



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

López Trujillo, Yucely; André Ampuero, Margarita; Infante Abreu, Ana Lilian  
Formación de roles y buenas prácticas en el trabajo por la calidad de un ingeniero informático  
Ingeniare. Revista Chilena de Ingeniería, vol. 19, núm. 3, 2011, pp. 382-395  
Universidad de Tarapacá  
Arica, Chile

Disponible en: <http://www.redalyc.org/articulo.oa?id=77221486008>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en 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

## **Formación de roles y buenas prácticas en el trabajo por la calidad de un ingeniero informático**

*Training of roles and good practice for quality work of a computer engineer*

Yucely López Trujillo<sup>1</sup>    Margarita André Ampuero<sup>1</sup>    Ana Lilian Infante Abreu<sup>1</sup>

Recibido 19 de julio de 2010, aceptado 9 de diciembre de 2011

*Received: July 19, 2010    Accepted: December 9, 2011*

### **RESUMEN**

La universidad es la responsable de formar a los profesionales informáticos que ingresan a la industria; por tanto, tiene el deber de impartir los conocimientos y desarrollar las habilidades necesarias para que los estudiantes sean capaces de trabajar, de manera individual y en equipo, de modo disciplinado y con calidad. En este sentido, para satisfacer las expectativas de la industria de software es preciso perfeccionar continuamente el proceso de formación de roles en la carrera de ingeniería informática. En este trabajo se hace referencia, específicamente, a los planes de estudio que forman a los ingenieros informáticos en las universidades cubanas. Aunque, en opinión de las autoras, la problemática identificada y su posible solución es aplicable a otros contextos. En el artículo se enuncia una estrategia para la formación de roles y hábitos de trabajo disciplinado, puesta en ejecución hace cuatro cursos académicos y se proponen acciones para perfeccionar el proceso de formación, en función del análisis realizado.

Palabras clave: Calidad de software, disciplina personal, proceso de software personal, PSP, roles, formación.

### **ABSTRACT**

*The university is responsible for training software professionals entering industry. Therefore, it has the duty to impart knowledge and develop the necessary skills for students to be able to work individually and in teams, so that they are disciplined and produces quality work. In this sense, to meet the expectations of the software industry, we must continually improve the process of the formation of roles in computer engineering career. This paper refers specifically to the curriculum to train software engineers in Cuban universities. Although, it is the opinion to the authors that the problems identified and its possible solution is applicable to other contexts. The article sets out a strategy for the creation of roles and disciplined work habits, implemented for four academic years and proposes actions to improve the training process, based on analysis.*

*Keywords: Software quality, personal discipline, personal software process, PSP, roles, training.*

### **INTRODUCCIÓN**

La industria de software tiene gran impacto en prácticamente todas las ramas del desarrollo de la sociedad. Sin embargo, a pesar de sus reconocidos resultados, aún resulta significativo el número de proyectos de software que no culminan con

éxito [1]. Entre los principales problemas se identifican: insuficiente calidad de los productos finales, estimaciones inexactas de la duración de los proyectos, retrasos en la entrega de los productos, descontrol de los costos de desarrollo y mantenimiento de los productos, pobre definición de los requisitos, descontrol de los requerimientos de cambios [2].

---

<sup>1</sup> Facultad de Ingeniería Informática. Instituto Superior Politécnico José Antonio Echeverría (CUJAE). Calle 114 N° 11901 - 119 y 127. La Habana, Cuba. E-mail: ylopez@ceis.cujae.edu.cu; mayi@ceis.cujae.edu.cu; ainfante@ceis.cujae.edu.cu

Las principales dificultades apuntan a la no existencia de procesos de desarrollo de software bien definidos que garanticen un buen uso de los talentos y recursos con que cuentan las organizaciones y que estos procesos mejoren de forma continua. Sin embargo, para lograr las mejoras deseadas resulta vital entrenar al personal de forma tal que sea capaz de desempeñar los roles que les corresponda de manera disciplinada ya sea trabajando individualmente o en equipo. La universidad debe jugar un papel protagónico en la formación del ingeniero informático en correspondencia con las exigencias que demanda la industria. Por su parte, la industria se siente presionada por mejorar su desempeño y desarrollar software de calidad, sobre todo teniendo en cuenta que en la actualidad se experimenta un crecimiento del volumen y complejidad de los productos de software que, indudablemente, ha convertido el desarrollo de sistemas en una actividad de equipo.

El personal es un factor poco formalizado en los modelos de procesos y en las metodologías de desarrollo de software, las cuales se centran más en aspectos técnicos que en los aspectos humanos [3-5]. No obstante, existen modelos líderes en el tópico de la mejora de procesos que sí se centran en los recursos humanos, el Proceso de Software Personal (PSP) y el Proceso de Software en Equipo (TSP) [1, 6-8]. Ambos procesos fueron desarrollados en el Software Engineering Institute (SEI)<sup>2</sup> por Watts S. Humphrey<sup>3</sup>. El objetivo de PSP es proveer a los individuos de mecanismos de trabajo para convertirse en miembros efectivos de equipos, partiendo de que la base para alcanzar una disciplina a nivel de equipo es la disciplina personal. TSP, por su parte, pretende formar y desarrollar equipos exitosos.

En cuanto a los cursos que se imparten en la carrera de ingeniería informática y de manera general en carreras afines, se puede afirmar que éstos se centran en cuestiones técnicas y en el desempeño individual del estudiante [9-10]. La disciplina personal y de trabajo en equipo, la formación de roles, la comunicación y el liderazgo, son temas que se abordan, en muchas ocasiones, de manera teórica

o no logran implementarse a plenitud en la ejecución de proyectos reales [11]. Las insuficiencias en la formación académica entran en contradicción con las exigencias que impone la industria (en función de las dificultades que enfrenta). Esta situación indica que es preciso trabajar desde el proceso de formación con el propósito de eliminar la brecha existente entre la academia y la industria.

La carrera de Ingeniería Informática en Cuba ha transitado por diferentes planes de estudio, todos estructurados en cinco años académicos (diez semestres). Todos los planes han exigido que la totalidad de los estudiantes, a partir del tercer año, se incorporen a la ejecución de proyectos reales, los cuales en la mayoría de los casos se desarrollan en empresas. Adicionalmente, un conjunto de estudiantes de primero y segundo año se integran a estos equipos de proyectos.

Como parte del proceso de elaboración del plan de estudio vigente (denominado “Plan D”) y en respuesta a los retos que enfrenta el proceso de formación de los profesionales de software, se elaboró una estrategia para la formación de los roles y las competencias que precisa la industria, la cual incorpora, entre otros aspectos, la introducción de las prácticas de PSP y TSP de forma incremental [11-12]. Desde la puesta en marcha del “Plan D” han transcurrido cuatro cursos académicos, por lo que en la actualidad se cuentan con estudiantes de primero, segundo, tercero y cuarto año formados con la estrategia definida. Existen además estudiantes de quinto año formados con el antiguo plan de estudios (denominado “Plan C”). Por tal motivo, se considera que se está en un buen momento para analizar cómo marcha la puesta en práctica de la estrategia.

El presente trabajo se ha estructurado de la forma siguiente: se enuncian un conjunto de conceptos importantes relacionados con la temática en cuestión, se plantea la estrategia de formación de roles y hábitos de trabajo disciplinados, se realiza un análisis de cómo ha marchado el proceso de formación de los estudiantes con la estrategia definida y, por último, se proponen un conjunto de acciones para perfeccionar el proceso de formación.

---

<sup>2</sup> Centro de investigación y desarrollo, de la Universidad de Carnegie Mellon.

<sup>3</sup> Watts S. Humphrey, científico e investigador del Software Engineering Institute (SEI). De 1959 a 1986 estuvo asociado con IBM Corporation donde fue director del departamento de proceso y calidad.

## CALIDAD DE SOFTWARE

### Hábitos de trabajo disciplinado

Humphrey define el trabajo de la ingeniería del software como “*la entrega de productos de alta calidad de acuerdo a un costo y a un cronograma fijado*”. Y precisa que hay tres aspectos claves para que se realice de manera efectiva: planificar el trabajo, trabajar acorde a un plan y esforzarse para producir productos de alta calidad [13].

Sin embargo, esto exige disciplina y resulta muy difícil para los ingenieros de software desarrollar de manera regular un trabajo personal disciplinado debido principalmente a tres razones [14]:

- La ingeniería de software no tiene tradición de ejecución personal disciplinada.
- El proceso de software no le impone una disciplina natural a los ingenieros ya que el hecho de que el diseño de software no implique producción a gran escala no ha exigido una revisión minuciosa de éste.
- A pesar de que el trabajo disciplinado en cualquier campo siempre ha exigido contar con buenos estándares y soporte competente, la industria de software carece de una adecuada formación para el desempeño de los diferentes roles.

Con vistas a lograr cambiar esta actitud es preciso que los ingenieros comprendan la necesidad de utilizar métodos disciplinados, conozcan cómo aplicarlos y constaten que el empleo de los métodos realmente mejora su trabajo. Por lo tanto, para alcanzar disciplina en el trabajo individual es preciso: usar un proceso personal definido, planificar cada tarea, registrar tiempos, tamaños y defectos, seguir la ejecución del proceso y medir la calidad del producto.

### Proceso de Software Personal (PSP)

El Proceso de Software Personal tiene como propósito ayudar a los ingenieros a organizar y planificar su trabajo, chequear su ejecución, dirigir la calidad de software y analizar y mejorar su proceso personal. Para lograrlo ofrece instrucciones, guiones, formularios y métricas, los que resultan claves para alcanzar la disciplina [15-16].

En líneas generales, durante el desarrollo de un software con PSP tienen lugar tres fases:

*Planificación:* donde se elabora un plan del ingeniero.  
*Desarrollo:* donde se lleva a cabo la construcción del software.

*Post mortem:* donde se recogen y analizan los datos y problemas presentados con vistas a proponer mejoras al proceso y mejorar futuras planificaciones.

PSP está estructurado en niveles y cada uno incorpora buenas prácticas incrementalmente, exigiendo del personal una mayor disciplina en el desempeño del trabajo individual. La Figura 1 muestra la estructura incremental de PSP.

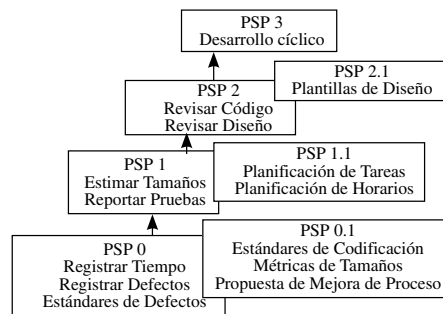


Figura 1. Estructura incremental de PSP [17].

## ESTRATEGIA PARA LA FORMACIÓN DE ROLES Y HÁBITOS DE TRABAJO DISCIPLINADOS

En el documento de la comisión nacional de carrera donde se plasma la estrategia de formación de roles [18] se realiza un análisis crítico del proceso de formación del ingeniero informático formado con el plan de estudios denominado “Plan C”, que se ha aplicado desde hace años en la facultad de Ingeniería Informática del “Instituto Superior Politécnico José Antonio Echeverría”. En el análisis se concluye que los temas de planificación, asignación de recursos, estimación de costos, tamaños y tiempos, definición y trabajo con estándares, calidad, entre otros, no son abordados de forma profunda.

Por otra parte, se analiza [18] que el Plan C está concebido de forma tal que en los primeros años se imparte un ciclo de programación donde apenas se introducen algunos aspectos relacionados con la ingeniería de software y sólo se ejercita con fuerza el rol de programador. Es a partir del segundo semestre del tercer año que se comienzan a enseñar las prácticas de la ingeniería de software, por lo

que resulta muy difícil tratar de cambiar la visión y los hábitos adquiridos por los estudiantes en los años previos.

Con el objetivo de mejorar el proceso de formación concebido en el Plan C se propuso una estrategia para lograr no sólo que los estudiantes adquieran una disciplina personal, sino que aprendan a desempeñar los roles a lo largo de toda la carrera de forma incremental, consciente y disciplinada, con vistas a poder ser miembros efectivos de un equipo de proyecto. Así, a partir del análisis del contenido de las asignaturas que se encuentran dentro del plan de estudio y con el objetivo de formar los nuevos profesionales que necesita la industria, se formularon las siguientes consideraciones como parte de la estrategia:

**Consideraciones sobre las prácticas y roles a formar en el primer año de la carrera:**

- Resulta muy difícil para los estudiantes de primer año asimilar la temática de administración de tiempo ya que se encuentran ante el desafío de aprender a programar en un ambiente de desarrollo y un lenguaje determinado y, además, no tienen idea de lo que significa un proyecto ni de su magnitud en cuanto a tamaño, tiempo de desarrollo y esfuerzo. Sin embargo, resulta factible introducir la temática de prevención de defectos, incluso desde las primeras asignaturas de programación, identificando sólo dos tipos de defectos: sintácticos y lógicos. El resto de los tipos de defectos incluidos en el estándar propuesto por PSP deberán incorporarse paulatinamente. Lo importante es darle la noción de lo que es un defecto y la importancia de registrarlo y prevenirlo. Los estudiantes desde su primera tarea deben demostrar que son capaces de depurar su código y de elaborar un reporte de defectos para que así, a partir de este momento, sean capaces de registrar defectos de forma rutinaria.
- Si bien el estudiante no tiene habilidades para realizar estimaciones de tiempo, se propone introducir su registro. Por el momento sólo será necesario que registren el tiempo que dedican a las clases, al estudio, a la realización de tareas y del proyecto de curso. El objetivo es que conozcan en qué emplean su tiempo y que comprendan lo importante que resulta registrarlo para poder administrarlo adecuadamente. Esto,

además, les permitirá desarrollar el hábito de registrar métricas y conformar su propia base de datos históricos para efectuar estimaciones en el futuro.

- También, desde el primer curso de programación, los estudiantes deben ser capaces de escribir código siguiendo un estándar definido en el lenguaje utilizado. Esta disciplina debe tornarse en un hábito por el resto de su carrera. Además, deben ser capaces de elaborar su propio estándar de estilo de código de forma tal que lo definan y apliquen en los proyectos de cursos, las prácticas profesionales y en el trabajo de diploma. Un buen estándar de codificación facilita la lectura y entendimiento del código y garantiza que las revisiones, el mantenimiento, el reuso y el proceso de depuración sea más fácil.
- Los estudiantes antes de programar no sólo deben ser capaces de esclarecer los requerimientos del programa, sino que deben realizar un diseño en base a los patrones adecuados, seleccionando de manera eficiente los tipos de datos y diseñando e implementando todas las validaciones necesarias para evitar fallas de seguridad. Además, deben ser capaces de buscar y reusar soluciones con vistas a ganar en productividad y calidad del código.
- El concepto de lista de chequeo debe ser introducido y utilizado para revisar el código tratando de encontrar defectos antes de la primera compilación.
- Los estudiantes están en condiciones de modelar procesos de la vida cotidiana actuando como analista, utilizando artefactos de apoyo como los Diagramas de Actividades. Ya en el desarrollo del proyecto están en condiciones de identificar, sin alto grado de formalidad, los requisitos funcionales y no funcionales.
- Los estudiantes también están en condiciones, al finalizar el año, de ejercitar el rol de probador. Para ello deben elaborar los casos de prueba asociados al proyecto de programación, ejecutar las pruebas y reportar los resultados obtenidos.
- Resulta importante desarrollar la habilidad de gestionar las versiones de los documentos y proyectos que desarrollan.

Por lo tanto, para un estudiante de primer año se establecen como primarios los roles de analista del negocio, diseñador, programador y probador y, como secundarios, los de analista del sistema, planificador, gestor de configuración, especialista en calidad, seguridad y soporte.

### **Consideraciones sobre las prácticas y roles a desarrollar en segundo año de la carrera:**

- En segundo año se trabaja con mayor fuerza en la formación de los roles de diseñador y programador. Aquí los estudiantes diseñan e implementan algoritmos más complejos donde el diseño preliminar les ahorrará tiempo. Por lo tanto, deben aprender a revisar el diseño tan pronto como sea posible y se sugiere el uso de listas de chequeo para facilitar las revisiones. El diseño puede ser documentado utilizando artefactos como el Diagrama de Clases aunque sin un grado de formalización elevado.
- Para enfrentar la solución de problemas de mayor complejidad deben continuar identificando los requisitos funcionales y no funcionales ahora con mayor grado de formalidad, utilizando artefactos como los Casos de Uso.
- El estudiante está listo para implantar sus soluciones, desarrollando instaladores y ayudas.
- Además, deben continuar registrando de forma disciplinada los tiempos y los defectos encontrados, utilizando estándares para codificar, probando y reportando los resultados obtenidos y controlando versiones de sus proyectos.

Por lo tanto, para un estudiante de segundo año se establecen como primarios los roles de diseñador y programador; y como secundarios, los de analista del sistema, planificador, gestor de configuración, implantador, especialista en calidad, seguridad y soporte.

### **Consideraciones sobre las prácticas y roles a desarrollar en tercer año de la carrera:**

- En este año el estudiante debe formarse como planificador. Las métricas de tamaño y los métodos de estimación deben ser introducidos en las asignaturas de ingeniería de software.
- Debe formalizarse y evaluarse con rigor el desempeño del estudiante como analista del negocio y del sistema en la realización del proyecto de curso.
- Se debe fortalecer aún más los roles de diseñador y programador. Los estudiantes deben ser capaces de realizar implementaciones en otros lenguajes, gestores, etc.

Así, para un estudiante de tercer año se establecen como primarios los roles de planificador, analista, diseñador y programador; y como secundarios, los

de gestor de configuración, implantador, especialista en calidad, seguridad y soporte.

### **Consideraciones sobre las prácticas y roles a desarrollar en cuarto año de la carrera:**

- A partir de este año los estudiantes están listos para desarrollar el rol de arquitecto de software, el cual debe formalizarse y evaluarse con rigor en la realización del proyecto de curso.
- También debe formalizarse y evaluarse con el proyecto el desempeño del estudiante como diseñador.
- El estudiante debe completar su formación como especialista en calidad. Está en condiciones de seguir todas las prácticas de PSP y de registrar métricas de su proceso. Se sugiere el uso del *Personal Process Dashboard* como herramienta de apoyo.
- Debe formalizarse aún más el rol de gestor de configuración y formarse el rol de gestor de cambios.
- El estudiante debe adquirir la disciplina para aprender a trabajar en equipo y debe ejercitarla mediante el desempeño de uno o más roles en el desarrollo de un proyecto académico.
- Para el desarrollo de las prácticas profesionales a final del año, el estudiante debe estar listo para dirigir un equipo de proyecto, formado por estudiantes de su año y/o de años inferiores.

### **Consideraciones sobre las prácticas y roles a desarrollar en quinto año de la carrera:**

- El estudiante está certificado en el desempeño de todos los roles que forma la carrera, lo que permite que en su trabajo de tesis desempeñe los roles que se precisen acorde a los objetivos del trabajo. Se establecen como roles primarios el de Jefe de Proyecto, Analista, Arquitecto, Diseñador y Programador.

En la estrategia propuesta se identifican doce roles a formar durante la carrera: Analista (negocio y sistema), Diseñador, Programador, Arquitecto, Planificador, Especialista en calidad, Especialista en seguridad, Jefe de proyecto, Implantador, Probador, Especialista en soporte y Gestor de configuración y cambios. De ellos, son secundarios: el Especialista en Seguridad, el Implantador y el Especialista en Soporte; mientras que los restantes son primarios en al menos un año académico. Se proponen entonces como roles primarios en la formación de un ingeniero

informático: el Analista, el Arquitecto, el Diseñador, el Programador, el Jefe de proyecto, el Especialista en calidad, el Planificador, el Probador y el Gestor de configuración y cambios.

Analizando la Tabla 1 se puede concluir que el Proceso de Formación para la totalidad de los roles es iterativo e incremental, alcanzándose en el último año de la carrera el cien por ciento de la formación requerida en cada uno de los casos. Hay roles cuya formación crece en una misma proporción porque es primario o secundario en la totalidad de los años donde se forman. Hay roles como: Jefe de Proyecto, Arquitecto, Programador y Diseñador que son primarios en todos los años donde se forman.

Resulta válido señalar que en la propuesta [19] se encuentra un análisis más detallado de cómo tributa cada asignatura a la formación de los roles en cada año académico precisando por cada una los objetivos fundamentales, los roles primarios y secundarios, las tareas definidas para cada rol y la forma de evaluación.

Tabla 1. Formación de roles primarios y secundarios por años.

| Roles a ejercitar<br>Primarios (P) y<br>Secundarios (S) | Año Académico |   |   |   |   |   |   |   |   |   |
|---|---------------|---|---|---|---|---|---|---|---|---|
|   | 1             |   | 2 |   | 3 |   | 4 |   | 5 |   |
|   | P             | S | P | S | P | S | P | S | P | S |
| Analista (Negocio)                                      |               |   |   |   |   |   |   |   |   |   |
| Analista (Sistema)                                      |               |   |   |   |   |   |   |   |   |   |
| Diseñador   |               |   |   |   |   |   |   |   |   |   |
| Arquitecto  |               |   |   |   |   |   |   |   |   |   |
| Programador   |               |   |   |   |   |   |   |   |   |   |
| Probador  |               |   |   |   |   |   |   |   |   |   |
| Planificador  |               |   |   |   |   |   |   |   |   |   |
| Esp. en Soporte   |               |   |   |   |   |   |   |   |   |   |
| Esp. en Calidad   |               |   |   |   |   |   |   |   |   |   |
| Esp. en Seguridad                                       |               |   |   |   |   |   |   |   |   |   |
| Gestor de Config.                                       |               |   |   |   |   |   |   |   |   |   |
| Gestor de Cambios                                       |               |   |   |   |   |   |   |   |   |   |
| Implantador   |               |   |   |   |   |   |   |   |   |   |
| Jefe de Proyecto  |               |   |   |   |   |   |   |   |   |   |

## ANÁLISIS DE LA EJECUCIÓN DE LA ESTRATEGIA DE FORMACIÓN

A partir del análisis crítico realizado al proceso de formación del ingeniero informático [18] se conciben una serie de transformaciones a realizar en el plan de estudios de la carrera, surgiendo así un nuevo plan de estudios denominado Plan D. En el Plan D, se unen las disciplinas “Técnicas de Programación de Computadoras” (TPC) e “Ingeniería y Gestión de Software”, surgiendo una nueva disciplina integradora denominada “Ingeniería y Gestión de Software” (IGS).

La nueva disciplina surge teniendo en cuenta que en la formación como programador de los graduados de la carrera, concentrada en los tres primeros años en las asignaturas de la disciplina TPC, no se podía carecer de las mejores prácticas de la ingeniería de software, de ahí que resultara imprescindible integrar los conocimientos y las habilidades de las asignaturas de estas dos áreas de conocimiento. En las asignaturas de esta disciplina se introduce y materializa la estrategia de formación de roles descrita en este artículo.

A partir del curso académico 2007-2008 se comenzó a aplicar el Plan D, en la facultad de Ingeniería Informática del Instituto Superior Politécnico José Antonio Echeverría. El plan de estudio D tiene una organización docente formada por un currículo base, un currículo propio y un conjunto de asignaturas optativas.

Dentro de las asignaturas de la disciplina “Ingeniería y Gestión de Software” que tributan directamente a la formación de roles y al trabajo en equipo se encuentra la asignatura Ingeniería de Software III. Los objetivos instructivos declarados en el Plan D para dicha asignatura son los siguientes:

1. Ejercitar habilidades en el desempeño de roles desarrollando un proyecto informático, incluyendo el rol Jefe de Proyecto.
2. Ejercitar habilidades de trabajo en equipo.
3. Evaluar las competencias de los estudiantes en la ejecución de determinados roles dentro del equipo.

La asignatura Ingeniería de Software III se imparte en el segundo semestre del cuarto año de la carrera,

en el curso 2010-2011 se impartió por primera vez a los estudiantes iniciadores del Plan D. Teniendo en cuenta los objetivos de esta asignatura, se considera que al evaluar los resultados obtenidos en su ejecución se está evaluando cómo ha marchado la formación de roles y hábitos de trabajo por la calidad en las asignaturas precedentes que tributan a este empeño.

Ejercitar y evaluar las habilidades de trabajo en equipo en la ejecución de un proyecto informático, constituyó un gran reto a la hora de organizar la asignatura. Para el logro de este objetivo se constituyeron equipos de entre cinco y seis miembros, a cada equipo se le asignó un proyecto real. Se utilizaron los proyectos reales que se desarrollan en la facultad donde participan profesores y estudiantes desde el tercer año de la carrera fundamentalmente.

En la preparación de la asignatura y en el estudio individual orientado a los estudiantes se consultaron bibliografías básicas en el tema de la gestión de proyectos, como es el caso del “Cuerpo de conocimientos para la Gestión de Proyectos”, conocido por sus siglas en inglés como PMBoK [19] y en el tema de la mejora de procesos de desarrollo de software se recomendó el estudio del Modelo de Madurez, conocido como CMMi [20].

Los roles ejercitados fueron: Jefe de Proyecto, Arquitecto, Analista, Programador, Planificador, Gestor de Configuración, Especialista de Calidad. Es de señalar que la asignatura no aportó conocimientos nuevos para la formación de los roles Arquitecto, Analista y Programador, sólo evaluó el desempeño de los estudiantes. Para los cuatro restantes roles, sí se aportaron conocimientos nuevos y se logró su ejercitación en la práctica.

Se considera que los roles (Jefe de Proyecto, Planificador, Gestor de Configuración y Especialista de Calidad) son los que más influyen en el trabajo por la calidad en un proceso de desarrollo de software y sin embargo son los que tradicionalmente no se forman de manera práctica, muchas veces solo de manera teórica. En el caso del rol Jefe de Proyecto fue la primera vez que los estudiantes tuvieron la oportunidad de desempeñarlo y, en sentido general, de experimentar el trabajo en equipo dirigidos y controlados por un jefe y con las responsabilidades bien delimitadas.

Teniendo en cuenta la novedad que representa la formación en la práctica de los roles Jefe de Proyecto, Planificador, Gestor de Configuración y Especialista de Calidad, a continuación se describen las responsabilidades asignadas a los estudiantes en el marco de esta experiencia docente.

#### *Responsabilidades del rol Jefe de Proyecto:*

1. Conformar el equipo, que implica asignar los roles y responsabilidades.
2. Dar seguimiento al plan de proyecto con vistas a que se cumplan todas las tareas y se obtengan los entregables en las fechas programadas.
3. Garantizar que se realice de forma objetiva la evaluación y autoevaluación del desempeño del rol y del aporte al equipo de todos los miembros.
4. Garantizar la adecuada comunicación y control del trabajo en el equipo.

#### *Responsabilidades del rol Planificador:*

1. Estudiar el área de proceso de Planificación de Proyecto de CMMi.
2. Estudiar en el libro PMBoK los procesos de dirección de proyecto y en especial estudiar el grupo de procesos de planificación.
3. Asimilar la herramienta Microsoft Project.
4. Apoyar al especialista de calidad de manera que garantice que en la herramienta de tiempo (excel) aparezca planificado el tiempo a dedicar a las tareas propias de cada rol.
5. Ejecutar el Plan de Proyecto utilizando la herramienta Project, incluyendo todas las tareas que indican tanto CMMi como PMBoK que debe contener un proyecto.
6. Entregar el Plan de proyecto utilizando la herramienta Microsoft Project

#### *Responsabilidades del rol Gestor de Configuración:*

1. Estudiar el área de proceso de Gestión de Configuración de CMMi.
2. Definir la herramienta a utilizar en el proyecto para garantizar la gestión de configuración, velando porque la herramienta cumpla con lo establecido en las prácticas específicas para el área de gestión de configuración de CMMi.
3. Capacitar al equipo para el uso de la herramienta de gestión de configuración.



*Responsabilidades del rol Especialista de Calidad:*

1. Garantizar semanalmente que todos los miembros del equipo entreguen el registro de tiempo.
2. Chequear que estén desglosadas en los registros de tiempo las actividades que corresponden a cada rol.
3. Proponer listas de chequeo para la revisión de los principales artefactos de cada rol.
4. Ejecutar la revisión, registrar y dar seguimiento a los defectos.
5. Realizar análisis de la distribución del tiempo de manera individual y colectiva, y de las principales dificultades y lecciones aprendidas tomando en cuenta el seguimiento dado al registro de tiempo de todos los miembros del equipo a lo largo del desarrollo del proyecto.
6. Responsable de la calidad del entregable final y del producto desarrollado.

La asignatura, además, abordó el contenido de disciplina personal para el trabajo en equipo y elementos de gestión de tiempo. En este sentido se orientó a los estudiantes, desde el inicio del semestre, identificar las principales tareas que desempeñaban, planificar el tiempo para cada una y registrar el tiempo real que les consumía su ejecución.

La evaluación de cada miembro del equipo se realizó en función de tres criterios:

1. *Desempeño del rol:* Depende de la calidad de los artefactos presentados y defendidos (demostrar habilidades de comunicación) en los cortes parciales y en el entregable final. Además depende de la autoevaluación y de la evaluación recibida por todos los miembros del equipo acerca de cómo fue el desempeño del rol asignado.
2. *Aporte al equipo:* Con independencia de tener que realizar las tareas propias del rol todos los miembros del equipo deben velar por el cumplimiento de todas las tareas del proyecto. Es por eso que cada miembro se autoevalúa y recibe una evaluación del resto de los miembros acerca de su aporte al equipo. El responsable de que esta tarea se haga al finalizar el proyecto es el Jefe de proyecto.
3. *Objetividad en la evaluación:* Objetividad al autoevaluarse y evaluar al resto de los miembros del equipo.

Al finalizar la ejecución de la asignatura se obtuvieron los resultados siguientes:

1. El equipo asignó los roles en función de las competencias reales de cada miembro.
2. Realizaron actividades de capacitación para preparar a los miembros en las temáticas necesarias para el desarrollo del proyecto en particular.
3. El rigor, la exigencia y el control sistemático de las tareas asignadas propiciaron elevar el nivel de responsabilidad y laboriosidad porque del correcto desempeño de los roles dependía la nota del equipo.
4. Al analizar los resultados de la evaluación se pudo constatar que los estudiantes fueron honestos y exigentes en los criterios y las evaluaciones otorgadas, penalizando a los miembros del equipo que no apoyaban al colectivo y que no tuvieron un adecuado cumplimiento de sus tareas en el tiempo. Asimismo, se pudo observar que los estudiantes fueron justos y honestos en sus autoevaluaciones.

En la ejecución de esta experiencia docente intervinieron un total de 96 estudiantes y se constituyeron 17 equipos de proyectos. Al concluir la asignatura se aplicó una encuesta a los estudiantes, con el objetivo de que evaluaran la contribución de la asignatura a su formación profesional. El total de alumnos encuestados fueron 40 que representan el 41,6% del total, existieron representantes de todos los equipos de proyectos. En la encuesta se solicitó que valoraran en Alta, Media, Baja o Nada la contribución a su formación profesional de cada uno de los siete aspectos siguientes:

1. Valore la contribución del proyecto al desarrollo de la capacidad de trabajar en equipo.
2. Valore la contribución del aprendizaje de las buenas prácticas de PSP en la formación de la disciplina personal de cada miembro del equipo.
3. Valore la contribución de la gestión de tiempo practicada en su formación.
4. Valore la contribución de la evaluación de pares (todos a todos) en su formación.
5. Valore la contribución del desarrollo de un plan de proyecto en su formación.
6. Valore la contribución de la gestión de configuración practicada en su formación.

7. Valore la contribución de la asignatura en su formación profesional.

Los resultados obtenidos con respecto a los siete aspectos mencionados anteriormente se reflejan en la Figura 2.

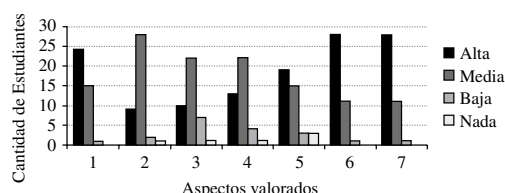


Figura 2. Valoración de los estudiantes sobre la contribución de la asignatura en su formación.

Además, en la encuesta se solicitó que valoraran en Alta, Media, Baja o Nada la contribución al desarrollo de cada uno de los roles siguientes:

- Analista
- Jefe de Proyecto
- Arquitecto
- Planificador
- Gestor de Configuración
- Especialista de Calidad
- Programador

Los resultados obtenidos con respecto a cómo valoran los estudiantes la contribución de la asignatura al desarrollo de cada uno de los roles, se pueden observar en la Figura 3.

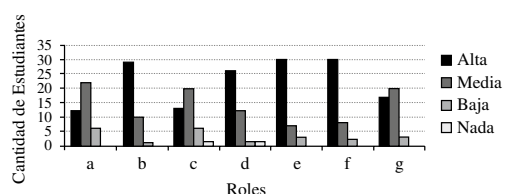


Figura 3. Valoración de los estudiantes sobre la contribución a la formación de roles.

Los resultados obtenidos en la aplicación de la encuesta demuestran que de manera general existe un alto grado de satisfacción de los estudiantes con respecto a la contribución de la asignatura en su formación profesional. Particularmente resalta la valoración que tienen respecto a la contribución de la gestión de configuración practicada y al desarrollo

de un plan de proyecto en su formación. Asimismo fue valorado entre alto y medio por casi la totalidad de los encuestados el aspecto de la contribución del proyecto al desarrollo de la capacidad de trabajar en equipo.

Sin embargo, cerca del 50% de los estudiantes valoran como medio el aspecto de la contribución del aprendizaje de las buenas prácticas de PSP en la formación de la disciplina personal de cada miembro del equipo. Este resultado demuestra que aún hay que trabajar en la formación de hábitos de disciplina personal desde las asignaturas precedentes, pues en un solo semestre académico no es posible lograrlo.

Con respecto a la contribución de la gestión de tiempo practicada en su formación, los estudiantes la valoraron entre medio y bajo. Este resultado se debe en gran medida a que en esta asignatura fue donde por primera vez se exigió que de manera real registraran sus tiempos y realizaran análisis para obtener conclusiones y lecciones aprendidas. En las asignaturas precedentes se está impartiendo el contenido de gestión de tiempo de manera formal y no se está insistiendo en la recolección de métricas de tiempo a lo largo de la carrera.

En cuanto a la contribución de la asignatura en el desarrollo de cada uno de los roles, las opiniones de los estudiantes muestran que los roles en los que más contribuyó la asignatura fueron el de Jefe de Proyecto, Planificador, Gestor de Configuración y Especialista de Calidad. El resto de los roles solo se ejercitaron y se pudo constatar que los estudiantes tenían los conocimientos precedentes necesarios para desempeñarlos.

Partiendo del análisis realizado hasta el momento, acerca de cómo ha marchado la ejecución de la estrategia de formación incluida en el plan D, se está en condiciones de enumerar los logros alcanzados hasta el momento con respecto a la formación que se alcanzaba con el Plan C.

### Logros de la estrategia de formación del Plan D con respecto al Plan C

- Consolidación del trabajo en equipo:** En el Plan C los proyectos de cursos se realizaban en equipos pequeños de no más de dos estudiantes. En la propuesta de formación del Plan D se hace

énfasis en la creación de equipos no menores de cinco estudiantes y en la asignación de proyectos reales.

2. *Formación enfocada en la contribución individual al trabajo del equipo:* En el plan C la formación estaba enfocada solo en la contribución individual de los estudiantes. En la nueva estrategia se hace énfasis en la contribución de cada estudiante al desempeño del equipo al que pertenece.
3. *Desempeño de roles:* En el plan anterior la asignación de roles y la evaluación del desempeño de cada rol no estaban formalizadas, todos los miembros del equipo se identificaban con un único rol, el programador. En la nueva estrategia se definen claramente las responsabilidades de cada rol y en función de las competencias necesarias para su desempeño se asignan a los integrantes de un equipo.
4. *Visión total del proceso de desarrollo de software:* Con la nueva estrategia el egresado alcanza una visión total del proceso de desarrollo de software. En el plan anterior no existía ninguna asignatura que permitiera la realización de un proyecto completo en el que intervinieran múltiples roles.
5. *Formación de hábitos y habilidades en temas de Calidad de Software:* Los conocimientos de calidad de software en el plan C solo eran abordados de forma teórica. Con la nueva estrategia se han formado estudiantes en el desempeño del rol Especialista de Calidad, Planificador y Gestor de Configuración. Además, se hace énfasis en las asignaturas precedentes en el trabajo con estándares, en la gestión de tiempo, en la realización de revisiones al análisis, al diseño y al código mediante la utilización de listas de chequeo.
6. *Formación de hábitos y habilidades en temas de Planificación:* En la nueva estrategia se hace énfasis en la formación del rol Planificador. Se ha logrado que los estudiantes pongan en práctica estos conocimientos y además realicen tareas de organización, control y seguimiento al avance del proyecto.

#### **Principales problemas detectados en la ejecución de la estrategia:**

A partir de la ejecución de la asignatura Ingeniería de Software III se pudo evaluar cómo ha marchado

la formación de roles y hábitos de trabajo por la calidad en las asignaturas precedentes que tributan a este empeño.

Si bien los estudiantes formados con la estrategia han adquirido hábitos de trabajo disciplinado, aún persisten insatisfacciones. Por ejemplo, no se logra que registren las métricas y construyan paulatinamente sus bases de datos de desempeño personal, útiles para futuros análisis. La razón fundamental que alegan estudiantes y profesores es que resulta necesario automatizar la recolección de estas métricas.

Por otra parte, influye en este comportamiento la no exigencia de los profesores de las diferentes asignaturas que forman roles. Así, los profesores pertenecientes a estos colectivos de asignaturas no están evaluando el desempeño de los roles formalmente. Al no evaluar los roles los estudiantes pueden no sentir que están siendo formados para su desempeño de forma disciplinada. Se detectó, también, que los profesores no están monitoreando la confección de la base de datos histórica personal. En este sentido, se limitan sólo a aportar los conocimientos teóricos al respecto y carecen de una orientación precisa acerca de cómo utilizar alguna herramienta informática que los asista en el proceso de enseñanza y aprendizaje.

En este sentido, al llegar al cuarto año, en la asignatura Ingeniería de Software III resultaría muy valioso que los estudiantes contaran con su base de datos histórica para poder realizar estimaciones más precisas y así formar el rol Planificador de una manera más real. De igual manera, resultaría sumamente valioso que los estudiantes lleguen con los roles certificados, que sientan que los han estado formando por roles, para poder realizar las asignaciones de manera más objetiva.

Lograr que los estudiantes posean bases de datos históricas con sus registros de desempeño personal de tiempo, tamaño y defectos constituye un resultado valioso a obtener con la puesta en marcha de la estrategia. Además, resulta de extrema importancia que se puedan realizar análisis postmortem con estos datos. Así, se considera importante definir acciones que propicien el logro de este objetivo.

## ACCIONES PARA EL PERFECCIONAMIENTO DE LA ESTRATEGIA DE FORMACIÓN

Como parte de la estrategia de formación de roles y hábitos de trabajo disciplinados resulta muy valioso el uso de una herramienta informática. En este sentido, existen numerosas herramientas que permiten la planificación, estimación y seguimiento de proyectos [21]. En la Tabla 2 se muestran las características de un conjunto de estas herramientas. Como se observa, todas ellas permiten la planificación y registro del tiempo de desarrollo de las tareas de un proyecto, pero pocas contabilizan las métricas de defectos y solo el “Personal Process Dashboard” (PPD) incluye la métrica de tamaño. Por otra parte, todas dan soporte al trabajo en equipo, pero solo el PPD contempla el Proceso de Software Personal.

Tabla 2. Comparación de herramientas de gestión de proyectos.

| Herramienta                           | Registro de tiempo | Registro de Defectos | Registro de tamaño | Licencia    |
|---------------------------------------|--------------------|----------------------|--------------------|-------------|
| Microsoft Office Project Professional | Sí                 | No                   | No                 | Propietario |
| Teamwork                              | Sí                 | Sí                   | No                 | Propietario |
| DotProject                            | Sí                 | Sí                   | No                 | Libre       |
| RHPlan                                | Sí                 | No                   | No                 | Propietario |
| Visual Studio 2005 Team System        | Sí                 | No                   | No                 | Propietario |
| Atlassian Jira                        | Sí                 | Sí                   | No                 | Propietario |
| ToDoList                              | Sí                 | No                   | No                 | Libre       |
| GanttProject                          | Sí                 | No                   | No                 | Libre       |
| Planner                               | Sí                 | No                   | No                 | Libre       |
| Process Dashboard                     | Sí                 | Sí                   | Sí                 | Libre       |

Después de analizar detalladamente cada una de las herramientas referidas, se considera que la que más se ajusta a los fines del proceso de enseñanza y aprendizaje es el PPD. El PPD [22], es un software libre de apoyo a PSP con módulos que incluyen los guiones y formularios que aparecen en el libro de Watts Humphrey “A discipline for Software Engineering” [15]. No obstante, la herramienta tiene la desventaja de que no permite realizar análisis del comportamiento de las métricas de un grupo de clases o de determinados estudiantes.

Por esta razón, se está considerando como trabajo futuro el desarrollo de una herramienta propia que tome como base las funcionalidades del PPD y las extienda para asistir la labor docente del profesor y permita arrastrar las métricas personales de los estudiantes a lo largo de toda la carrera.

Por otra parte, ninguna de las herramientas analizadas permite la recolección de métricas del desempeño de roles. Solo el PPD pudiera adaptarse a estos fines, teniendo en cuenta que fue diseñado basándose en la utilización de plantillas que definen el proceso a seguir en el desarrollo de un proyecto. Así, es posible utilizar el PPD para apoyar la asignatura Ingeniería Software III. A continuación se analiza la forma en la que esta herramienta puede utilizarse.

### Adaptación del PPD para el trabajo por roles:

Las funcionalidades soportadas por el PPD están básicamente encaminadas a automatizar lo planteado en PSP para adquirir disciplina personal, lo que implica que exista registro de tiempo e interrupciones, registro de defectos, trabajo con estándares de tipo de defectos, estimaciones de tiempo y tamaño y conformación de planes de trabajo. Sin embargo, la arquitectura de este software fue concebida con posibilidades de modelar cualquier proceso que pudiera ser segmentado en fases o tareas.

De esta forma, es posible definir un proceso que no tenga la estructura de fases planteadas en PSP y donde resulte de interés la recolección de métricas similares. Esta es la característica principal que permite que la herramienta se adapte a procesos personalizados. Un aspecto esencial es que es una herramienta desarrollada bajo la modalidad de código abierto, contándose con su código fuente para modificar y añadir nuevas funcionalidades. A continuación se explican las características fundamentales del PPD que permiten dar seguimiento a procesos personalizados, considerándose necesario comenzar por cómo concibe el PPD un proyecto.

### Concepción de un proyecto en el PPD:

El PPD ofrece dos modos de trabajo, *Proyecto* (“Project”) y *No Proyecto* (“Non Project”). Estos modos están dados porque la herramienta puede utilizarse para controlar el tiempo dedicado a actividades que están fuera de un proyecto, para estos casos se selecciona el modo *No Proyecto*. En el modo *Proyecto* se definen, a través de un

editor de jerarquía y aplicando una plantilla con las tareas correspondientes, los proyectos que serán desarrollados. Definir un proyecto en el PPD equivale a aplicar una plantilla de proceso definida previamente; en las plantillas se establecen las tareas o fases por las que transitará el proyecto en cuestión y para las que se registrarán métricas.

#### **Plantillas de procesos personalizadas:**

El PPD incluye plantillas para los distintos niveles de PSP. Además, ofrece plantillas como “Timer” y “Generic” que buscan dar flexibilidad a la herramienta, en el caso de que no se desee seguir exactamente el proceso de ninguno de los niveles de PSP, pero resulte valioso registrar algunas de las métricas de tiempo, tamaño y/o defectos. No obstante, estas plantillas ofrecidas por el PPD es posible crear nuevas plantillas que describan un proceso personalizado y crear y asociar guiones y formularios que serán dinámicamente integrados a la herramienta. El marco de trabajo proporcionado por el PPD para crear un proceso personalizado es muy flexible. Solo se precisa definir las fases que componen el proceso y crear un fichero XML con una estructura específica.

En la estrategia de formación se identifican doce roles a desarrollar durante la carrera [18]. Teniendo en cuenta que el PPD ofrece la posibilidad de crear plantillas de procesos personalizados, con sus guiones y formularios, se consideró crear plantillas de procesos asociadas a los roles propuestos, donde las fases del proceso fueran las tareas propuestas a desempeñar por cada rol. De manera que con el PPD sea posible controlar la disciplina personal en el desempeño un rol.

Para definir las plantillas de proceso de cada uno de los roles se identifican una serie de elementos esenciales a tener en cuenta:

- Resulta importante seleccionar cuidadosamente las fases del proceso porque una vez que se haya definido un proceso personalizado y se comience a usar en el PPD no pueden existir cambios, porque se invalidarían los datos históricos.
- Cuando se defina el proceso, se prevé recolectar datos de tiempo y de defectos, y debe hacerse para fases que tengan cierto nivel de granularidad. Si se divide el proceso en muchas fases registrar

el tiempo para todas ellas puede tornarse engorroso y poco práctico.

- Las fases que se definen en el proceso personalizado definen los "recipientes" para la colección de métricas. Deben escogerse estas fases previsoramente, para permitir que el proceso pueda crecer y mejorar. Resulta vital definir fases que constituyan invariantes en el proceso, de lo contrario el proceso que se define se puede hacer obsoleto rápidamente.
- Cuando se describe el proceso personalizado se pueden tener fases con nombres largos. Al usar esta fase, la ventana principal del PPD se alargaría mucho en la pantalla. Una abreviación resulta más apropiada. Cuando se creen los guiones HTML y formularios se puede referir a la fase por su nombre completo.
- Incluir en cualquier proceso que se defina las fases Planificar y Análisis de Resultados o Postmortem. La fase de planificación con el propósito de estimar el tamaño y los tiempos. La fase de Post mortem para realizar el análisis de los datos registrados y registrar las propuestas de mejora al proceso.

Un estudiante desempeñando el rol que le ha sido asignado el profesor, al encontrarse en su estación de trabajo debe tener la plantilla de proyecto correspondiente a ese rol. Una vez creado el Proyecto del estudiante aplicando la plantilla correspondiente, el PPD permite un conjunto de funcionalidades como: registrar el tiempo que dedican al desempeño de las tareas de las que son responsables al asumir un rol determinado, registrar los defectos que cometen, planificar las tareas y permite un aspecto muy importante que es actualizar la base de datos históricos con las métricas recogidas del desempeño del rol por parte del estudiante en el proyecto.

Con el objetivo de propiciar el uso de la herramienta por parte del claustro de profesores y teniendo en cuenta que en el mecanismo de adaptación se necesitan conocer determinadas especificaciones de la herramienta se elaboró un manual [23] que se encuentra a disposición de estudiantes y profesores.

#### **CONCLUSIONES**

Al evaluar la ejecución de la estrategia de formación, luego de cuatro cursos académicos se arriban a las conclusiones siguientes:

- Resulta necesario establecer un mecanismo de certificación de roles a lo largo de la carrera. En este sentido, los estudiantes podrán percibir de manera consciente que están siendo formados para el desempeño de roles y así establecer tempranamente los roles de su preferencia.
- Para ejercitar el desempeño de los estudiantes en el trabajo en equipo, resulta muy valioso insertarlos en proyectos reales y constituir equipos de no menos de cinco estudiantes.
- Realizar la evaluación del trabajo del equipo y de los desempeños individuales mediante la autoevaluación y la evaluación por pares permite la formación de valores tales como: honestidad, sinceridad, responsabilidad, laboriosidad y sentido de justicia.
- Resulta necesaria una herramienta informática propia que apoye el proceso de enseñanza y aprendizaje, en este sentido se encamina el trabajo futuro.
- Con la aplicación de la estrategia en estos cuatro cursos académicos se puede afirmar que los estudiantes: se han apropiado de buenas prácticas en el trabajo por la calidad de software, han logrado consolidar el trabajo en equipo desempeñando diferentes roles y logran tener una visión más general del proceso de desarrollo de software.

#### REFERENCIAS

- [1] "New Standish Group". The Standish Group International. Date of visit: June, 2010. URL: [http://www1.standishgroup.com/newsroom/chaos\\_2009.php](http://www1.standishgroup.com/newsroom/chaos_2009.php)
- [2] R.S. Pressman. "Software Engineering: A Practitioner's Approach". McGraw-Hill Science. 2004.
- [3] D. Wastell. "The Human Dimension of the Software Process, in Software Process: Principles, Methodology and Technology". A.K.J. Derniame and D. Wastell Eds. Springer-Verlag, pp. 165-199. 1999.
- [4] S.T. Acuña. "Capabilities-Oriented Integral Software Process Model". Tesis para optar por el título de doctor. Universidad Politécnica de Madrid. Madrid, España. 2002.
- [5] S.T. Acuña, N. Juristo and A.M. Moreno. "Emphasizing Human Capabilities in Software Development". IEEE Software. Vol. 23, Issue 2, pp. 94-101. 2006.
- [6] W.S. Humphrey. "The Personal Software Process". Technical Report CMU/SEI-2000-TR-022, November, 2000.
- [7] W.S. Humphrey. "The Personal Process in Software Engineering". III International Conference on the Software Process IEEE, Reston, Virginia, pp. 69-77, October, 10-11. 1994.
- [8] W. Hayes. "The Personal Software Process: An Empirical Study of the Impact of PSP on Individual Engineers". Technical Report CMU/SEI-97-TR-001 ESC-TR-97-001. 1997.
- [9] M. André Ampuero. "El proceso de software en equipo: de la disciplina personal a la disciplina organizacional". XI Convención Informática 2005. La Habana, Cuba. Febrero 2005.
- [10] J.M. Hogan and R. Thomas. "Developing the Software Engineering Team". Proceedings of the 7th Australasian Conference on Computing Education. 203-210, Newcastle. 2005.
- [11] M. André Ampuero y J. Martínez. "Transformaciones en el plan de Estudio de Ingeniería Informática en busca de una mayor calidad en el proceso de formación". XII Convención Internacional Informática. La Habana, Cuba. Febrero 2007.
- [12] M. André Ampuero y Y. López. "Creando un profesional con disciplina en el proceso de desarrollo de software". Revista de Ingeniería Industrial. Vol. XXVII, No. 1, pp. 27-30. 2006.
- [13] W.S. Humphrey. "Introducción al Proceso Software Personal (PSP)". SEI Series in Software Engineering. Addison-Wesley. 2000.
- [14] W.S. Humphrey. "Introduction to the Team Software Process". SEI Series in Software Engineering. Addison-Wesley. 2000.
- [15] W.S. Humphrey. "A discipline for software engineering". SEI Series in Software Engineering. Addison-Wesley. 2000.
- [16] M. Pomeroy-Huff, R. Cannon, E. Czerwinski, T. Kelly, J. Mullaney and J. Welch. "The Personal Software Process (PSP) Body of Knowledge". Version 2.0. Date of visit: June, 2010. URL: <http://www.sei.cmu.edu/tsp/tools/bok/>. Special Report CMU/SEI-2009-SR-018. 2009.

- [17] W.S. Humphrey. "Pathways to Process Maturity: The Personal Software Process and Team Software Process". Date of visit: June, 2010. URL: <http://stsc.hill.af.mil/CrossTalk>. 1998.
- [18] M. André Ampuero, J. Martínez, A. Hernández, Y. López and I. Wilford. "Estrategia para la formación de roles en la carrera de Ingeniería Informática". Documento Comisión Nacional de Carrera. 2007.
- [19] PMI, Guía de Fundamentos de la Dirección de Proyectos. (Guía del PMBOK®). Tercera Edición, Project Management Institute. 2004.
- [20] M. Beth Chrissis, M. Konrad and S. Shrum. "CMMi. Guía para la integración de procesos y la mejora de productos". Pearson Educación. Segunda edición. 2009. ISBN: 9788478290963.
- [21] "Open Source Project management Software Directory: planning, estimating, tracking". Date of visit: August, 2011. URL: <http://www.opensourceprojectmanagement.org/>
- [22] Sitio Oficial de la herramienta PPD Fecha de Consulta: Junio 2010. URL: <http://processdash.sourceforge.net/> 2010.
- [23] Y. López. "Manual para el uso de la herramienta: Personal Process Dashboard". Reporte anual de investigaciones del Centro de Estudios de Ingeniería y Sistemas CEIS. 2009.