



## DISEÑO DE PROGRAMAS DE ACTUALIZACIÓN PROFESIONAL INTERACTIVA A DISTANCIA

Montilva, Jonás <sup>(p)</sup> (Universidad de Los Andes, Grupo GIDyC, Venezuela, [jonas@ula.ve](mailto:jonas@ula.ve))  
Sandia, Beatriz (Universidad de Los Andes, CEIDIS, Venezuela, [bsandia@ula.ve](mailto:bsandia@ula.ve))  
Martínez, Ana Y. (Universidad de Los Andes, Grupo GIDyC, Venezuela, [amartinez94@gmail.com](mailto:amartinez94@gmail.com))  
Rivero, Dulce M. (Universidad de Los Andes, Grupo GIDyC, Venezuela, [milagro@ula.ve](mailto:milagro@ula.ve))  
Barrios, Judith (Universidad de Los Andes, Grupo GIDyC, Venezuela, [judith@ula.ve](mailto:judith@ula.ve))  
Besembel, Isabel (Universidad de Los Andes, Grupo GIDyC, Venezuela, [ibc@ula.ve](mailto:ibc@ula.ve))

### Resumen

El diseño de programas de actualización profesional es un proceso complejo, en el cual se conjugan aspectos instruccionales, estructurales, organizativos y tecnológicos muy particulares. La estructura del programa, los modos de interacción facilitador-participante, la forma y estructura de los contenidos y los tipos de actividades instruccionales son algunos de los aspectos que deben ser diseñados cuidadosamente, a fin de lograr una alta efectividad del programa. En este artículo, se discuten e ilustran estos aspectos a través del estudio de un caso real: El Programa de Actualización Profesional en Ingeniería de Software de la Universidad de Los Andes. Este programa está compuesto de un conjunto de cursos que cubren diferentes áreas de conocimiento de la Ingeniería de Software. Estos cursos son desarrollados y ofrecidos bajo la modalidad de enseñanza-aprendizaje a distancia. Cada uno de ellos es implementado como un sitio web de tipo instruccional, para lo cual se emplea un sistema de gestión de cursos en línea de código abierto denominado MOODLE. La experiencia descrita, en este artículo, puede ser de mucho interés para aquellas instituciones universitarias que se inician en este tipo de programas de extensión.

Palabras claves: Programas de Actualización Profesional, Sitios Instruccionales, Métodos de Diseño Curricular

### Abstract

The process of designing an online professional program is a complex process, in which it is combined particular and diverse aspects, such as: instructional, structural, organizational, and technological ones. There are four main aspects that have to be carefully designed for achieving a high quality and effective program. These four aspects are the structure of the program, the kind of facilitator-participant interaction, the structure and form of the contents, and the types of the instructional activities. In this paper, we present and discuss the aforementioned aspects by following a real case study: The Professional Program in Software Engineering at The University of Los Andes-Venezuela. This program is composed by a set of courses that covers different knowledge areas of the Software Engineering. These courses are developed and offered under the distance-learning approach. Each one of the courses is implemented as an instructional web site, for which it is used an open source course management system called MOODLE. Our presented experience may be of great interest for those academic institutions that are now initiating extension programs of this type.



## **1. INTRODUCCIÓN**

En los últimos años, la oferta y la demanda de programas de actualización profesional interactiva a distancia han crecido significativamente. Un número importante de universidades latinoamericanas ofrece regularmente programas de esta naturaleza. Un programa de este tipo es un cuerpo coherente, estructurado y muy actualizado de cursos cortos, apoyados en la tecnología web, que le permite a un profesional complementar su formación, ganar nuevos conocimientos, adquirir habilidades y mejorar sus competencias en un área de conocimientos determinada.

La creación, operación y administración de un programa de actualización profesional son procesos complejos, que requieren ser gestionados apropiadamente para garantizar la efectividad, eficacia y eficiencia del programa. La creación, en particular, es un proceso en el cual se conjugan aspectos de naturaleza muy diferente, tales como: aspectos instruccionales, tecnológicos, metodológicos, organizativos y gerenciales. Esta complejidad se ve incrementada, aún más, por el carácter interdisciplinario del proceso; pues, se requiere la participación activa de distintos actores, incluyendo: diseñadores instruccionales, diseñadores web, administradores de sistemas, facilitadores, expertos en contenidos, etc. Para manejar la complejidad del proceso de creación se hace necesario ordenar, sistematizar y gestionar el proceso. El orden y la sistematización del proceso se logran mediante la utilización de métodos apropiados que indiquen que actividades hacer, como hacerlas, quien debe hacerlas y por que deben hacerse. La gestión del proceso se alcanza mediante la aplicación de técnicas gerenciales de planificación, organización, dirección y control.

El proceso de creación de un programa de este tipo tiene mucho en común con el proceso de desarrollo de software. Este último se caracteriza por seguir un conjunto de fases o etapas en las que se analiza el problema, se especifican los requisitos de la solución, se diseña en detalle la solución, se implanta y, finalmente, se evalúa el producto antes de proceder a su entrega. Los métodos existentes para el desarrollo de software difieren, esencialmente, en la manera en que las fases mencionadas se ejecutan. Algunos métodos las ejecutan secuencialmente, otros de manera iterativa, otros de forma evolutiva o incremental (Thayer & Dorfman, 2003).



En este artículo, se describe un marco metodológico para la creación de programas de actualización profesional que está inspirado en el método WATCH de Montilva & Barrios (2004). WATCH es un método para el desarrollo de aplicaciones web que se fundamenta en la aplicación de procesos, modelos y mejores prácticas de la Ingeniería de Software.

El marco metodológico propuesto consta de dos modelos complementarios. El primero de ellos es un modelo conceptual que describe la estructura de un programa de actualización profesional. El segundo de ellos, denominado modelo de procesos, describe el conjunto de actividades que ha de ejecutarse para elaborar un programa de actualización e indica, también, en que orden deben hacerse estas actividades, como deben hacerse y que producen. El marco propuesto tiene las siguientes características: (1) es iterativo e incremental; (2) combina actividades instruccionales, tecnológicas y gerenciales y (3) la elaboración del programa es gestionado bajo la forma de un proyecto, mediante la aplicación de actividades gerenciales, tales como la planificación y control del proyecto, la organización del equipo de trabajo, la gestión de riesgos y la verificación/validación de los productos obtenidos.

El modelo de procesos consta de siete procesos técnicos (análisis del contexto, definición de requisitos, diseño de la estructura curricular, diseño de cursos, elaboración de los cursos, evaluación y entrega del programa) y tres procesos gerenciales (gestión del proyecto, gestión de riesgos y verificación/validación). En este artículo, se enfatiza el proceso de diseño de la estructura curricular ilustrándolo mediante la aplicación del marco metodológico a un caso real: el Programa de Formación Profesional para la Industria Nacional de Software (GIDyC, 2005, 2006a).

El resto de este artículo está organizado de la siguiente manera: la Sección 2 presenta el modelo conceptual de los programas de actualización profesional, el cual describe las características generales de este tipo de programa de estudios. La Sección 3 describe el modelo de procesos que se propone para elaborar programas de actualización profesional. La Sección 4 describe brevemente la aplicación del marco metodológico a un caso real. En esta sección, se hace énfasis en el proceso de diseño curricular y se describen los resultados de este proceso, incluyendo la estructura curricular del programa y sus aspectos instruccionales, organizativos y tecnológicos. Finalmente, en la Sección 5 se presentan las conclusiones del trabajo.



## 2. MODELO CONCEPTUAL DE LOS PROGRAMAS DE ACTUALIZACIÓN PROFESIONAL

Un programa de actualización profesional está integrado por una colección estructurada de cursos cortos vinculados con un área de conocimientos determinada. El objetivo general de este tipo de programas de estudio es complementar y/o actualizar la formación que los profesionales de una determinada disciplina tienen en relación a un área de conocimientos generalmente especializada.

Estos programas se ofrecen, por lo general, bajo dos modalidades diferentes, no necesariamente excluyentes: modalidad presencial y modalidad interactiva a distancia (Sandía & Montilva, 2002). Bajo la modalidad interactiva a distancia, el programa es elaborado, usado y administrado aplicando tecnología web. Cada uno de sus cursos es ofrecido a través de Internet, usando como medio de interacción a un sitio web de tipo instruccional. Los sitios web se crean, acceden, operan y mantienen utilizando un tipo de sistema de software denominado Sistema de Gestión de Cursos en línea (*CMS - Course Management System*) o Sistema de Gestión de Aprendizaje (*LMS - Learning Management Systems*).

El modelo conceptual ilustrado en la figura 1 muestra los componentes fundamentales de un programa de actualización profesional. Cada programa persigue un conjunto de objetivos curriculares previamente establecidos. Para alcanzar estos objetivos se diseña e implanta una estructura curricular compuesta por una colección de cursos en línea interrelacionados. Estos cursos son implementados usando una plataforma tecnológica apoyada en Internet. El hardware que integra esta plataforma está compuesto por un servidor conectado a Internet y un número cualquiera de máquinas PC que acceden al programa a través de la dirección IP del servidor. El software de la plataforma está integrado por el sistema de gestión de cursos (LMS o CMS), un conjunto de herramientas multimedia y software de oficina, los cuales son usados para la elaboración y publicación de contenidos instruccionales. El programa es operado, administrado y actualizado por un conjunto de actores, en los que se incluyen los estudiantes que cursan el programa y el personal encargado de administrar el programa y mantener actualizado los cursos que lo conforman.

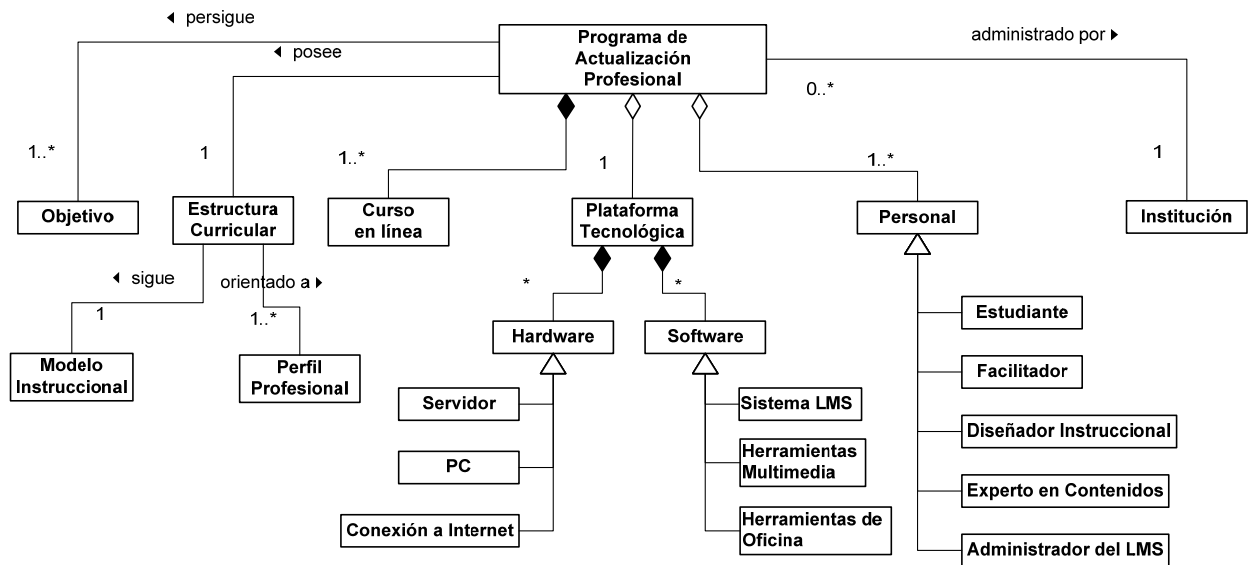


Figura 1. Modelo general de un programa de actualización profesional

### 3. MODELO DE PROCESOS PARA LA ELABORACIÓN DE PROGRAMAS DE ACTUALIZACIÓN PROFESIONAL

Durante los últimos diez años, se ha publicado un número importante de trabajos relacionados con el proceso de creación o elaboración de cursos en-línea (ver, por ejemplo, Khan, 1997 y Hall, 1999). Montilva, Sandia & Barrios (2002) discuten este proceso y describen un método de desarrollo de cursos en línea que explica en detalle las fases y pasos que se requieren para elaborar los sitios instruccionales sobre el cual se apoyan los cursos en línea. Sin embargo, el proceso de elaboración de un programa de actualización profesional va más allá del proceso de elaboración de cursos en línea. El primero enmarca el segundo. Tal como lo indica la figura 1, un curso en línea es sólo un componente más de un programa. Se requiere, por consiguiente, de métodos que vayan más allá de los cursos. En esta sección, se propone un marco metodológico para elaborar programas de actualización. Este marco complementa aquellos orientados a cursos en línea, tal como el descrito por Montilva, et al (2002).

Para elaborar un programa de actualización profesional es necesario, primero, analizar el contexto donde el programa operará, luego, establecer sus requisitos, diseñar su estructura curricular conformada por el conjunto de cursos, realizar el diseño instruccional de cada curso, implementar cada uno de ellos y, finalmente, evaluar el programa como un todo. Este conjunto de actividades es

complejo y requiere la participación de un equipo de trabajo integrado por diseñadores instruccionales, administradores web, facilitadores y expertos en contenidos, entre otros.

Para garantizar que el programa sea elaborado a tiempo, de acuerdo al presupuesto asignado y siguiendo los estándares de calidad establecidos, se hace necesario manejar su elaboración bajo la forma de un proyecto. El proyecto permite que el proceso de elaboración del programa sea eficaz y eficiente.

De acuerdo a las consideraciones anteriores, se organizan las actividades, que se requieren para elaborar un programa de actualización profesional, en dos grupos: procesos técnicos y procesos gerenciales (ver figura 2).

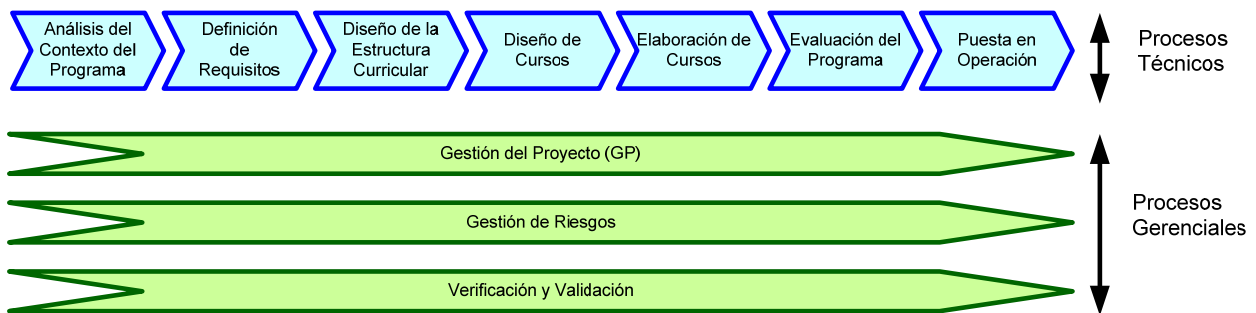


Figura 2. Grupos de procesos para la elaboración de programas de actualización profesional

A partir de la clasificación de procesos de la figura 2, se diseñó el modelo de procesos ilustrado en la figura 3. Este modelo describe las actividades que debe seguir un equipo de trabajo para desarrollar ordenada y sistemáticamente un programa de actualización profesional y está basado en la metáfora del reloj, descrita en Montilva y Barrios (2004).

Como se puede observar en la figura 3, en el centro del modelo está ubicado el grupo de procesos gerenciales, el cual dirige y controla las actividades del grupo de procesos técnicos.

El proceso técnico está integrado por las actividades técnicas e instruccionales y están representadas en el entorno del modelo simulando el dial de un reloj. Estas actividades se ejecutan en el sentido de las agujas del reloj y en el orden indicado por la numeración. Aunque el proceso se sigue secuencialmente,

el orden puede ser alterado por el Líder del Proyecto, bien para corregir algún aspecto del programa o para incorporar un nuevo requisito.

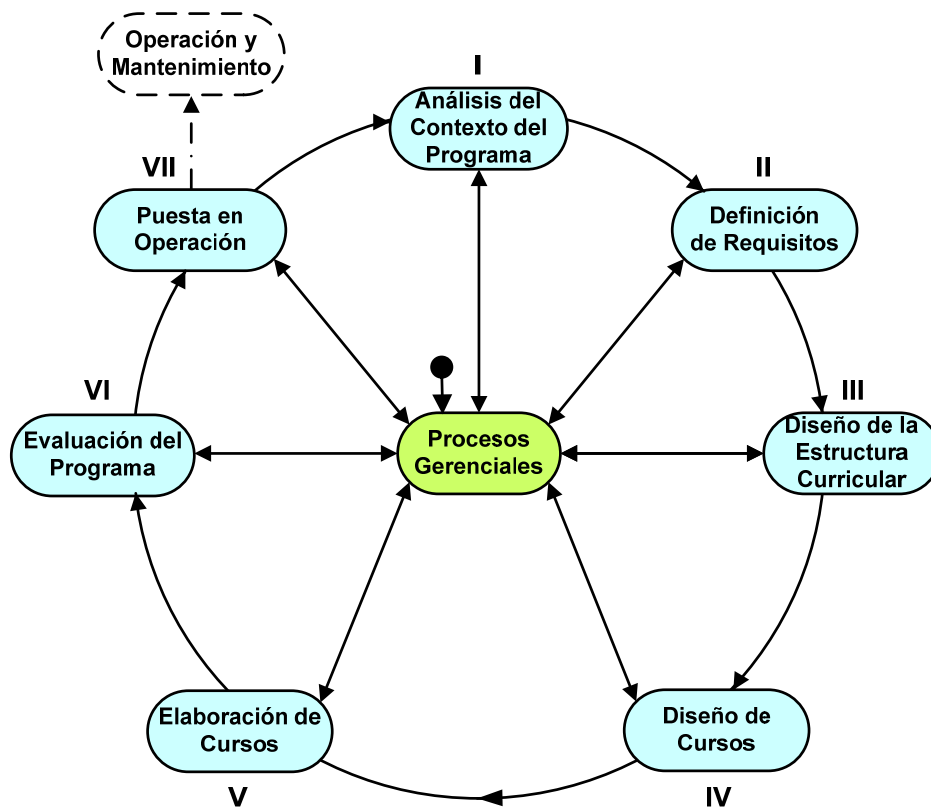


Figura 3. Modelo de procesos para la elaboración de programas de actualización profesional

Las siguientes subsecciones describen resumidamente los dos grupos de procesos que integran el modelo.

### Los procesos gerenciales de elaboración de programas

La gestión de proyectos es el medio empleado por excelencia en ingeniería para manejar la complejidad de un proceso tecnológico. Basado en los estándares de gestión de proyectos de Ingeniería de Software (Thayer & Dorfman, 2003, Vol. 2) y en la guía de dirección de proyectos PMBOK del Project Management Institute (2004), se diseñó el grupo de procesos gerenciales que se resume en la tabla 1.

Tabla 1. Procesos gerenciales para la elaboración de programas de actualización

<i>Procesos</i>	<i>Actividades</i>	<i>Productos</i>
Gestión del Proyecto	Planificar el proyecto	<ul style="list-style-type: none"> <li>• Plan del Proyecto</li> </ul>
	Organizar el equipo de trabajo	<ul style="list-style-type: none"> <li>• Estructura del equipo de trabajo</li> <li>• Definición de roles y responsabilidades de los actores</li> </ul>
	Administrar recursos económicos y tecnológicos	<ul style="list-style-type: none"> <li>• Informes de gestión de los recursos utilizados</li> </ul>
	Dirigir el proyecto	<ul style="list-style-type: none"> <li>• Lineamientos, directrices y políticas de gestión del proyecto</li> </ul>
	Controlar el proyecto	<ul style="list-style-type: none"> <li>• Actualizaciones del Plan del Proyecto</li> </ul>
Gestión de Riesgos	Identificar riesgos	<ul style="list-style-type: none"> <li>• Lista de riesgos</li> </ul>
	Analizar riesgos	<ul style="list-style-type: none"> <li>• Impacto y probabilidad de riesgos</li> </ul>
	Planificar respuestas	<ul style="list-style-type: none"> <li>• Planes de contingencia</li> </ul>
	Monitorear y controlar riesgos	<ul style="list-style-type: none"> <li>• Medidas de mitigación</li> </ul>
Verificación y Validación	Planificar y controlar las revisiones técnicas de los productos	<ul style="list-style-type: none"> <li>• Plan de revisión de productos</li> </ul>
	Planificar y controlar las pruebas del programa de actualización	<ul style="list-style-type: none"> <li>• Plan de pruebas del programa de actualización</li> </ul>

El grupo de procesos gerenciales incluye tres procesos: gestión del proyecto, gestión de riesgos y verificación/validación de productos. Estos procesos son ejecutados por aquella persona que tenga la responsabilidad principal del proyecto. A este actor se le identifica con el rol de Líder del Proyecto.

La gestión del proyecto enmarca, a su vez, un conjunto de actividades gerenciales que incluyen la planificación del proyecto, la organización del equipo de trabajo, la dirección de este equipo, la administración de los recursos económicos, humanos y tecnológicos asignados y el control del proyecto.

Las actividades de Planificación y Control del proyecto están relacionadas con la definición, seguimiento y ajuste de las actividades técnicas del proyecto, los tiempos de ejecución de estas actividades y los recursos humanos asignados a cada una de ellas.

La Gestión de Riesgos tiene que ver con la identificación, análisis y manejo de todas aquellas situaciones problemáticas que puedan afectar el normal desenvolvimiento del proyecto. El Líder del





Proyecto debe identificar estas situaciones, analizarlas, medir su impacto y definir los planes de contingencia que deben activarse si alguna de estas situaciones acontece durante la ejecución del proyecto.

La Verificación y Validación de Productos agrupa a un conjunto de actividades gerenciales que se encargan de evaluar los productos que se elaboran durante la ejecución de las actividades de los procesos técnicos y gerenciales. Cada producto intermedio o final debe ser sometido a una revisión técnica que garantice el cumplimiento de los requisitos y estándares de calidad que se hayan establecido para el programa.

### Los procesos técnicos de elaboración de programas

Los procesos técnicos están relacionados con las actividades instruccionales y tecnológicas necesarias para definir, diseñar, implementar y poner en operación el programa. Cada uno de estos procesos genera un producto intermedio o final. Estos productos son documentos que describen diferentes aspectos del programa de actualización profesional. Las actividades y los productos asociados a este tipo de procesos se resumen en la tabla 2.

Tabla 2. Procesos técnicos para la elaboración de programas de actualización

<i>Procesos</i>	<i>Actividades</i>	<i>Productos</i>
Análisis del Contexto del Programa	Analizar el entorno (institución) donde operará el programa	Documento de Contexto del Programa
	Establecer las necesidades de actualización profesional	
	Establecer las características generales del profesional que se desea actualizar	
Definición de Requisitos	Definir el alcance del programa	Documento de Requisitos del Programa
	Definir los requisitos instruccionales	
	Definir los requisitos estructurales	
	Definir los requisitos tecnológicos	
Diseño de la Estructura Curricular del Programa	Definir el perfil profesional deseado	Documento de la Estructura Curricular del Programa
	Establecer los objetivos instruccionales del programa	
	Diseñar el modelo instruccional del programa	
	Diseñar la estructura curricular del programa	
	Especificar la plataforma tecnológica del programa	
	Especificar los aspectos organizativos del programa	
Diseño de Cursos	Establecer los objetivos instruccionales	Documento de Diseño de Cursos



<i>Procesos</i>	<i>Actividades</i>	<i>Productos</i>
	de cada curso	
	Establecer las estrategias instruccionales específicas de cada curso	
	Diseñar la estructura programática de cada curso	
	Definir el plan de curso	
	Definir las actividades de aprendizaje	
	Diseñar los mecanismos de evaluación de cada curso	
	Diseñar el sitio instruccional de cada curso	
Desarrollo de Cursos	Acondicionar la infraestructura de desarrollo y operación de los cursos: hardware y ambiente físico	Cursos en línea del programa
	Seleccionar, adquirir e instalar el sistema de gestión de cursos (LMS ó CMS)	
	Desarrollar los contenidos instruccionales	
	Producir los ítems multimedia	
	Crear el sitio instruccional	
	Montar contenidos, ítems y demás elementos en el sitio instruccional	
Evaluación del Programa	Realizar las revisiones técnicas de los productos técnicos y gerenciales	Producto intermedio revisado
	Diseñar y ejecutar las pruebas de aceptación del programa de actualización	Programa de actualización evaluado
Puesta en Operación del Programa	Instalar el programa en la plataforma tecnológica de operación del programa	Programa de actualización en operación
	Probar la operación del programa en su plataforma de operación	

#### 4. ESTUDIO DE UN CASO: EL PROGRAMA DE ACTUALIZACIÓN PROFESIONAL EN INGENIERÍA DE SOFTWARE

En los últimos años, se ha generado en nuestro país un crecimiento significativo de la demanda de profesionales de la Ingeniería de Software (GIDYC, 2006b). Esta demanda es ocasionada primordialmente por dos factores:

- Las fuerzas del mercado nacional e internacional, que requieren cada vez más productos de software para apoyar a una sociedad del conocimiento cuyo motor principal es, precisamente, este tipo de producto.
- El establecimiento de políticas nacionales tendentes a impulsar el crecimiento del sector de tecnologías de información y comunicación, muy particularmente, de aquel denominado *Industria Nacional del Software (INS)*, el cual abarca a todas aquellas empresas radicadas en Venezuela cuyo objetivo es la producción de software y/o la prestación de servicios de desarrollo, operación y mantenimiento de software.



En Octubre 2005, los autores dieron inicio a un proyecto titulado “Programa de Formación Profesional para el Desarrollo de la Industria Nacional de Software (INS)”, cofinanciado por el FONACIT y la ULA (GYDYC, 2005). Este programa tiene como propósito contribuir a la actualización tecnológica del recurso humano que integra la INS, a través de un programa de formación profesional en Ingeniería de Software basado en la utilización de la tecnología WWW. El programa está integrado por un conjunto de cursos en línea de corta duración que son ofrecidos a profesionales de la INS, a través de Internet, usando la tecnología WWW. Por comodidad, en este artículo, nos referiremos a este programa de actualización profesional como el *Programa DINSoft*<sup>1</sup>

El Programa DINSoft se ha desarrollado siguiendo el método descrito en las secciones anteriores. En esta sección, se resumen los principales resultados (productos) obtenidos durante la ejecución de los tres primeros procesos técnicos: Análisis de Contexto, Definición de Requisitos y Diseño Curricular del programa. Los detalles de estos resultados, y de los demás procesos del proyecto, están disponibles en el sitio web del proyecto: [www.centauro.ing.ula.ve/dinsoft](http://www.centauro.ing.ula.ve/dinsoft)

### **Necesidades, requisitos y perfiles**

A través de una encuesta estadística realizada sobre una población estimada en más de 150 empresas nacionales que integran la INS, se diagnosticó la situación actual del recurso humano empleado por esta industria y se establecieron sus necesidades de formación profesional (GIDYC, 2006b).

Además de satisfacer las necesidades de formación profesional de la INS, se estableció un requisito instruccional muy importante: El programa debía ser consistente con las directrices y lineamientos curriculares establecidos en documentos relacionados con la Educación en Ingeniería de Software, entre los cuales se destacan los siguientes:

- Lineamientos de Educación en Ingeniería de Software del Instituto de Ingeniería de Software (Bagert et al, 1999).
- El cuerpo de conocimientos de la Ingeniería de Software SWEBOK elaborado por la Sociedad de Computación de la IEEE (IEEE Computer Society, 2004).

---

<sup>1</sup> Siglas derivadas del título del proyecto: Programa de actualización profesional para el Desarrollo de la Industria Nacional del Software.



- El Currículo de Ingeniería de Software del Grupo de Trabajo Conjunto para Currícula de Computación de la IEEE/ACM (2004).
- El programa de certificación profesional de la IEEE Computer Society (2007).
- Artículos selectos de reconocidos autores y relacionados con la Educación en Ingeniería de Software, tales como: (Shaw, 2000), (Parnas, 1999) y (Ford & Gibbs, 1996).
- El modelo de mejora de procesos CMMI desarrollado por el Instituto de Ingeniería de Software (SEI, 2002).

A partir del análisis de los resultados de la encuesta y de los requisitos establecidos, se definieron cuatro perfiles profesionales que cubren los procesos técnicos, gerenciales y de apoyo que requiere la INS para desarrollar y mantener software de alta calidad. Estos perfiles son los siguientes:

- Gestor o Líder de Proyectos de Software
- Ingeniero de Soporte
- Desarrollador de Software
- Ingeniero de Mantenimiento

Cada uno de estos perfiles tiene asociado un conjunto de roles y responsabilidades que definen las competencias y conocimientos que el profesional debe poseer. Los detalles de estos perfiles se describen en (GIDYC, 2006c).

### **Objetivos del Programa**

El objetivo general del Programa DINSOFT es contribuir a la formación del recurso humano especializado requerido por la INS, mediante una oferta de cursos a distancia que, empleando la tecnología web, coadyuven a la capacitación de los profesionales que integran este sector económico del país. El objetivo específico del programa es ofrecer a los profesionales de la INS un medio eficaz de enseñanza-aprendizaje a distancia que, sin restricciones de tiempo y lugar, les permita:

- Actualizar sus conocimientos en los nuevos métodos, conceptos, prácticas y enfoques de la Ingeniería de Software.
- Mejorar sus competencias y adquirir habilidades en el uso de técnicas y herramientas modernas empleadas en la Industria del Software para el desarrollo de software.
- Prepararse para la certificación profesional en Ingeniería de Software.



## **Aspectos del programa**

El programa DINSoft fue diseñado mediante el análisis de los objetivos y requisitos establecidos en la encuesta anteriormente señalada y la definición de los perfiles profesionales requeridos por la INS. Este diseño fue elaborado y documentado desde varias perspectivas o puntos de vista que definen cuatro aspectos del programa que son diferentes, pero complementarios: aspectos instruccionales, aspectos estructurales, aspectos organizativos y aspectos tecnológicos. Cada uno de estos aspectos se describe a continuación.

### *Aspectos instruccionales*

El diseño y desarrollo de los contenidos de un programa de actualización en línea, requieren una adecuada estructuración y una minuciosa planificación, de tal forma que facilite el proceso de aprendizaje por parte de los estudiantes, así como el control y seguimiento por parte del docente.

La estructuración del programa se ve beneficiada con la incorporación de objetos de aprendizaje que permitan garantizar contenidos adaptables, actualizables y reutilizables. Estos objetos de aprendizaje se definen como pequeños componentes informáticos que plantean un objetivo instruccional, a partir del cual se estructuran contenidos, módulos, y materiales de estudio. Estos objetos son, además, escalables, independientes entre sí y pueden ser reutilizados y reensamblados, en diferentes contextos, para producir distintos cursos o un mismo curso con distinto enfoque.

Cada unidad de aprendizaje está compuesta de un contenido específico, actividades a desarrollar y la evaluación o autoevaluación (Moreno y Bailly-Baillière, 2002). Se debe tener presente el equilibrio entre las tres variables esenciales de un proceso enseñanza aprendizaje: la cantidad y calidad del diálogo, la estructura o diseño instruccional del curso y la autonomía del estudiante (Fandos, 2003).

Estas variables determinan los elementos claves en el desarrollo del proceso de formación: el tipo de aprendizaje que se quiera generar (por ejemplo, memorizar, aplicar técnicas), el control del aprendizaje (centrado en el profesor o centrado en el alumno), la dirección del aprendizaje (área específica o interdisciplinaria), la necesidad de agrupamiento para aprender (individual, pareja, grupo), los tipos y niveles de interacción (alumno - herramientas; alumno - contenido; estudiante - estudiante; estudiante -



profesor, etc.). Cada unidad de aprendizaje requiere, por consiguiente, aplicar una o más estrategias de enseñanza para alcanzar su objetivo de aprendizaje.

Por otro lado, en el proceso técnico para la elaboración de programas de actualización, es importante tomar en consideración que el diseño de éstos debe realizarse teniendo como objetivo el desarrollo de las competencias laborales de los participantes en el área de interés.

Como sugiere Catalano, De Cols y Sladogna, (2004): “El diseño curricular basado en competencias es un documento elaborado a partir de la descripción del perfil profesional, es decir, de los desempeños esperados de una persona en un área ocupacional, para resolver los problemas propios del ejercicio de su rol profesional. Procura de este modo asegurar la pertinencia, en términos de empleo y de empleabilidad, de la oferta formativa diseñada”

Teniendo en consideración lo anteriormente planteado, las estrategias de enseñanza definidas para el programa DINSoft están enfocadas en las siete competencias (niveles de aprendizajes) que se pretenden desarrollar en el participante y que fueron establecidas a partir de los perfiles profesionales definidos en (GIDYC, 2006c), a saber:

- **Abstracción:** es la capacidad que tiene el ingeniero de software de centrar su atención en los aspectos relevantes de un problema y dejar de lado aquello que no es vital para su solución.
- **Conceptuación:** capacidad del profesional de la ingeniería de software para formar conceptos y atribuirle propiedades a los objetos de estudio.
- **Simbolización:** aptitud en el ingeniero de software para representar, a través de lenguajes artificiales, conceptos asociados al espacio de un problema o al de su solución.
- **Revisión:** competencia presente en el ingeniero de software para someter sus productos de ingeniería de software a una auto-evaluación, juicio o examen con el fin de realizar mejoras o correcciones necesarias.
- **Estructuración:** capacidad del ingeniero de software de definir, ordenar y relacionar las partes o componentes de un sistema.
- **Descomposición:** capacidad del ingeniero de software para dividir, en forma jerárquica y recursiva, las partes de un sistema.
- **Innovación:** aptitud que posee el ingeniero de software para formular y elaborar soluciones novedosas (innovadoras) a un problema dado.

### Aspectos estructurales

La figura 4 muestra la estructura curricular que fue diseñada para el programa DINSoft. Esta estructura está compuesta de cinco grupos de cursos no excluyentes. El primer grupo, denominado “Cursos Introdutorios”, es obligatorio para todos los estudiantes o participantes del programa. Estos cursos proveen las bases conceptuales, metodológicas, instrumentales y tecnológicas que el estudiante requiere poseer para avanzar a los niveles siguientes del programa. Una vez culminado los cursos introductorios, el participante elige una de cuatro líneas diferentes relacionadas con los perfiles profesionales requeridos por la INS. Así, por ejemplo, aquellos participantes que deseen mejorar sus competencias y actualizar sus conocimientos en la gestión de proyectos han de seguir la Línea de Formación para Líderes de Proyectos.



Figura 4. Estructura curricular del Programa DINSoft

Cada línea de formación consta de cinco cursos adicionales a los introductorios. Cada curso requiere, en promedio, 30 horas de dedicación del participante. La duración total del programa, para cada línea de formación, está estimada en 240 horas.

### Aspectos organizativos

La organización del personal que opera y mantiene un programa de actualización profesional es crucial para lograr su efectivo y eficiente funcionamiento. La estructura organizacional propuesta para el

Programa DINSoft es de tipo jerárquica y está integrada por un conjunto de roles compuesto de un coordinador del programa, un administrador de la plataforma, un grupo de diseñadores instruccionales, un grupo de facilitadores y un grupo de expertos en contenidos (ver figura 5). Las responsabilidades de cada uno de estos roles se resumen en la tabla 3.

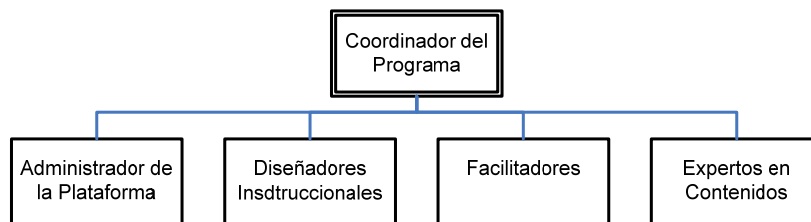


Figura 5. Estructura organizativa del Programa DINSoft

Tabla 3. Roles y principales responsabilidades del personal del Programa DINSoft

<b>Roles</b>	<b>Responsabilidades</b>
Coordinador del Programa	<ul style="list-style-type: none"> <li>• Gestionar el programa de actualización profesional</li> <li>• Planificar cada cohorte del programa</li> <li>• Supervisar el desarrollo de cada cohorte y de sus diferentes cursos</li> <li>• Evaluar el dictado de los cursos y los resultados de cada cohorte</li> <li>• Administrar los procesos de inscripción, pago y culminación de cada curso</li> <li>• Dirigir el personal a su cargo</li> <li>• Promocionar el programa</li> </ul>
Administrador de la plataforma	<ul style="list-style-type: none"> <li>• Administrar y mantener operativa la plataforma tecnológica del programa</li> <li>• Asegurar la operación y disponibilidad del sistema de gestión de cursos <i>MOODLE</i></li> <li>• Administrar el uso de los componentes de la plataforma</li> <li>• Dar apoyo técnico a los usuarios en el uso de la plataforma tecnológica</li> </ul>
Diseñador instruccional	<ul style="list-style-type: none"> <li>• Diseñar los componentes instruccionales de cada curso: objetivos, estructura programática, plan de curso, lecciones, actividades, autoevaluación, etc.</li> <li>• Diseñar, implementar y mantener el sitio instruccional de cada curso</li> <li>• Diseñar y producir los ítems multimedia que requiera cada curso</li> </ul>
Facilitador	<ul style="list-style-type: none"> <li>• Apoyar el proceso de aprendizaje en línea</li> <li>• Interactuar periódicamente con los estudiantes a través del sitio instruccional</li> <li>• Atender las consultas en línea</li> <li>• Supervisar el proceso de aprendizaje de los estudiantes inscritos en el curso</li> </ul>
Experto en contenidos	<ul style="list-style-type: none"> <li>• Elaborar los contenidos de cada una de las lecciones que integran un curso</li> <li>• Definir las actividades complementarias de cada lección</li> <li>• Elaborar las preguntas y respuestas de la autoevaluación de cada curso</li> <li>• Mantener actualizado los contenidos instruccionales de los cursos</li> </ul>





### *Aspectos tecnológicos*

Dos aspectos tecnológicos importantes de un programa de actualización son la plataforma tecnológica que lo soportará y la arquitectura de software que agrupa a los diferentes componentes del programa.

Para el desarrollo y operación del programa de actualización DINSOFT se adquirió e instaló una plataforma tecnológica especializada que consta, esencialmente, de: Una red local conectada a Internet e integrada por un servidor y un conjunto de máquinas PC; el sistema de gestión de cursos MOODLE (2007); una suite de herramientas de oficina para la edición de textos, gráficos y documentos; un conjunto de medios audiovisuales para la producción de videos y un conjunto de herramientas multimedia para la edición de imágenes, videos y audio.

La figura 6 ilustra la arquitectura de software del Programa DINSOFT. Como puede apreciarse en dicha arquitectura, el personal (usuarios) accede al programa a través de un sitio web desarrollado expresamente para este proyecto (<http://www.centauro.ing.ula.ve/dinsoft>). Este sitio se conecta con el sistema MOODLE para permitir, a los usuarios autorizados, el acceso a los cursos del programa o a las facilidades que este sistema presta para la creación y administración de cursos en línea.

Cada curso en línea puede acceder directamente a uno o más componentes de aprendizaje SCO (*Sharable Content Object*). Estos componentes son objetos de aprendizaje, tal como los define el comité LTSC (2007) y están basados en el estándar SCORM (ADL, 2004, 2007). Son creados por los diseñadores instruccionales y/o expertos en contenidos (profesores) del programa DINSOFT, usando un editor de SCO de código abierto denominado RELOAD (2006). Cada componente creado con RELOAD es catalogado y almacenado en una Base de Componentes de Aprendizaje (BCA) usando una aplicación de base de datos creada expresamente para documentar, almacenar, consultar y actualizar la información de cada SCO almacenado en la BCA (ver GIDYC, 2006d).

La BCA consta de tres elementos de software estrechamente relacionados:

- Una **base de datos** relacional que almacena los componentes de aprendizaje SCO.
- Un **sistema de gestión de SCO** que consiste en una aplicación de base de datos para la documentación, catalogación, consulta y actualización de la información relacionada con cada uno de los componentes de aprendizaje almacenados en la base de datos.

- Un *editor RELOAD* para la creación de componentes de aprendizaje SCO.

Cada componente SCO almacenado en la BCA puede ser invocado desde cualquier sistema de gestión de cursos en línea (LMS) compatible con el estándar SCORM.

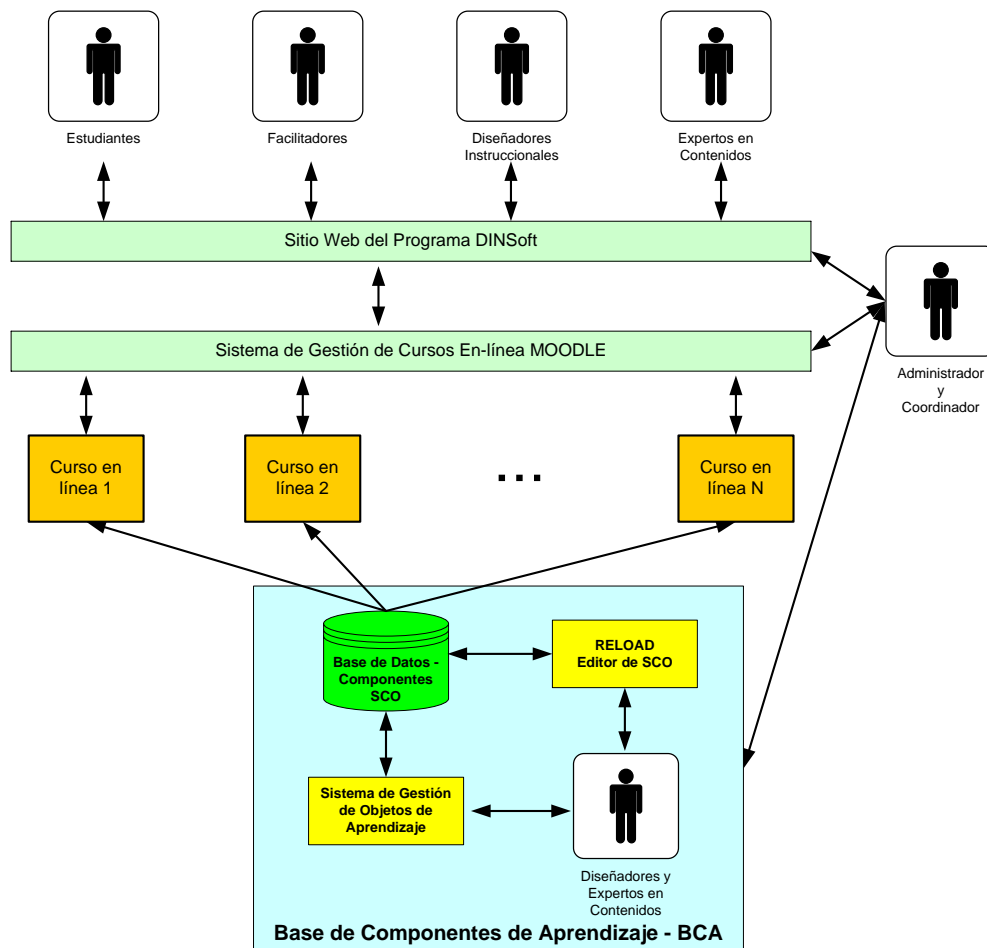


Figura 5. Arquitectura de software del programa DINSOFT

## 5. CONCLUSIONES

Este artículo describió un marco metodológico para la creación de programas de actualización profesional inspirado en el método WATCH de Montilva & Barrios (2004). El marco contempla los distintos aspectos instruccionales, estructurales, organizativos y tecnológicos que deben ser considerados al momento de diseñar un programa de este tipo. Estos aspectos fueron ejemplificados a



través del estudio de un caso real: El Programa de Actualización Profesional en Ingeniería de Software de la Universidad de Los Andes.

Una propiedad importante del marco propuesto es su adaptabilidad. Puede, por consiguiente, ser adaptado a los requisitos particulares del desarrollo de cualquier programa de actualización profesional. Otras características que lo distinguen son: (1) el carácter iterativo e incremental de su modelo de procesos, lo cual facilita la revisión y corrección de los productos del proceso; (2) la combinación de actividades instruccionales, tecnológicas y gerenciales, la cual le imprime al marco un carácter integral; pues, se consideran de manera integrada los aspectos más relevantes de la elaboración de un programa de actualización profesional; y (3) la gestión del desarrollo del programa bajo la modalidad de proyecto, lo cual contribuye a planificar y controlar los costos, esfuerzos y tiempo empleados en el desarrollo del programa.

El marco fue aplicado en el diseño del programa de actualización DINSOFT. Este diseño estuvo basado en el análisis de los objetivos y requisitos establecidos en la encuesta estadística realizada sobre una población estimada en más de 150 empresas nacionales que integran la Industria Nacional del Software (GIDYC, 2006b). Los resultados de esta encuesta permitieron definir los perfiles profesionales requeridos por la INS, y consecuentemente las competencias a desarrollar en los participantes (GIDYC, 2006c).

Algunos de los aspectos instruccionales y organizacionales más resaltantes del diseño del programa DINSOFT son los siguientes:

- La estructuración y planificación de los contenidos del programa fundamentados en la incorporación de objetos de aprendizaje permiten garantizar que los mismos sean adaptables, actualizables y reutilizables. Es decir, los contenidos pueden ser reutilizados y reensamblados en diferentes contextos para producir distintos cursos o un mismo curso con distinto enfoque.
- La estructura curricular del Programa DINSOFT presenta cuatro líneas de formación diferentes relacionadas con los perfiles profesionales requeridos por la INS: Líderes de Proyectos, Desarrolladores de Software, Ingenieros de Soporte e Ingenieros de Mantenimiento.



- La estructura organizacional propuesta para el Programa DINSOFT es de tipo jerárquica y está integrada por un conjunto bien definido de roles: coordinador del programa, administrador de la plataforma, diseñadores instruccionales, facilitadores y expertos en contenidos. Esta estructura garantiza el funcionamiento eficaz del programa, su continua actualización y, por ende, su permanencia en el tiempo.

## Agradecimientos

Los autores agradecen el cofinanciamiento otorgado por el FONACIT, bajo el Proyecto Número 2001003171, y el CDCHT de la Universidad de Los Andes, bajo el programa ADG, para la realización del proyecto bajo el cual se enmarca este artículo.

## Referencias

- Advanced Distributed Learning – ADL (2004). Sharable Content Object Reference Model (SCORM) Overview. 2<sup>nd</sup>. Edition. Extraído el 02 de Febrero de 2006 de [www.adlnet.org](http://www.adlnet.org).
- Advanced Distributed Learning - ADL (2007). SCORM 2004 Third Edition. Extraído el 05 Febrero, 2007 de <http://www.adlnet.org>
- Bagert, D.J. et al. (1999). *Guidelines for Software Engineering Education. Version 1*. Technical report CMU/SEI-99-TR-032. Software Engineering Institute. Carnegie Mellon University. October.
- Catalano, A., Avolio, S. & Sladogna, M. (2004). *Diseño curricular basado en normas de competencia laboral – Conceptos y orientaciones metodológicas*. Banco Interamericano de Desarrollo.
- Fandos, M. (2003). *Formación Basada en las Tecnologías de la Información y Comunicación: Análisis didáctico del proceso de enseñanza-aprendizaje*. Tesis Doctoral. Universidad Rovira i Virgili. Tarragona.
- Ford, G. & Gibbs, N.E. (1996). A Mature Profession of Software Engineering. Technical Report CMU/SEI-96-TR-004. Software Engineering Institute, Pittsburgh, Pennsylvania, USA.
- GIDYC - Grupo de Investigación en Ingeniería de Datos y Conocimiento (2005). *Programa de Formación Profesional para el Desarrollo de la Industria Nacional del Software*. Propuesta técnica del Proyecto FONACIT 2001003171. Mérida: Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación.
- GIDYC - Grupo de Investigación en Ingeniería de Datos y Conocimiento (2006a). *Proyecto FONACIT 2001003171 – Primer Informe de Avance*. Mérida: Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación.
- GIDYC - Grupo de Investigación en Ingeniería de Datos y Conocimiento – GIDyC (2006b). *La Industria Nacional del Software y sus Recursos Humanos – Estado Actual*. Informe Técnico. Proyecto FONACIT 2001003171. Mérida: Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación.



GIDYC - Grupo de Investigación en Ingeniería de Datos y Conocimiento (2006c). *Escenarios y Perfiles para Actualización Profesional en Ingeniería de Software*. Informe técnico. Proyecto FONACIT 2001003171. Mérida: Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación.

GIDYC - Grupo de Investigación en Ingeniería de Datos y Conocimiento (2006d). *Diseño y creación de la Base de Componentes de Aprendizaje del Programa de Formación Profesional en Ingeniería de Software*. Informe técnico. Proyecto FONACIT 2001003171. Mérida: Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación.

Hall, R. H. (1999) Instructional Web Site Design Principles: A Literature Review and Synthesis. *The Virtual University Journal*, Vol. 2, pp. 1-12.

IEEE/ACM (2004). *Curriculum Guidelines for Undergraduates Degree Programs in Software Engineering*. The Joint Task Force on Computing Curricula IEEE/ACM

IEEE Computer Society (2007). The IEEE Computer Society CSDP Examination. Extraído de <http://www.computer.org/csdp>

IEEE Computer Society (2004). Guide to the Software Engineering Body of Knowledge - SWEBOK. 2004 Version. Professional Practices Committee.

Khan, B. H. (1997). *Web-Based Instruction*. Educational Technology Publications, Englewood Cliffs.

Learning Technology Standards Committee – LTSC (2007). The Learning Object Metadata Standard. Extraído el 05 Febrero 2007 de <http://ieeeltsc.org>

MOODLE (2007). A Free, Open Source Course Management System for Online Courses. Extraído de <http://www.moddle.org>

Montilva, J., Sandia, B. & Barrios, J. (2002). Developing Instructional Web Sites - A Software Engineering Approach. *Education and Information Technologies*. Kluwer Publishing. Vol. 7, No. 3.

Montilva, J. & Barrios, J. (2004). The WATCH Method for Developing Web Applications. En Montilva, J., Besembel, I., Pérez, M. & Losavio, F. (Eds.). *Sistemas de Información e Ingeniería de Software: Temas Selectos*. (pp. 328-343). Mérida, Venezuela: Centro de Estudios en Informática.

Moreno, F. y Bailly-Baillièrre, M. (2002) *Diseño instructivo de la formación online. Aproximación metodológica a la elaboración de contenidos*. Barcelona: Ariel Educación.

Parnas, D. (1999). Software Engineering Programs are not Computer Science Programs. IEEE Software, November/December. pp. 19-30.

Project Management Institute – PMI (2004). *Guía de los fundamentos de la Dirección de Proyectos (Guía del PMBOK)*. Pennsylvania, EE.UU: Project Management Institute, Inc.

RELOAD (2006). Reusable eLearning Object Authoring and Delivery. Disponible en <http://www.reload.ac.uk/>

Sandia, B. & Montilva, J. (2002). Los estudios interactivos a distancia en la Universidad de Los Andes. *Revista Acción Pedagógica*. Vol. 11, No. 1.

SEI - Software Engineering Institute (2002). CMMI - Capability Maturity Model Integration, Version 1.1. CMMI for Software Engineering. Technical Report # CMU/SEI-2001-TR-029. Carnegie Mellon University, Software Engineering Institute.

Shaw, M (2000). *Software Engineering Education: A Roadmap*. In Finkelstein, A. (Ed.), *The Future of Software Engineering*". ACM Press.



Thayer, R.H. & Dorfman, M., Eds. (2003). *Software Engineering Volumes 1 and 2*. IEEE Computer Society. Wiley- Interscience.

## **CURRICULUM VITAE**

Jonás Montilva: Profesor Titular del Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes. Ingeniero de Sistemas, Universidad de los Andes; Magíster en Computación, Case Western Reserve University, Ohio, EEUU; Doctor en Computación, Universidad de Leeds, Inglaterra.

Beatriz Sandia: Profesora Asociada del Departamento de Ciencias Aplicadas y Humanísticas de la Escuela Básica de la Facultad de Ingeniería de la Universidad de Los Andes. Ingeniero Civil, Universidad de Los Andes; Magíster en Educación en el área de Tecnología Educativa, George Washington University, Washington DC, USA; Candidata a Doctorado del Programa Inter-universitario en Tecnología Educativa, ofrecido por la Universidad de las Islas Baleares, España.

Ana Y. Martinez: Ingeniera de Sistemas, Universidad de Los Andes. Estudiante de la Maestría en Educación Abierta y a Distancia, UNA, Venezuela. Egresada del Programa de Capacitación Docente, UNA, Venezuela. Profesora de Instituto Universitario de la Frontera (IUFRONT), Mérida. Ingeniera contratada, Grupo GiDYC, ULA, Mérida.

Dulce M. Rivero: Profesora Asociada del Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes. Ingeniera de Sistemas, Universidad de Los Andes, Magister en Computación, Universidad de Los Andes, Doctora en Computación, Universidad Carlos I, España.

Judith Barrios: Profesora Titular del Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes. Ingeniera de Sistemas, Universidad de Los Andes, Magister en Sistemas de Información, Instituto Tecnológico de Monterrey, México. DEA y Doctorado en Informática, Universidad de Paris I, Francia.

Isabel Besembel: Profesora Titular del Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes. Ingeniera de Sistemas, Universidad de Los Andes. DEA en Informática, Universidad de Paris VI, Francia. Ph.D. en Computación, Universidad de Leeds, Inglaterra.