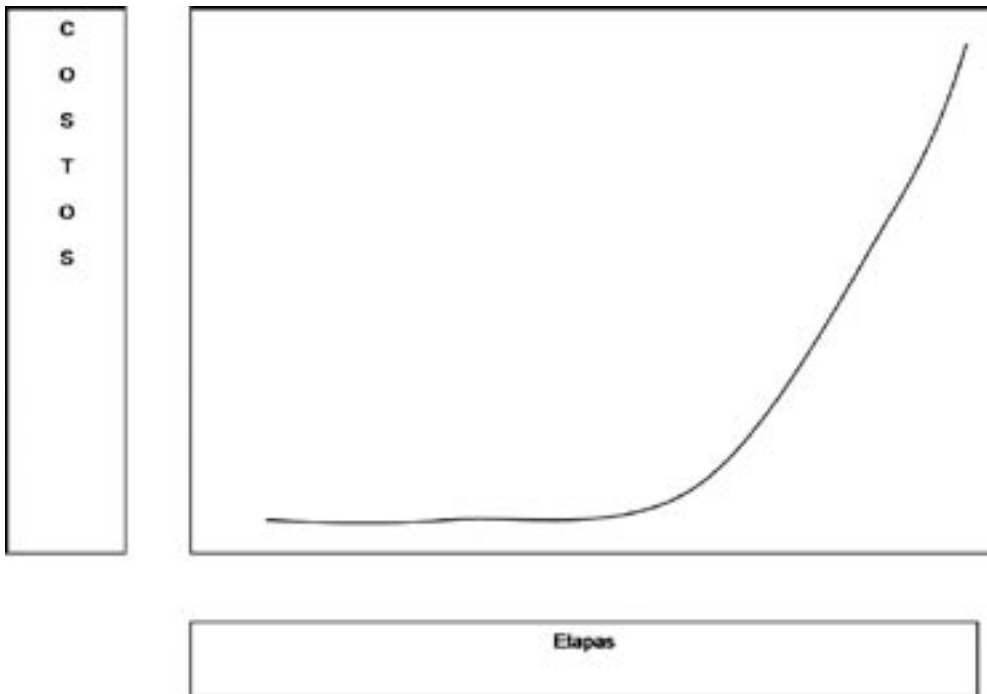
En una versión sucinta la calidad en la ingeniería del *software* es un grupo de características que representan la efectividad y la eficiencia de un sistema de información. Es importante enfatizar en dos puntos.

Un *software* de calidad debe ser eficaz; es decir, que debe realizar las funciones establecidas, debe ser amigable. Un usuario debe utilizar el *software* porque produce resultados confiables, realiza todas las operaciones que se requieren, ejecuta las operaciones en un tiempo aceptado y es fácilmente usado por el grupo de usuarios a quien esté dirigido.

Un *software* de calidad debe ser eficiente; es decir, el costo de su desarrollo tomando todos los recursos y el costo de su operación deben ser tales, que las organizaciones involucradas en su desarrollo y uso obtengan el máximo beneficio o por lo menos un aprovechamiento aceptable en un período de tiempo establecido.

Para lograrlo es necesario planear validaciones y verificaciones, que permitan disminuir la histórica curva de valoración de costos económicos, con base en la etapa en la cual se encuentra una falla en el *software* desarrollado. (ver figura 1)

Los procesos de validación y verificación son herramientas complementarias y obligadas, al momento de dar gestión a los proyectos de *software*, considerando que estos elementos no son los mismos, y para dar claridad a su diferenciación, se presenta una aclaración simple.



*Figura 1. Costo error vs. etapa de desarrollo*

**Validación:** ¿Estamos construyendo el producto correcto?

**Verificación:** ¿Estamos construyendo el producto correctamente?<sup>1</sup>

La validación se da como elemento obligado en las diferentes etapas del ciclo de vida y esta asociada a la implantación y control del sistema.

La implantación está presente en dos fases generales:

- Construcción del nuevo sistema.
- La entrega del nuevo sistema.

Estos elementos van relacionados principalmente con las fases de construcción y entrega y en menor proporción con las demás del ciclo de vida.

Todas en función de:

- Alcanzar los propósitos y objetivos propuestos.
- Completar las tareas y actividades a realizarse.
- Gestionar la secuencia y solapamiento de actividades.
- Aplicar técnicas seleccionadas.
- Administrar de manera adecuada el tiempo invertido.

Por lo tanto, el proceso de validación se define como **el acompañamiento a la construcción del nuevo sistema y la entrega de dicho sistema a producción.**<sup>2</sup>

Para alcanzar este propósito es necesaria la verificación y guía a través de cuatro fases, las cuales se aplican en el proceso de desarrollo, obligando la presencia de responsables<sup>3</sup> del grupo de trabajo elegido para alcanzar el objetivo, estas son:

Fases	Responsables
Construir y probar redes y bases de datos	Diseñador
Construir y probar programas	Programadores
Instalar y probar el nuevo sistema	Programadores
Entregar el sistema para explotación	Analistas

*Tabla 1. Fases y responsables (V&V)*

Cada una de estas fases está compuesta por actividades, las cuales son llevadas a cabo por los responsables indicados en la tabla.

**Fase 1. Construir y probar redes y bases de datos.** En esta fase es necesario verificar la existencia de redes y bases de datos, puede presentarse que estén implantadas y en funcionamiento; cualquiera que sea el caso, es necesario garantizar las condiciones para el funcionamiento del sistema.

- **Actividad 1.** Valorar, verificar y detallar las condiciones de red necesarias.
- **Actividad 2.** Verificar, contrastar y medir las necesidades de las bases de datos.

**Fase 2. Construir y probar programas.** Los objetivos de esta fase son: Planear el desarrollo y pruebas del sistema, y desarrollar programas que respondan a las necesidades. Para esta fase se encuentran definidas dos actividades que se desprenden de la aplicación de métodos de análisis y diseño, y del uso de ciclos de vida.

- **Actividad 1.** Plan de programación. En este se incluyen tres etapas, en las cuales se ve involucrada la toma de decisiones por parte del personal responsable, es decir los programadores.
  - Etapa 1. Revisión del diseño.
  - Etapa 2. Organización del equipo de programación.
  - Etapa 3. Plan de programación detallado.

- **Actividad 2.** Escribir y probar programas. En esta actividad se determinan :

- Componentes a ser reutilizables.
- Documentación de los programas (Revisiones)
- Recomendaciones y requisitos de calidad

**Fase 3. Instalar y probar el nuevo sistema.** Esta responsabilidad recae sobre los programadores, encargados de determinar las características finales del sistema. A este grupo de trabajo, se le asignan cuatro actividades que permiten llevar a buen término esta fase.

- **Actividad 1.** Instalación y prueba del sistema.

- **Actividad 2.** Integración. Es necesario se evalúe el montaje y ejecución en maquinas de bajo desempeño, posibles casos de error y sus respuestas.

- **Actividad 3.** Prueba completa. El caso a ser verificado es la necesidad que el sistema funcione como un conjunto, a través de un grupo de datos de prueba seleccionados y determinados, tanto en el diseño como en la codificación.

- **Actividad 4.** Conversión y entrega. En este caso es necesario se garanticen las conversiones de bases de

datos, archivos y todos los elementos físicos y lógicos de manejo de información.

**Fase 4. Entregar el sistema para explotación.** Este es el paso final donde se lleva el sistema a la organización y a los usuarios; es allí donde el sistema adquiere la característica de usable. Esta responsabilidad entregada a los analistas está compuesta por cuatro actividades :

- **Actividad 1.** Instalar archivos y base de datos.

- **Actividad 2.** Capacitar. Para este caso es necesario tener presente: Documentación apropiada, Manuales de usuario, Capacitación o formación del usuario final.

- **Actividad 3.** Definir un plan de conversión al nuevo sistema. Para este paso se plantean estrategias, como: total, en paralelo, por casos, por puestos de trabajo, por etapas entre otros.

- **Actividad 4.** Evaluar. En este punto se evalúa el proyecto y el proceso.

Cumplir con estas condiciones necesariamente requiere de la intervención decidida de la totalidad del equipo de trabajo, y la estructuración de las actividades asociadas a estas revisiones; así se plantea una alternativa pensa-

da desde la visión de la “revisión entre pares”<sup>4</sup> y los procesos de “Aseguramiento de calidad en el proceso” de CMMI (Capability Maturity Model Integration)<sup>5</sup>, esta propuesta se presenta a través de

diagramas de casos de uso y un diagrama de flujo final.

Una mirada a la revisión técnica a ser aplicada se presenta en la siguiente imagen:

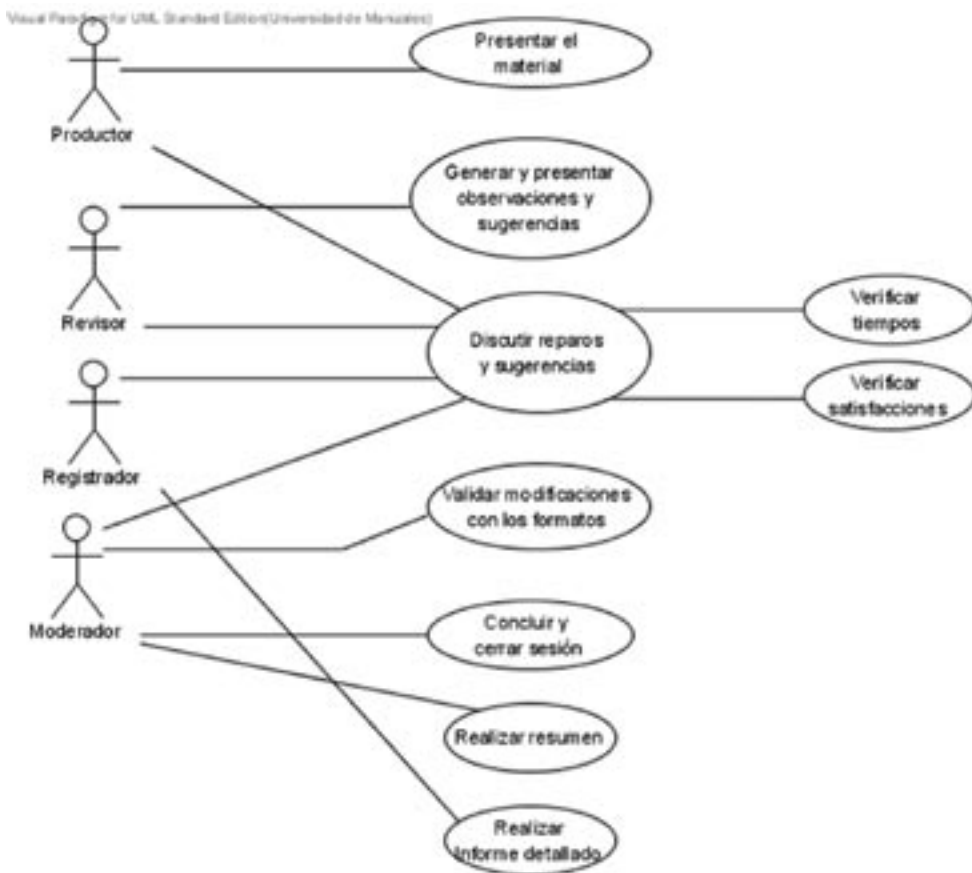


Figura 2. Caso de uso: Aplicar revisión técnica

La figura 2. presenta las siguientes particularidades: Los productores, son cada uno de los responsables de las fases que se describen previamente; los revisores, son un grupo seleccionado de la totalidad del equipo de trabajo (máximo cinco personas); el

registrador, es la persona encargada de documentar todas las acciones, y observaciones asociadas a la revisión que se realiza; el moderador, es el responsable de las acciones totales de la revisión desde su planeación, hasta la valoración final de resultados.

Para dar claridad acerca de las formas y procederes al interior de la revisión, se presentan los diagramas de

las acciones asociadas a la revisión y de las responsabilidades de cada uno de los actores.

Visual Paradigm for UML, Standard Edition (Universidad de Medellín)



Figura 3. Caso de Uso: Preparar la revisión

Este conjunto de procesos visto de esta forma plantea alternativas de seguimiento y control, pero solo aparecen como actividades que se pueden desarrollar sin un orden específico, por ello, la revisión por pares plantea tres actores principales o procesos a ser detallados, así:

1. Preparar la revisión entre iguales
2. Conducir la revisión entre iguales
3. Analizar los datos de la revisión<sup>6</sup>

Todas las actividades anteriores deben estar enmarcadas en estas acciones, para darle claridad, se presenta como alternativa la siguiente estrategia para cualquier revisión técnica a ser tenida en cuenta en los diferentes procesos descritos.

En el diagrama de flujo se hace notorio el uso de numerales en cada una de las actividades presentadas, estas numeraciones están asociadas a las tres actividades principales enunciadas. De esta manera, se reconoce

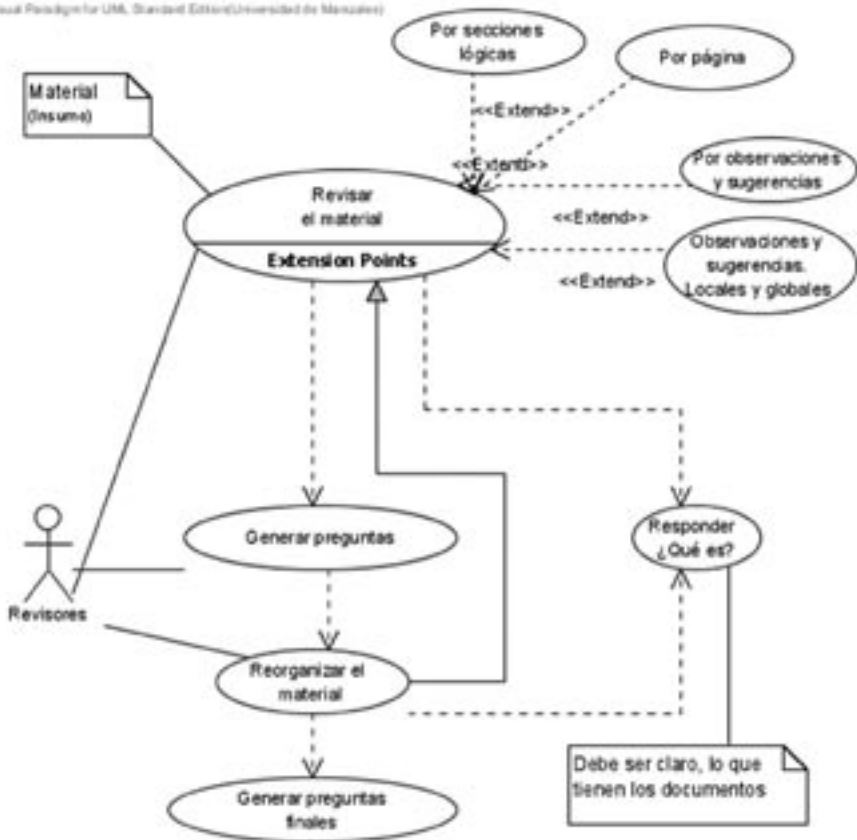


Figura 4. Caso de Uso: Revisar documentación (Material entregado y provisto por los productores)



Figura 5. Caso de Uso: Seleccionar elementos para el trabajo



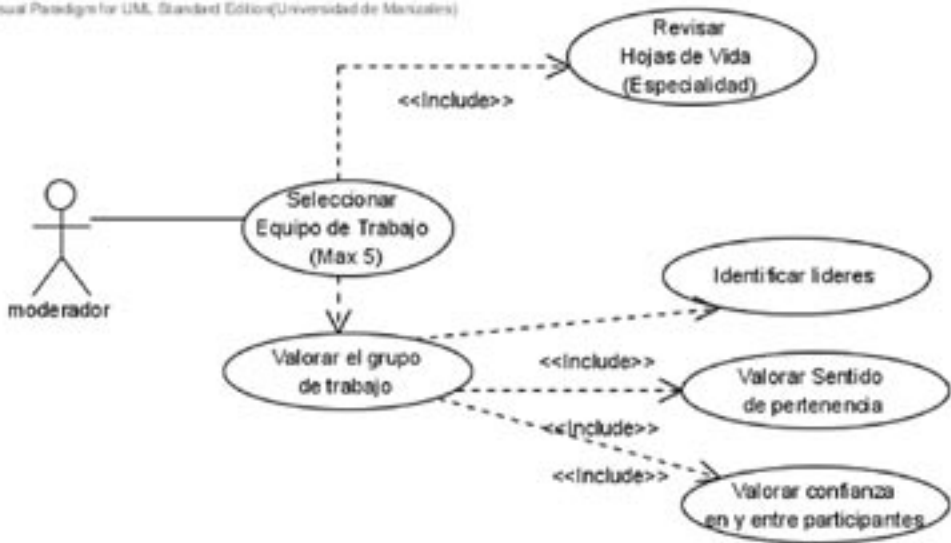


Figura 6. Caso de Uso: Seleccionar grupo para revisión

fácilmente a cuál fase está asociada cada una de las actividades, además de permitir una fácil modificación en el momento de pensar en nuevas acciones al interior de la propuesta (Ver Figura 7)

**Para finalizar se puede afirmar:**

- La estrategia presentada es un aporte aplicable a las acciones de verificación y validación en cualquiera de las etapas de desarrollo de *software* descritas, ya que se puede considerar como plantilla de revisión.
- La adecuada definición de objetivos y las listas de verificación, permiten centrar la atención de la

revisión en productos concretos, nunca en el productor.

- Esta forma de trabajo permite revisar las resignificaciones o desviaciones de los requerimientos de cliente.
- A través de la aplicación de esta estrategia se pueden valorar elementos de calidad como: cohesión, modularidad, reutilización, eficiencia, interoperatividad, uso, entre otros, ya que los resultados documentados aportan al mejor desempeño de estas características.
- La estrategia presentada es apoyo a las diferentes normas de calidad, en el aspecto de validación y verificación.

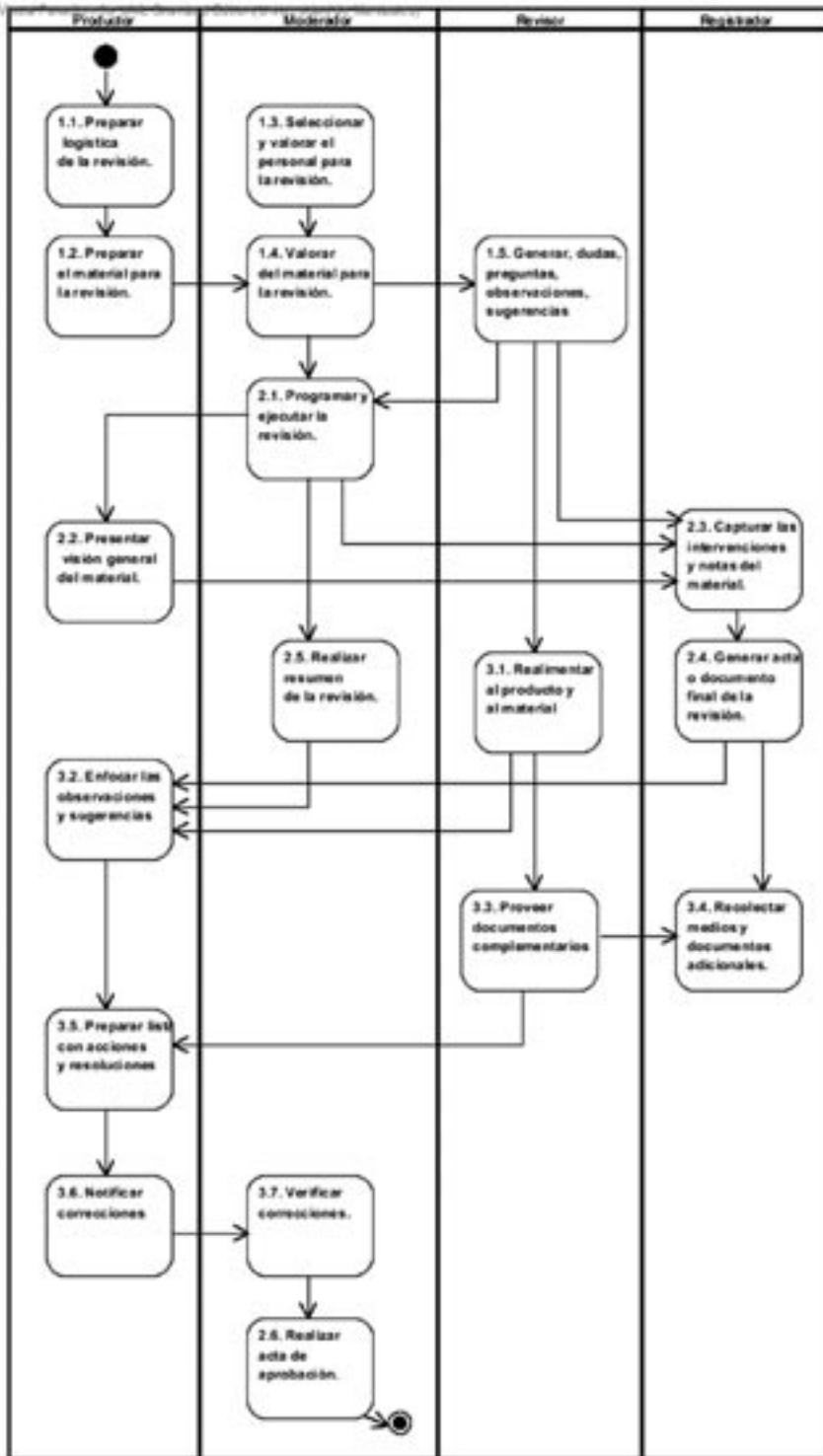


Figura 7: Diagrama de Flujo: Realizar revisión técnica

## Notas al Pie de Página

<sup>1</sup>SOMMERVILLE, Ian. *Ingeniería del Software*. 2005. Pág. 472.

<sup>2</sup>WHITTEN, Jeffrey L. *Análisis y Diseño de Sistemas de Información*. 1996.

<sup>3</sup>Mas adelante se reconocen con el nombre de *productor(es)* – Documento “Revisión entre iguales” – Universidad Politécnica de Madrid - 2000

<sup>4</sup>Documentos Cátedra de Mejora de Procesos de Software – Universidad Politécnica de Madrid - 2000.

<sup>5</sup>CMMI for development – Versión 1.2 – Agosto 2006 - Carnegie Mellon University.

<sup>6</sup>Documentos Cátedra de “Integración de modelos y métodos en la mejora del proceso y de la tecnología de software: modelo de capacidad y madurez (modelos cmmi, cmm)” Universidad Pontificia de Salamanca (Madrid) - 2006.

## Referencias

[1] CMMI Product Team - CMMI® for Development, Version 1.2 CMMI-DEV, V1.2

CMU/SEI-2006-TR-008 ESC-TR-2006-008 August 2006 - Carnegie Mellon University.

[2] Documentos de clase – Universidad Politécnica de Madrid – Mejora de procesos de Software – 2000

[3] Documentos Cátedra “Integración de modelos y métodos en la mejora del proceso y de la tecnología de software: modelo de capacidad y madurez (modelos cmmi, cmm)” Universidad Pontificia de Salamanca - 2006

[4] DONALDSON, Scoot – SIEGEL, Stanley G. *Successful Software Development*. Segunda Edición. Prentice Hall. 2001

[5] PRESSMAN, Roger S. *Ingeniería del Software, Un enfoque práctico*. Tercera edición. McGraw Hill - 1995

[6] SOMMERVILLE, Ian. *Ingeniería del Software*. Séptima Edición. Pearson Addison Wesley. 2005.

[7] WHITTEN, Jeffrey L. *Análisis y diseño de sistemas de información*. Tercera Edición - 1996

**Juan Pablo Giraldo R.** Ingeniero de Sistemas. Estudiante de doctorado Ingeniería Informática, Universidad Pontificia de Salamanca, Madrid. Docente del programa de Ingeniería de Sistemas y Telecomunicaciones. Investigador Grupo de Investigación y Desarrollo en Informática y Telecomunicaciones. Escalafonado en Colciencias, Categoría A. Investigador en la facultad de ingeniería de la universidad de Manizales. Desarrolló labores como contratista en el área informática para Ecopetrol y Terpel.

**Carlos Alberto Cortés C.** Ingeniero de Sistemas. Profesional en Ciencias de la Información y la Documentación. Magíster en Gerencia del Talento Humano. Especialista en Telecomunicaciones, Informática y Computación, Auditoría de Sistemas. Decano, facultad de Ingeniería, universidad de Manizales. Investigador Grupo de Investigación y Desarrollo en Informática y Telecomunicaciones, Escalafonado en Colciencias, Categoría A. Investigador Grupo de Investigación y Desarrollo en Geomática y Medio Ambiente, Escalafonado en Colciencias, Categoría B. Director Grupo de Investigación y Desarrollo Sociedad de la Información, Gestión e Innovación del Conocimiento, registrado en Colciencias. Coordinador especialización Telecomunicaciones. Convenio Universidad de Manizales – UNAB. Gestor Doctorado en Ingeniería Informática. Convenio Universidad de Manizales – Universidad Pontificia de Salamanca España.