

# Ingeniería de software y universidad

## Software engineering and university

Juan Camilo Giraldo Mejía  
MsC. en Ingeniería de Sistemas  
Docente ocasional  
Tecnológico de Antioquia  
jgiraldo1@tdea.edu.co

Fabio Alberto Vargas Agudelo  
MsC. en Ingeniería de Sistemas  
Docente de planta  
Tecnológico de Antioquia  
fvargas@tdea.edu.co

Darío Enrique Soto Durán  
MsC. en Ingeniería de Sistemas  
Docente de planta  
Tecnológico de Antioquia  
dsoto@tdea.edu.co

*Recibido: 3 de septiembre 2012  
Aprobado: 13 de octubre 2012*

### Resumen

Este artículo aborda una reflexión académica e investigativa con el objetivo de evidenciar la relevancia del profesional en Ingeniería de software, contrastando diferentes dimensiones que permiten un enfoque ingenieril pertinente para nuestro contexto. Esto surge debido a que los sistemas de software se utilizan cada vez más en actividades críticas y con ella, aumentan los costos y niveles de complejidad para administrarlos, construirlos y mantenerlos. También estos deben prever una larga vida útil, capaz de adaptarse con el transcurso del tiempo a nuevas condiciones. Por estas razones cobra gran importancia el ingeniero en software, como profesional dentro de un ciclo de vida de las soluciones informáticas, que genere en la academia la responsabilidad de formular una propuesta curricular consistente con las necesidades de la industria nacional e internacional.

**Palabras clave:** Currículo, ingeniería, software.

## Abstract

This paper addresses a scholarly and research-oriented reflection aiming to make evident the relevance of software engineering professionals, by contrasting various dimensions allowing an engineering approach suitable to our context. This reflection arises from a view that software systems are increasingly used in critical activities, thus increasing costs and levels of complexity to manage, build and maintain them. Also, they should foresee a long service life, and be able to adapt over time to new conditions. In this line, software engineers have become more and more important as professionals accounting for IT solutions' life cycle. Thus, the academy is responsible to formulate curriculum proposals that are coherent with the needs of national and international industry.

**Keywords:** Curriculum, engineering, software.

## 1. Introducción

La ingeniería en Software se define, según Humphrey (1989, p. 82), como el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. Según Shaw (1990), es la disciplina para producir software de calidad desarrollado sobre las agendas y costos previstos y satisfacer requisitos, y según Boehm (1976), es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, la operación y el mantenimiento de software, es decir, la aplicación de la ingeniería al software.

En este artículo, en primer lugar, se muestra una evolución del software, teniendo en cuenta su capacidad productiva y los beneficios funcionales frente a las necesidades del sector. Luego se indican elementos más académicos, concretamente la relación del software y los aspectos curriculares de las universidades, y en ella el total de módulos de ingeniería de software que existen por cada plan de estudios de algunas universidades. Posteriormente se da una perspectiva ingenieril, donde se resalta qué es la ingeniería de software y la importancia de inculcarla en los estudiantes, se analiza históricamente la disciplina de la ingeniería del software y, desde una perspectiva ingenieril, se justifica por qué esta disciplina debe ser tratada como una ingeniería distinta de cualquier otra, haciendo referencia a su cuerpo de conocimiento. Luego se presentan las tendencias curriculares propuestas por la

ACM (2004). También se evidencian las necesidades de la industria del software a través de cifras del mercado, mediante un rastreo a nivel nacional e internacional, que permiten presentar la propuesta curricular del programa denominado Ingeniería en Software por ciclos propedéuticos, diseñada por la Facultad de Ingeniería del Tecnológico de Antioquia y aprobada por el Ministerio de Educación Nacional.

## 2. Marco teórico y trabajos previos

En los primeros años del software, las actividades de elaboración de programas las realizaba una sola persona utilizando lenguajes de bajo nivel y ajustándose a un computador en especial, y generaban programas difíciles de entender aun para su creador después de algún tiempo de producirse. Esto implicaba tener que repetir el mismo proceso para desarrollar el mismo programa para otras máquinas. Por consiguiente, la confiabilidad, la facilidad de mantenimiento y el cumplimiento no se garantizaban y la productividad era muy baja. Posteriormente, con la aparición de técnicas estructuradas y con base en las experiencias de los programadores, se mejoró la productividad del software, sin embargo este seguía teniendo fallas, como por ejemplo: documentación inadecuada, dificultad para su correcto funcionamiento y, por tanto, insatisfacción del cliente. Conforme se incrementaba la tecnolo-

gía de los computadores, también crecía la demanda de los productos de software, pero más lentamente, tanto que hacia la década de los noventa se decía que las posibilidades de software estaban obsoletas respecto a las del hardware en un mínimo de dos generaciones de procesadores, y que la distancia continuaba en aumento (De Marco y Lister, 1999). Por ejemplo, el software en tiempo real (software encargado de analizar datos del mundo en forma real, tales como análisis de datos, control automatizado, monitoreo de datos), el de gestión (en esta categoría se incluye el software comercial a nivel empresarial, de nóminas e inventarios), de ingeniería y científico (software que posee un amplio

manejo numérico, usado en biología, astronomía) y el CAD (empotrado que se encuentra residente en memoria) (Giraldo y Cortés, 2006). El software “permite abstraer de sí misma la máquina, y convertirse en una macro tarea de papel, diagramas y pseudolenguaje, entre otros, [...] cuando se quiere interactuar rápidamente con un computador” (Morales y Peláez, 2009).

La tabla 1 muestra la relación de 10 universidades y el total de módulos referentes a la ingeniería de software que soportan el plan de estudios de cada una de esas instituciones de educación superior.

**Tabla1.** Módulos de Software y planes de estudio

Número de modelos referentes a la Ingeniería de Software por planes de estudio	
Universidad	Respuesta
1	Cuatro
2	Tres
3	Tres
4	Más de cuatro
5	Uno
6	Dos
7	Tres
8	Cuatro
9	Más de cuatro
10	Cuatro

La Figura 1 muestra el total de universidades que tienen y aplican uno, dos, tres, cuatro, más de cuatro o ningún módulo de apoyo a los planes de estudio. Representación gráfica de la respuesta de universidades a la pregunta: ¿cuántos módulos referentes a la Ingeniería de Software soportan el plan de estudios?

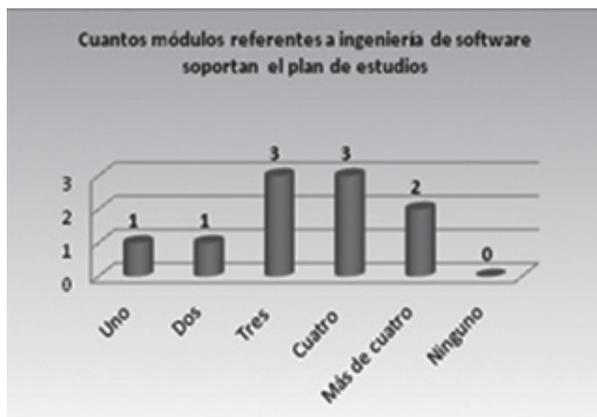


Figura 1. Total de módulos por universidad

### 3. Perspectiva ingenieril

Según el DRAE (2005), la ingeniería se define como el “estudio y aplicación, por especialistas, de las diversas ramas de la tecnología”, y el ingeniero(a) como “la persona que aplica los conocimientos de una o varias ramas de la ciencia para resolver cierto tipo de necesidad de la gente, mediante el diseño, construcción u operación de algún tipo de artefacto”. Es importante inculcar en los estudiantes y futuros profesionales que el objetivo de su profesión es diseñar, construir y hacer que funcionen pro-

ductos de software que sirven para resolver problemas de las personas, donde el foco de atención está en esto último, que es la razón de ser de cualquier ingeniería: solucionar problemas (Norman, 1990). La ingeniería existe porque las personas usan sistemas que se tornan cada vez más complejos, por tal motivo es necesario tener ingenieros capacitados para diseñarlos, construirlos y operarlos. El software se ha convertido en el mayor nivel de complejidad que el ser humano ha enfrentado a lo largo de su historia, por lo tanto merece tratarse como ingeniería (De Marco, 2009).

### 4. Particularidad de la ingeniería en software

Una fuente clara y clave para determinar el alcance de un ingeniero en software es *Guide to the Software Engineering Body of Knowledge –SWEBOK–* (IEEE-CS, 2004), donde se definen los conocimientos propios de la disciplina y las competencias que debe tener todo ingeniero en software para desempeñarse en el mercado de manera competente. SWEBOK establece los conocimientos propios de la ingeniería en software y los clasifica en 10 áreas (ver Tabla 2).

Tabla 2. Áreas de clasificación de la ingeniería de software

Número de área	Descripción de área
1	Requerimientos de software
2	Diseño de software
3	Construcción de software
4	Pruebas de software
5	Mantenimiento de software
6	Gestión de la configuración de software
7	Gestión de la ingeniería de software
8	Proceso de ingeniería de software
9	Herramientas y métodos en ingeniería de software
10	Calidad de software

Las cinco primeras son áreas tradicionales típicas del ciclo de vida del software, y las otras cinco (5) áreas permiten determinar el perfil profesional desde una perspectiva ingenieril, para diferenciar al ingeniero en software de otras disciplinas de la computación.

## 5. Orígenes de la denominación académica

### 5.1. Propuestas curriculares a nivel internacional

De acuerdo a la *Computing Curricula* (ACM, 2004), la computación se puede entender como cualquier actividad cuyo objetivo requiera, se beneficie o cree computadores. Gracias a su rápida evolución, la computación, lejos de ser una única disciplina, es hoy una familia de disciplinas. De acuerdo con este marco, se definen cinco disciplinas o perfiles: Ingeniería en computación, Ciencias computacionales, Sistemas de información, Ingeniería en software y Tecnología de información. De ellas la de mayor relevancia en la industria es la ingeniería en software (ACM, 2005): la “Ingeniería de Software es la disciplina del desarrollo y mantenimiento de sistemas de software que se comportan de manera confiable y eficiente, son factibles de desarrollar y mantener, y satisfacen todos los requerimientos que los clientes hayan definido para ellos”.

## 5.2. Necesidades del sector

### 5.2.1 Internacional

Según cifras tomadas de la agencia de trabajo de Estados Unidos, en el año 2004 se contabilizaron en este país 760.840 puestos de trabajo con el perfil de ingeniero de software, lo que representa la tercera parte del número total de ingenieros en el campo TIC (Bureau of Labor Statistics, 2004) del total de casi 6,5 millones de empleados.

### 5.2.2 Nacional

Según el documento publicado por Pro Argentina en el año 2005, en Colombia el sector de tecnología de la información es de alto crecimiento y gran dinamismo, y reporta actualmente una tasa cercana al 8%, lo que refleja su potencialidad en materia de generación de oportunidades. Sin embargo, aún no se logra desarrollar una industria de TI y software con una participación importante en las exportaciones. En 2001 estas fueron de US\$22 millones, con destino principalmente a los países limítrofes, Ecuador y Venezuela.

## 6. Contribución de la experiencia a los programas de ingeniería

Se puede afirmar que los programas de ingeniería de sistemas o ingeniería informática que se imparten en la actualidad recopilan un conjunto de conocimientos que abarcan muchas de las disciplinas planteadas en *Computing Curricula* (ACM, 2004), lo que en muchos casos genera poca profundización en algunas de ellas, como por ejemplo la ingeniería de software. En Colombia, los programas de ingeniería en el área de computación deben comenzar a mapear las realidades de la industria mundial, ya adoptadas en la industria del sector a nivel regional, cuyas empresas tienen como principal eje de acción los servicios tecnológicos en la construcción y el mantenimiento de productos del software, con incidencia a nivel nacional e internacional.

Partiendo de esta necesidad del mercado, el Tecnológico de Antioquia define esta especificidad de la informática con las siguientes características reflejadas en su currículo: Los métodos educativos y los programas de estudio deben reflejar de mejor manera el ritmo acelerado de desarrollo de las tecnologías de información y la globalización de esta actividad. La industria del software debe asumir los principios y buenas prácticas de la ingeniería propias del software para que este sector sea visible a nivel mundial. La academia y el sector empresarial son un binomio necesario para propiciar cambios significativos en nuestra economía, así como para

establecer estrategias pedagógicas en las que se puedan vivenciar las competencias transversales como trabajo en equipo, liderazgo y análisis definidos en los currículos académicos (ejemplo: La estrategia del proyecto integrador), para que los estudiantes lleven a la práctica los métodos y procesos definidos en la ingeniería.

## 7. Cómo la experiencia puede ser de utilidad para otras escuelas de ingeniería

Las experiencias que se dan en algunas facultades que tienen programa de ingeniería informática, de sistemas o de ingeniería en software, sirven como ejemplo para otras escuelas que carecen de estrategias pedagógicas, modelos de prácticas y estándares para la aplicación de los elementos adquiridos en el proceso de enseñanza-aprendizaje, pero es necesario considerar buenas estrategias de comunicación interinstitucionales. Los procesos educativos han evolucionado con el tiempo y con la utilización de diversas técnicas y estrategias que permiten generar un proceso de dinamización, siempre en busca de la interacción constante entre los actores y el empoderamiento de las relaciones que implica el proceso de enseñanza-aprendizaje (Manrique, 2009).

Actualmente el Tecnológico de Antioquia propone, concreta y aplica estrategias desde la investigación formativa y científica, que faciliten el proceso de enseñanza-aprendizaje de la ingeniería de software, y que este se vea reflejado en la industria del software aportando calidad y competitividad. Concretamente, desde la investigación formativa en el Tecnológico de Antioquia, se ha venido desarrollando un proyecto denominado “Integrador”, el cual permite a los estudiantes aplicar los conocimientos adquiridos desde el primer nivel de formación hasta el último. Este proyecto es coordinado en cada nivel por un docente conocedor del proceso, y es verificado por medio de tres formatos o fichas que los estudiantes deben diligenciar. La calidad y el alcance del proyecto permitirán que, en algunos casos, estos proyectos sean reconocidos como trabajos de grado.

## 8. Conclusiones

La industria del software se constituye en un sector estratégico de alto interés en el contexto socioeconómico del país, especialmente en la región, debido a la naturaleza de nuestro tejido industrial y empresarial.

La ingeniería en software requiere un fuerte componente de formación orientada a los procesos de dirección, organización, planificación y gestión, asociados a los proyectos de software, de manera que se garantice la calidad del producto y la satisfacción del cliente.

A nivel académico la ingeniería del software es tratada como un área dentro de los programas de ingeniería de sistemas o ingeniería informática, lo que ha llevado en muchos casos a una formación deficiente en esta área, que se refleja cuando los profesionales tienen que enfrentar proyectos informáticos de gran envergadura que impliquen la ingeniería de software como base primordial.

## Referencias

- ACM (2004). Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.
- ACM (2005). Computing Curricula 2005: The Overview Report.
- Boehm, B. (1976) Software Engineering. *IEEE Trans. Computers*, Vol. C-25, No. 12, pp. 1226-1241.
- Bureau of Labor Statistics (2004). Employment and Wages, Annual Averages.
- De Marco, T. (2009). Software engineering: An idea whose time has come and gone? *Revista IEEE Software*. Vol. 3, No. 1, pp. 95-101.
- De Marco, T. & Lister, T. (1999). *Peopeware: Productive Projects and Teams*. Dorset House.

Giraldo, Juan y Cortés, Carlos (2006). Los ciclos de vida de desarrollo, una necesidad latente. *Revista de Educación en Ingeniería*, No. 2, pp. 62-70. Facultad de Ingeniería, Universidad de Manizales, Colombia.

Humphrey, W. (1989). *The Software Engineering Process: Definition and Scope*. In: Representing and Enacting the Software Process: Proc. 4th Int'l Software Process Workshop. ACM Press.

IEEE Software (1990). In which she proposed a historical model of the professionalization of engineering based primarily on the development of chemical engineering. *IEEE Software*, Vol. 7, No. 6, pp.15-24.

IEEE Computer Society (2004). *Guide to the Software Engineering Body of Knowledge*. 2004 version. SWEBOK.

Manrique, B. (2009). La tutoría y el estudio de casos en la enseñanza de tópicos especiales en Soft-

ware, en el programa de Ingeniería de Sistemas. *Revista de Educación en Ingeniería*. No. 7, pp. 129-136. Universidad de Medellín.

Morales, Á. y Peláez, L. (2009). La importancia del análisis en la construcción de un sistema para la divulgación de información a través del teléfono celular: una propuesta desde la ingeniería de software. *Revista de Educación en Ingeniería*. No. 7, pp.146-154. Disponible en [www.educacioneningeneria.org/index.php/edi/article/download/67/57](http://www.educacioneningeneria.org/index.php/edi/article/download/67/57)

Norman, D. (1990). *The Design of Everyday Things*. University of California.

Pro Argentina. (2005). Disponible en: <http://www.proargentina.gov.ar> [Consultado, mayo de 2012]

Shaw, M. (1990). Prospects for an Engineering Discipline of Software. *IEEE Software*, Vol. 7, No. 6, pp. 15-24.