PUBLICACIONES
VICERRECTORADO ACADÉMICO

INGENIERÍA DEL SOFTWARE

Un enfoque basado en procesos

Jonás Montilva C. y Judith Barrios A.

UNIVERSIDAD DE LOS ANDES
VENEZUELA

INGENIERÍA DEL SOFTWARE

Un enfoque basado en procesos

Jonás Montilva C. y Judith Barrios A.

Este libro enfatiza dos aspectos esenciales del aprendizaje de la ingeniería del software. Por un lado, el libro identifica, describe y relaciona todos aquellos conceptos que son necesarios para comprender los fundamentos de esta ingeniería. Para ello, se presenta una ontología de dominio que define los conceptos que comúnmente usamos los ingenieros de software para desarrollar y mantener de este tipo de producto tecnológico. Por otro lado, el libro enfatiza y le da la debida importancia a los procesos que distinguen a la ingeniería del software como ingeniería, por esta razón se acude al modelado de procesos de negocios como medio de representación del desarrollo, operación y mantenimiento de software en contextos empresariales.

Describir los fundamentos de la ingeniería del software en función de conceptos y procesos le otorga al libro un carácter práctico indispensable para aplicar los métodos aquí descritos y asociar la teoría con la práctica. De allí, que los contenidos de este libro orientan tanto a estudiantes, en sus procesos de aprendizaje, como a profesionales de la computación e informática, en los aspectos metodológicos que facilitan el manejo de la complejidad inherente al desarrollo, operación y mantenimiento de productos de software.

Reconocemos este libro como un excelente material bibliográfico para actividades de enseñanza, investigación y desarrollo en el área de la Ingeniería del Software; en consecuencia, el uso de este libro lo recomendamos ampliamente a las instituciones de educación superior, las instituciones públicas y privadas, y cualquier otra organización que así lo requiera. Este libro se caracteriza por su enfoque teórico práctico que va de lo esencial a lo profundo, de una manera natural y comprensible, el cual es producto de la amplia trayectoria académica y experiencia profesional práctica de los autores que se han dedicado a esta área por más de tres décadas de fructífera labor; una obra maestra en su área, que tendrá un inestimable valor para sus lectores.

Sociedad Venezolana de Computación
https://www.svc.net.ve/

Algo que me encanta de este libro que comienzan a leer es que el tejido es impecable, que los conceptos se usan y se reutilizan con elegancia cada vez que se requiere y que siempre los autores emplean el apoyo visual del modelado para ratificar lo que quieren expresar.

Carlos Mario Zapata J., PhD
Presidente del Comité Ejecutivo del Capítulo Latinoamericano de la Iniciativa SEMAT (Teoría y Método de la Ingeniería del Software)

ISBN: 978-980-11-2029-2

Fundamentos de la ingeniería del SOFTWARE

Un enfoque basado en procesos

Jonás A. Montilva Calderón
Judith Barrios Albornoz

Departamento de Computación
Facultad de Ingeniería
Universidad de Los Andes

PUBLICACIONES
VICERRECTORADO ACADÉMICO

Universidad de Los Andes
Mérida Venezuela

FUNDAMENTOS DE LA
INGENIERÍA DEL SOFTWARE
Un enfoque basado en procesos
Primera edición, 2021

Hecho el depósito de ley
Depósito legal ME2021000035
ISBN 978-980-11-2029-2

ISBN: 978-980-11-2029-2
9 789801 120292

Diagramación
Jonás A. Montilva C.
Jessenia Torrealba
Corrección
Elsa Mora Gallardo
Diseño de carátula
Jessenia Torrealba

Universidad de Los Andes
Av. 3 Independencia,
Edificio Central del Rectorado,
Mérida, Venezuela 5101
publicacionesva@ula.ve
publicacionesva@gmail.com
http://www2.ula.ve/
publicacionesacademico

Editado en la República Bolivariana de Venezuela.

CONTENIDOS

Prólogo

El nuevo milenio ya se afianzó y comienza a mostrar los vestigios de lo que será la tecnología en el futuro. Parece increíble que solo hasta ahora se comience a hablar de la ingeniería de software como disciplina profesional y que solo hasta ahora comiencen a surgir en muchos países las carreras de ingeniería de software. Por ello, es importante el surgimiento de visiones frescas de lo que representa esta profesión y lo que representará a futuro, cuando pueda tener la madurez de las ingenierías milenarias. En efecto, a este último integrante del linaje de las ingenierías le tocó «madurar» a hipervelocidad, lo que hace que carezca de algunos de los fundamentos de sus hermanos mayores.

El libro que están a punto de leer es una muestra de esas visiones frescas sobre lo que representa esta profesión. Como piezas de un lego, los autores se adentran en los fundamentos de la ingeniería de software con una discusión clara en los capítulos 2 y 3, que culmina en el capítulo 4 con una elegante solución de modelado para las «10 P» del desarrollo de software, como construyendo los cimientos de esta edificación compleja que es el desarrollo de software con base en la notación típica del diagrama de clases de UML. A partir de allí empiezan a utilizar esos cimientos para explicar de forma precisa los diferentes tópicos de esta profesión, utilizando también el modelado con BPMN, la notación de modelos de procesos de negocios. Se discuten allí tópicos diversos como el proceso de desarrollo de software, la ingeniería de requisitos, el diseño e implementación del sistema y la validación y verificación del software, para culminar en el capítulo 10 con la gestión de proyectos de software y en el 11 con el mantenimiento de software como elementos transversales y definitivos del ciclo de vida del software. Finalmente, todo este tejido de los capítulos precedentes culmina con el método Blue Watch en el capítulo 12 para dar un contexto práctico a lo que se preparó durante todo el libro, mostrando nuevamente la aplicación de los conceptos generales de esta disciplina materializados en una guía en la que nuevamente recurren al modelado con UML y BPMN para describir cada componente del método.

Es difícil compendiar estas casi seis décadas de la ingeniería de software en pocas páginas. Es usual ver textos casi eternos donde se detallan los temas de

esta disciplina ingenieril en interminables descripciones con poca articulación. Algo que me encanta de este libro que están leyendo es que el tejido es impecable, que los conceptos se usan y se reutilizan con elegancia cada vez que se requiere y que siempre los autores emplean el apoyo visual del modelado para ratificar lo que quieren expresar. Adicionalmente, pocas visiones de la ingeniería de software pueden articular de manera consistente paradigmas tan diversos como el «agilismo», SWEBOK, SEMAT o los estándares de IEEE o ACM. Independientemente del rol que desempeñen en la ingeniería de software, se pueden acercar a esta obra para encontrar en ella lo que buscan. En esta visión fresca se podrán apoyar para encontrar los fundamentos de esta, nuestra más joven ingeniería, y descubrir en sus páginas una manera práctica de comprender aquello que la rodea y su forma de materialización. Se cumple así la promesa de los autores desde el título de esta obra, basada en los fundamentos de la ingeniería de software, cuidadosamente escrutados desde el lente de los procesos y su modelado.

Carlos Mario Zapata J., PhD.
Presidente del Comité Ejecutivo del Capítulo Latinoamericano
de la Iniciativa SEMAT (Teoría y Método de la Ingeniería de Software).

1 Introducción

Escribir un libro sobre una de las disciplinas más extensas y conocidas de la computación representa un reto de grandes proporciones, no solo por la existencia de un amplio conjunto de libros clásicos que por décadas han sido usados en muchas universidades para apoyar la enseñanza y el aprendizaje de la ingeniería del *software*, sino por la dinámica propia de una joven disciplina cuyo producto −el *software*− se caracteriza por:

- Un alto grado de complejidad estructural, funcional y dinámica que dificulta su especificación, diseño y programación.
- La aplicación de procesos y prácticas de desarrollo y mantenimiento que varían frecuentemente en función de la evolución de las tecnologías digitales, particularmente del enfoque y orientación de las metodologías de desarrollo de *software* y de las plataformas tecnológicas en las que este tipo de producto se elabora.
- La naturaleza cambiante de los requisitos que demandan los usuarios.
- La enorme variedad de dominios de aplicación para los cuales se desarrolla, día a día, un número cada vez más creciente de sistemas de *software*.
- Su ubicuidad, pues no existe un área de conocimientos, ni disciplina alguna, en la que el *software* no esté presente como elemento de apoyo fundamental para la ejecución de sus actividades.

La ingeniería del software es una de las disciplinas que, junto a la ingeniería de computadores, los sistemas de información y las tecnologías de información, conforman el campo de la computación como un área de conocimiento debidamente establecida. En este sentido, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) (2010) define a la ingeniería del software como "la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; esto es, la aplicación de la ingeniería al software. Este libro es una introducción a los conceptos, fundamentos, principios,

métodos y prácticas que los ingenieros de software emplean para desarrollar, operar y mantener sistemas de software que resuelvan problemas de información, conocimiento y automatización en diferentes contextos o dominios de aplicación.

1.1 ¿Qué hace a este libro diferente?

Este libro representa una visión particular de los autores, que es el resultado de una amplia experiencia docente de más de treinta y cinco años en la enseñanza de asignaturas vinculadas con la ingeniería del software y de una larga experiencia profesional asociada a la capacitación, consultoría y desarrollo de soluciones informáticas para un vasto número de empresas latinoamericanas.

Esta visión enfatiza dos aspectos que son fundamentales en el aprendizaje de una disciplina. Por un lado, el libro identifica, describe y relaciona todos aquellos conceptos esenciales para comprender los fundamentos de la ingeniería del *software* y de sus dos procesos más relevantes: el desarrollo y el mantenimiento. Entender estos conceptos y establecer sus relaciones es una condición necesaria en todo proceso de aprendizaje. Para ello, hemos construido una ontología de dominio, es decir, una descripción o especificación de los conceptos que comúnmente usamos los ingenieros de *software* para desarrollar y mantener este tipo de producto tecnológico. Por otro lado, el libro enfatiza y le da la debida importancia a los procesos que distinguen a la ingeniería del *software* como ingeniería, para lo cual se acudió al modelado de procesos de negocios para representar el desarrollo, operación y mantenimiento de *software* como procesos que se dan en un contexto empresarial, es decir, como parte de las actividades que se desarrollan en organizaciones. Para modelar estos procesos, utilizamos la extensión del lenguaje UML para negocios de Eriksson y Penker (2000), la cual facilita la representación y descripción gráfica de procesos de negocio en función de sus entradas, salidas, transformaciones, medios de control y descomposición funcional.

Ver estos procesos como conjuntos de actividades propias de una empresa u organización, le da al libro un carácter práctico indispensable para aplicar los métodos aquí descritos y asociar la teoría con la práctica. De allí la aplicabilidad que tienen los contenidos de este libro para orientar tanto a estudiantes, en sus procesos de aprendizaje de esta disciplina, como a los profesionales de la computación e informática, en los aspectos metodológicos que facilitan el manejo de la complejidad inherente al desarrollo, operación y mantenimiento de *software*.

1.2 ¿A quién está dirigido este libro?

Una primera versión fue publicada como manual en línea de la asignatura Fundamentos de Ingeniería del Software de la carrera de Ingeniería Informática de la Universidad Internacional de Valencia, España.

La edición actual es una extensión y actualización del mencionado manual al que se le ampliaron los capítulos originales y agregaron capítulos nuevos para darle a esta edición una aplicabilidad más allá del contexto universitario y facilitar su uso en ámbitos organizacionales asociados al desarrollo y mantenimiento de sistemas de *software*. El libro está, por lo tanto, dirigido a dos tipos de lectores:

- Estudiantes de Computación e Informática que cursen la asignatura Ingeniería del Software o cualquier otra relacionada con los procesos de desarrollo y mantenimiento de *software*, tales como: análisis de sistemas, diseño de software, arquitectura de software, programación y pruebas de *software*.
- Profesionales dedicados al desarrollo y mantenimiento de *software* que deseen mejorar o actualizar sus conocimientos y adquirir nuevas competencias en la aplicación de métodos, mejores prácticas y técnicas ampliamente utilizadas en la industria del *software*.

1.3 ¿Cómo está organizado este libro?

Los cuatro primeros capítulos introducen los conceptos sobre los cuales se basa el resto del contenido y que permiten comprender mejor los conceptos y términos empleados. El primero de ellos describe la disciplina de la ingeniería de *software* en el contexto de las ciencias de la computación, el cuerpo de conocimientos que respalda la formación en competencias profesionales incluidas en esta disciplina y los diferentes roles que los profesionales formados pueden ejecutar; igualmente, presenta el concepto de *software* como producto abstracto, las reglas que rigen su evolución pasando por diferentes fases de su ciclo de vida, así como los enfoques y paradigmas que definen su ciclo de desarrollo.

Los capítulos 5, 6 y 7 se centran en los procesos técnicos de desarrollo asociados con las etapas típicas del ciclo de desarrollo de *software*: análisis y especificación de requisitos, diseño de *software*, implementación y pruebas.

Los capítulos 8 y 9 describen los procesos de gestión de proyectos y apoyo a los procesos técnicos del desarrollo de *software*. Primero, se introducen conceptos y procesos asociados con la calidad del producto, su relación con los procesos

técnicos de desarrollo; luego se detalla el proceso de gestión que incluye la planificación, la dirección, el seguimiento, el control y el cierre de proyectos de *software*. Especial énfasis se hace en: (1) la organización y estructuración de los procesos, (2) la elaboración de documentos esenciales para desarrollar y mantener un sistema, (3) la gestión de recursos claves implicados en la ejecución del proyecto y (4) la interacción entre el cliente y el equipo de trabajo.

El capítulo 10 presenta de manera general el proceso de mantenimiento de productos de *software* llevándonos a completar la descripción de las etapas del ciclo de vida de un producto de *software*. Este proceso es fundamental para asegurar y prolongar la utilidad del producto de *software* desarrollado y se describe desde las perspectivas técnica y gerencial.

Finalmente, el capítulo 11 presenta el método Blue Watch, como ejemplo de una guía metodológica actualizada, sencilla y adaptable que integra los procesos técnicos de desarrollo y operación de productos de *software* con el proceso de gestión de proyectos. Este método se fundamenta en un conjunto selecto de prácticas de amplio uso en la industria del *software*, incluyendo las prácticas ágiles del modelo de procesos SCRUM y el enfoque unificado de desarrollo y operación conocido como DevOps.

1.4 Agradecimientos

La elaboración, edición y publicación de este libro no hubiesen sido posibles sin el apoyo y respaldo de nuestras familias, colegas, amigos y, muy especialmente, nuestros estudiantes de pre y postgrado.

Agradecemos a las Dras. Patricia Rosenzweig Levy y Marysela Morillo Moreno por su decidido respaldo y colaboración en todo el proceso editorial, al Dr. Carlos Mario Zapata por aceptar gustosamente nuestra petición de elaborar el prólogo y a la Sociedad Venezolana de Computación por evaluar y avalar la primera edición de este libro.

Gracias muy especiales a todos nuestros estudiantes por permitirnos poner en práctica un proceso de enseñanza-aprendizaje enriquecedor, que nos ha facilitado cuestionar, discutir y mejorar los conceptos, propuestas metodológicas y prácticas de la ingeniería del software que han sido incluidas en este libro.

2

¿Qué es la ingeniería del *software*?

La ingeniería del *software* es una disciplina relativamente joven. Sus orígenes se remontan a 1967, año en el cual un grupo de estudios de ciencias de la computación, promovido por la Organización del Atlántico Norte (OTAN) y dirigido por el Dr. Friedrich L. Bauer, propuso crear la primera conferencia con ese nombre: *The Working Conference on Software Engineering*. Esta conferencia tuvo lugar en octubre de 1968 en Garmisch, Alemania, y es considerada como el evento científico que dio origen a la ingeniería del *software*.

Durante las seis décadas de su existencia, la ingeniería del *software* ha evolucionado a grandes pasos, explorando a lo largo de su recorrido diferentes enfoques y orientaciones para convertirse en un proceso sistemático, creativo, metódico y gestionado, es decir, un proceso de ingeniería.

La ingeniería del software se puede entender desde tres perspectivas o puntos de vista diferentes pero complementarios. En primer lugar, puede verse como una rama de la ingeniería, concebirla, también, como una disciplina propia de la computación, y, en tercer lugar, como una profesión.

En este tema se aborda el concepto de «ingeniería del *software*» desde estas tres perspectivas. A partir de esta caracterización, se establecen las bases sobre las cuales se fundamenta la profesión del ingeniero de *software*. Seguidamente, se describe el proceso de formación de este ingeniero y, finalmente, se identifican y clasifican los procesos y actividades que ellos realizan.

2.1 Definiciones de ingeniería del *software*

No existe una definición única o consensuada del término «ingeniería del *software*». Sus definiciones varían de un autor a otro y de la perspectiva o punto de vista que cada autor asume para establecer tal definición. Algunas de las más citadas son las siguientes:

La ingeniería del software es la disciplina tecnológica y gerencial relacionada con la producción sistemática y el mantenimiento de productos de software que son desarrollados y modificados a tiempo y dentro de estimados de costo (Fairley, 1987).

> *(1) La aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software; esto es, la aplicación de la ingeniería al software. (2) El estudio de enfoques tales como el indicado en (1). (IEEE, 1993).*

> *La ingeniería del software es una disciplina o área de la informática o ciencias de la computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy en día es cada vez más frecuente la consideración de la ingeniería del software como una nueva área de la ingeniería… (Pressman, 2005).*

> *La ingeniería del software es una disciplina de la ingeniería que está relacionada con todos los aspectos de la producción de software desde las etapas tempranas de especificación del sistema hasta el mantenimiento del sistema después que ha entrado en uso (Sommerville, 2016).*

Pese a las diferencias que existen en estas definiciones, hay un elemento común a todas ellas y es el hecho de considerar a la ingeniería del *software* como una disciplina o área de conocimiento formal, que está asociada a los procesos de producción de *software*. El considerarla como una disciplina tecnológica y gerencial es fundamental, pues no es posible producir *software* de alta calidad sin la aplicación de principios y procesos de naturaleza gerencial.

La ingeniería del *software* es mucho más que un enfoque, práctica o conjunto de procesos. Es una disciplina formalmente establecida que tiene una identidad propia y evoluciona a lo largo del tiempo. Ella incluye, como se verá en la sección 2.9 de este tema, una amplia colección de procesos y servicios relacionados con el *software*. Puede, entonces, definirse como aquella rama de la ingeniería que aplica el conocimiento de la computación, las matemáticas y las ciencias gerenciales para adquirir, reutilizar, desarrollar, gestionar, operar, vender, mejorar y mantener productos, servicios y procesos de *software*.

En nuestra definición es importante destacar cuatro aspectos importantes:

1. La ingeniería del *software* tiene como objeto de estudio el *software* y

los procesos asociados que se requieren para producirlo y/o prestar servicios inherentes a este tipo de tecnología.

2. Para resolver sus problemas, emplea el conocimiento que generan, fundamentalmente, la computación, las matemáticas y las ciencias gerenciales.
3. Su ámbito de aplicación es la computación. Así como la ingeniería química aplica las ciencias químicas para resolver problemas de naturaleza química, la ingeniería del *software* aplica la ciencia de la computación para resolver problemas mediante el desarrollo y uso de *software*.
4. La producción de *software* y la prestación de sus servicios deben ser gestionados para poder cumplir con restricciones de costos, tiempo, calidad y alcance. Los ingenieros de *software* deben aplicar técnicas y métodos gerenciales que faciliten el control de estas cuatro variables.

2.2 La ingeniería del *software* como ingeniería

Para entender por qué la ingeniería del *software* es considerada como tal, es decir, como una ingeniería, es necesario tener claro el concepto de «ingeniería» y el de su objeto de estudio: el *software*.

La ingeniería es una actividad humana orientada a la solución de problemas del mundo que nos rodea a través de la aplicación del conocimiento científico. Esta actividad se ejerce profesionalmente, es decir, los individuos que la llevan a cabo deben tener una formación académica de nivel universitario.

El Diccionario de la Real Academia Española, DRAE (2020), define a la ingeniería de dos maneras. En primer lugar, como el «conjunto de conocimientos orientados a la invención y utilización de técnicas para el aprovechamiento de los recursos naturales o para la actividad industrial» y, en segundo lugar, como «la actividad profesional del ingeniero». La primera de estas definiciones considera a la ingeniería como una disciplina o área de conocimientos, mientras que la segunda la define como la profesión que ejerce el ingeniero.

La palabra «ingeniero» proviene del latín *ingenium* ('ingenio'), que se asocia a la acción de generar o producir; en sentido estricto, es el ingenio o facultad que tiene el hombre para reflexionar, crear o inventar

Una definición diferente del término «ingeniería» es dada por *The American Heritage® Dictionary* (2020), el cual lo define como: «la aplicación de principios científicos y matemáticos con fines prácticos, tales como el diseño, la manufactura y la operación de estructuras, máquinas, procesos y sistemas eficientes y económicos».

De manera similar, el portal significados.com (2020) considera a la ingeniería

como «la disciplina que se vale de un conjunto de conocimientos de tipo técnico, científico, práctico y empírico para la invención, el diseño, el desarrollo, la construcción, el mantenimiento y la optimización de todo tipo de tecnologías, máquinas, estructuras, sistemas, herramientas, materiales y procesos».

Por otro lado, el Colegio de Ingeniería de la Universidad de Maine en EE. UU., define a la ingeniería como una disciplina que «combina los campos de la ciencia y las matemáticas para resolver problemas del mundo real que mejoren el mundo que nos rodea. Lo que realmente distingue a un ingeniero es su habilidad para implementar ideas siguiendo un enfoque práctico y efectivo» (UMaine, 2020).

Se puede decir, entonces, que la ingeniería es una actividad realizada por el hombre −el ingeniero− con la finalidad de resolver problemas del mundo que le rodea. El ingeniero utiliza el conocimiento científico para producir soluciones técnicas −efectivas y eficientes− a los problemas que intenta resolver. Estas soluciones deben ser de bajo costo, producidas en el menor tiempo posible, de alta calidad y con el alcance especificado mediante requisitos.

La ingeniería es también una profesión, es decir, una actividad u oficio ejercido por un individuo que tiene una formación académica que lo acredita como ingeniero. A través de esta formación, este individuo adquiere los conocimientos, competencias y habilidades necesarias para realizar las actividades propias de su profesión.

A partir de estas definiciones, se extraen varios aspectos claves que caracterizan a la ingeniería:

1. Es una actividad humana que se aprende formalmente como una disciplina en instituciones universitarias y se ejerce como una profesión.
2. Aplica el conocimiento científico a la resolución de problemas del mundo.
3. Involucra procesos de diseño, construcción, operación, mantenimiento y mejora de soluciones tecnológicas de diversa naturaleza, que van desde sistemas físicos, tales como máquinas, herramientas y *hardware*, hasta sistemas abstractos como *software*, procesos empresariales e industriales y sistemas de información.
4. El ingenio, la invención y la innovación son actividades inherentes a la ingeniería.
5. Sus soluciones deben ser ejecutadas en el menor tiempo posible, a un bajo costo, con una alta calidad y con un uso eficiente de recursos, lo cual amerita la aplicación de procesos de gestión de proyectos, tales como: planificación, dirección, seguimiento y control de estas tres variables.

Dos elementos claves del proceso de ingeniería son la eficiencia y la eficacia. Estos elementos guardan una estrecha relación con las variables fundamentales de todo proceso gerencial, tales como objetivos o resultados, alcance, costos, tiempo y recursos.

La eficiencia es una relación entre los resultados obtenidos (la solución) y los recursos utilizados para producir tales resultados. La eficiencia involucra un uso óptimo o adecuado de recursos (p. ej., personas, tiempo, costos, equipos, etc.); implica, por lo tanto, alcanzar resultados con un uso mínimo de recursos, por ejemplo, desarrollar un sistema de *software* al menor costo posible.

Por su parte, la eficacia es una relación entre la eficiencia del proceso de ingeniería y la satisfacción del usuario, como satisfacer los requisitos de calidad que los usuarios de un sistema de *software* han establecido.

Los ingenieros de *software*, en particular, producen soluciones a problemas de gestión y uso de datos, información y conocimiento, problemas de comunicación y problemas de automatización de procesos por medio de sistemas de *software*.

Un ingeniero de *software* es un profesional que resuelve problemas del mundo real mediante:

- La aplicación del conocimiento científico propio de su disciplina.
- El uso de procesos de la ingeniería que describen cómo resolver problemas complejos.
- El empleo de prácticas, técnicas y métodos gerenciales o de gestión de proyectos que le permiten producir soluciones de *software* de bajo costo, a tiempo y de alta calidad.

De acuerdo con los Lineamientos Curriculares para Programas de Pregrado en Ingeniería del Software (ACM/IEEE, 2014), elaborado por el Grupo de Trabajo Conjunto sobre Currículos de la Computación, de dos de las asociaciones profesionales de mayor alcance y reputación internacional −la Sociedad de Computación de la IEEE (*IEEE Computer Society*) y la ACM (*Association for Computing Machinery*)−, los ingenieros se caracterizan y difieren de otros profesionales, en los siguientes aspectos:

- Los científicos observan y estudian objetos para desarrollar modelos que describan la estructura y comportamiento de estos objetos, mientras que los ingenieros utilizan estos modelos como punto de partida para diseñar y desarrollar nuevas tecnologías.
- Los ingenieros toman decisiones, para lo cual evalúan diferentes opciones o alternativas de solución y seleccionan aquellas que ofrezcan la mejor relación costo-beneficio.

- Los ingenieros realizan mediciones, validan sus mediciones y usan aproximaciones basadas en la experiencia o en datos empíricos.
- Usan procesos disciplinados para ejecutar sus actividades. Trabajan en equipo. No proceden a ciegas, ni mediante prueba y error.
- Los ingenieros ejecutan múltiples roles o actividades, tales como: investigación, desarrollo, diseño, producción, pruebas, operación, construcción, gestión, ventas, consultoría y docencia.
- Los ingenieros utilizan herramientas que les permiten aplicar los procesos disciplinadamente. Estas herramientas se seleccionan cuidadosamente dependiendo del problema que se quiere resolver.
- Los ingenieros diseñan artefactos y reutilizan estos artefactos para producir otros más complejos.
- Para resolver problemas, el ingeniero de software debe poseer un conjunto básico de competencias generales:
    - Debe poseer conocimientos científicos propios de la computación, las matemáticas, la estadística y las ciencias gerenciales, los cuales aplica para analizar el problema y diseñar sus soluciones.
    - Debe emplear métodos y técnicas de gestión para asegurar que las soluciones sean eficientes y eficaces.
    - Debe poseer destrezas y habilidades para llevar a cabo sus actividades profesionales.
    - Debe utilizar herramientas adecuadas para facilitar o simplificar sus actividades.
    - Debe ser capaz de representar el problema y sus soluciones a través de modelos formales, gráficos o narrativos.

2.3 La ingeniería del *software* como disciplina de la computación

Como disciplina, la ingeniería del *software* se enmarca en la computación. Durante las décadas pasadas, la ingeniería del *software* fue considerada como un área de conocimiento propia de las ciencias de la computación. El currículo de computación propuesto por la ACM y la Sociedad de Computación de la IEEE, en el año 1991, consideraba la enseñanza de la ingeniería del *software* como una actividad más de los programas de estudio en ciencias de la computación.

Sin embargo, en las dos últimas décadas, la ingeniería del *software* se ha consolidado como una disciplina propiamente dicha que se separa de las ciencias de la computación y se asocia cada vez más a la ingeniería, tal como

lo demuestra el número considerable de programas de pre y posgrado en ingeniería del *software*, que es ofrecido por muchas instituciones académicas de Europa y América.

En el año 2001, el Grupo de Trabajo de la IEEE/ACM sobre currículos de computación reconoce, por primera vez, a la ingeniería del *software* como una disciplina con identidad propia. En el informe curricular de ese año, este grupo ve a la computación como un campo del conocimiento que se extiende más allá de los límites de lo que, tradicionalmente, se conoce como ciencias de la computación y propone un currículo para la ingeniería del *software*, separado de aquel que rige la enseñanza de las ciencias de la computación (ACM/IEEE, 2001).

Posteriormente, en el año 2005, este mismo grupo plantea una redefinición de la computación y de las disciplinas que la constituyen. Redefine el término «computación» de la manera siguiente:

Computación significa cualquier actividad orientada a metas que requiere, se beneficia o crea computadores. En consecuencia, la computación incluye el diseño y producción de hardware y software; el procesamiento, estructuración y gestión de varios tipos de información; la elaboración de estudios científicos usando computadores; el hacer sistemas de computación que se comporten inteligentemente; crear y usar medios de comunicación y entretenimiento; buscar y recopilar información relevante a cualquier propósito [usando computadores], etc. La lista es virtualmente ilimitada (ACM/IEEE, 2005).

De acuerdo con este grupo de trabajo, la computación se entiende como una familia de disciplinas estrechamente vinculadas, cada una con su propia identidad y con su propio currículo o plan de estudio. Esta familia está integrada por las siguientes disciplinas:

- Ciencias de la computación.
- Ingeniería de computadores.
- Ingeniería del *software*.
- Sistemas de información.
- Tecnologías de información.

La ingeniería del *software* difiere de las otras disciplinas de la computación, pero está estrechamente vinculada a ellas. Se alimenta de las Ciencias de la Computación para producir tecnologías digitales, concretamente, *software*. Utiliza los productos de la ingeniería de computadores para producir sistemas de *software*. Apoya el desarrollo y mantenimiento de sistemas de información y crea nuevas tecnologías para el procesamiento de información en computadores.

Cada una de estas disciplinas posee su propia guía curricular. La versión más reciente del Currículo para Ingeniería del Software fue publicada en el año 2014 (ACM/IEEE, 2014). Actualmente, esta versión se encuentra en revisión.

Como se puede apreciar, la ingeniería del *software*, además de ser una rama de la ingeniería, es también una disciplina de la computación con identidad propia y mundialmente reconocida como tal.

2.4 La ingeniería del *software* como profesión

La ingeniería del *software*, vista como una profesión, es bastante joven comparada con las ingenierías tradicionales y tiene aún mucho que aprender de ellas. Existen aspectos que caracterizan a las ingenierías tradicionales que todavía no han sido dominados por la ingeniería del *software*; es decir, no se practican con la rigurosidad que una ingeniería establecida requiere. Denning y Riehle (2009) describen seis de estos aspectos: la predictibilidad de los resultados, el uso de métricas de diseño, la tolerancia a fallas, la separación entre diseño e implementación, la reconciliación entre restricciones o requisitos conflictivos y la adaptación a ambientes cambiantes. Estos aspectos han ido cambiando en los últimos años gracias a los avances de la ingeniería del *software* como disciplina establecida. Así, por ejemplo, la predictibilidad de los resultados del proceso de ingeniería ha sido abordado por la ingeniería del *software* a través de la aplicación de los procesos de gestión de proyectos y el uso de una amplia variedad de métodos ágiles y disciplinados que existen actualmente y cuya aplicación al desarrollo de *software* permiten predecir el resultado de un proyecto de desarrollo de *software*. De igual manera, la adaptación a ambientes cambiantes se ha abordado a través del uso de métodos ágiles y la aplicación de principios y prácticas de la ingeniería de requisitos, una de las subramas de la ingeniería del *software*.

Si bien es cierto que en la comunidad científica existen aún sectores que dudan acerca del carácter profesional de la ingeniería del *software*, no menos cierto es que la ingeniería del *software* ha sentado, en los últimos años, las bases de su consolidación como una profesión.

Una profesión consolidada tiene cuatro elementos o pilares sobre los cuales se apoya:

1. Un cuerpo de conocimientos.
2. Un código de ética.
3. Modelos curriculares.
4. Programas de certificación profesional.

El primero de ellos describe y organiza el conocimiento que es, generalmente, aceptado por su comunidad profesional. El segundo establece los principios del ejercicio de la profesión, la responsabilidad social que los profesionales tienen y los compromisos que ellos adquieren con la sociedad. El tercero describe, a través de modelos curriculares, cómo se debe formar un ingeniero y cuáles son los conocimientos y competencias que debe adquirir durante sus estudios universitarios. Finalmente, el cuarto pilar permite evaluar si un profesional posee los conocimientos y competencias que su profesión demanda.

El trabajo conjunto de la Sociedad de Computación de la IEEE y la ACM han elaborado estos cuatro elementos, que son esenciales para el desarrollo de la ingeniería de *software* como profesión.

2.4.1 El cuerpo de conocimientos de la ingeniería del software – SWEBOK

El cuerpo de conocimientos de una disciplina está constituido por toda su literatura. Para facilitar el manejo y acceso a este conocimiento, las organizaciones profesionales de dicha disciplina crean las guías o manuales que seleccionan, organizan y documentan la literatura más influyente que conforma dicho cuerpo de conocimientos.

La guía del cuerpo de conocimientos de la ingeniería del *software*, elaborada por la Sociedad de Computación de la IEEE y mejor conocida como Guía SWEBOK (IEEE CS, 2014), describe el conocimiento que es generalmente aceptado y que caracteriza a la ingeniería del *software*. Uno de sus usos más importantes está en la elaboración de programas curriculares y de certificación profesional en ingeniería del *software*.

La guía SWEBOK persigue tres fines relacionados:

1. Identificar aquellas partes del cuerpo de conocimientos sobre las cuales existe un consenso generalizado entre los miembros de la comunidad académica y profesional de la ingeniería del *software*.
2. Organizar estas partes en áreas de conocimiento.
3. Crear los medios necesarios para acceder a ellas.

Para cumplir con estos objetivos la IEEE CS ofrece esta guía en varios formatos, dos de ellos se distribuyen de manera gratuita y están disponibles en la dirección www.swebok.org. La versión 3.0 de la guía SWEBOK organiza el conocimiento de la ingeniería del *software* en quince secciones denominadas áreas de conocimiento. Estas áreas son las siguientes:

- Requisitos de *software*.
- Diseño de *software*.
- Construcción de *software*.
- Pruebas de *software*.
- Mantenimiento de *software*.
- Gestión de la configuración del *software*.
- Gestión de la ingeniería del *software*.
- Procesos de ingeniería del *software*.
- Modelos y métodos de ingeniería del *software*.
- Calidad del *software*.
- Práctica profesional de la ingeniería del *software*.
- Economía de la ingeniería del *software*.
- Fundamentos de computación.
- Fundamentos matemáticos.
- Fundamentos de ingeniería.

Para cada una de estas áreas la guía describe los conceptos fundamentales del área e indica cuál es su literatura más importante. Es, por lo tanto, un recurso educativo de suprema importancia en la formación de los ingenieros de *software*.

2.4.2 El código de ética de la ingeniería del software

El Código de Ética y Práctica Profesional de la Ingeniería del Software fue elaborado conjuntamente por la ACM y la Sociedad de Computación de la IEEE, como un estándar para la enseñanza y práctica de la ingeniería del *software*.

El código prescribe las obligaciones éticas que un ingeniero de *software* debe cumplir. Tiene, además, una función educativa importante por cuanto es también un medio para educar a quienes aspiran ejercer la profesión de ingeniero de *software*, acerca de cómo actuar y cómo comportarse profesionalmente. Está compuesto por ocho principios fundamentales (ACM, 2020):

1. **Sociedad**. Los ingenieros de *software* actuarán en forma congruente con el interés social.
2. **Cliente y patrón**. Los ingenieros de *software* actuarán de manera que se concilien los mejores intereses de sus clientes y patrón, congruentemente con el interés social.
3. **Producto**. Los ingenieros de *software* asegurarán que sus productos y modificaciones correspondientes cumplan con los estándares profesionales más altos posibles.

4. **Juicio**. Los ingenieros de *software* mantendrán integridad e independencia en su juicio profesional.
5. **Administración**. Los ingenieros de *software*, gerentes y líderes promoverán y se suscribirán a un enfoque ético en la administración del desarrollo y mantenimiento de *software*.
6. **Profesión**. Los ingenieros de *software* incrementarán la integridad y reputación de la profesión congruentemente con el interés social.
7. **Colegas**. Los ingenieros de *software* apoyarán y serán justos con sus colegas.
8. **Personal**. Los ingenieros de *software* participarán toda su vida en el aprendizaje relacionado con la práctica de su profesión y promoverán un enfoque ético en la práctica de la profesión.

2.4.3 El modelo curricular de la ingeniería del software

Así mismo, mediante un esfuerzo conjunto entre la ACM y la Sociedad de Computación de la IEEE, se elabora y actualiza periódicamente el Modelo Curricular de la Ingeniería del Software (ACM/IEEE, 2014). Este modelo sirve de guía a las instituciones académicas que requieran elaborar programas de estudio para carreras en ingeniería del *software*. El modelo identifica y define las áreas de conocimiento que son fundamentales para la formación de ingenieros de *software* y proporciona los lineamientos para la enseñanza de estas áreas (ver figura 2.1).

Para cada una de estas áreas de conocimiento, el modelo curricular propone las unidades y tópicos que deben ser cubiertos por los cursos o asignaturas que integran un programa de estudios en ingeniería del *software*.

2.4.4 Los programas de certificación profesional

Un último elemento que complementa la caracterización de la ingeniería del *software* como una profesión es la certificación. La certificación es un medio que contribuye a verificar y medir el conocimiento fundamental y las competencias que un profesional tiene en relación con una o más áreas de la ingeniería del *software*. Existen diferentes programas de certificación. Muchos de ellos son de carácter instrumental, esto es, están orientados a medir el conocimiento y las habilidades que un profesional tiene en el uso de herramientas de *software*. Desde la perspectiva profesional, existen programas promovidos por asociaciones profesionales, tales como el CSDA/CSDP (*Certified Software Development Associate* / *Certified Software Development Professional*) de la Sociedad

de Computación de la IEEE y el CITP (*Chartered IT Professional*) de la Sociedad Británica de Computación.

Figura 2.1. Las principales áreas de conocimiento del currículo de ingeniería del software.

2.5 ¿Qué hace un ingeniero de *software*?

Los ingenieros de *software* resuelven problemas de datos, información, conocimiento, comunicación y automatización mediante la ejecución de procesos de *software*, tales como gestión, desarrollo, operación, mantenimiento, reutilización, adquisición, mejora y reutilización de productos y servicios de *software*.

Un proceso de *software* es un conjunto estructurado de actividades relacionadas, cuya ejecución contribuye a alcanzar objetivos predefinidos que están, generalmente, relacionados con la solución de problemas de información, comunicación y/o automatización a través de productos y servicios de *software*.

La guía SWEBOK (IEEE CS, 2014) define un proceso de *software* como "un conjunto de actividades interrelacionadas que transforman productos de trabajo de entrada en productos de trabajo de salida". Un producto de trabajo es un objeto de información (p. ej., un diseño, una especificación, un modelo, una pieza de código o programa, un recurso o un documento). Este objeto es utilizado como insumo (entrada) o transformado como producto (salida) durante la ejecución de un proceso de *software*.

Tal como se ilustra en la figura 2.2, un proceso se puede descomponer o subdividir en un conjunto de subprocesos o actividades formando una estructura jerárquica de descomposición del trabajo que puede tener varios niveles de desagregación.

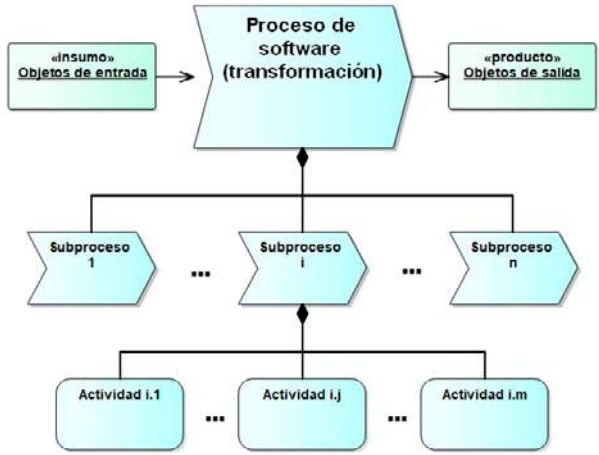

Figura 2.2 Descomposición de un proceso de software en subprocesos y actividades.

El desarrollo de *software* es uno de los principales procesos que realizan los ingenieros de *software*. James Peters y Witold Pedrycz definen a este proceso, desde la perspectiva de la ingeniería, como sigue:

> *Un enfoque de ingeniería al desarrollo de software se caracteriza por la aplicación de principios, métodos, modelos, estándares y teorías científicas que hacen posible administrar, planificar, modelar, diseñar, implementar, medir, analizar, mantener y evolucionar un sistema de software (Peters & Pedrycz, 2000).*

Si bien el desarrollo de *software* es uno de los procesos más conocidos y cotizados en el mercado laboral, no es el único proceso que los ingenieros de *software* pueden ejecutar. Existe una amplia variedad de procesos que un ingeniero de *software* es capaz de realizar como parte del ejercicio de su profesión. El estándar ISO/IEC 15504 (2004), también denominado SPICE (*Software Process Improvement Capability dEtermination*), clasifica estos procesos en las categorías que se describen seguidamente.

2.5.1 Procesos relacionados con clientes y proveedores

Estos procesos apoyan la contratación, el desarrollo y la entrega del *software* al cliente y contribuyen a que su uso y operación sea correcto. Se subdividen en:

- Adquisición de productos y/o servicios de *software*.
- Establecimientos de contratos con clientes y proveedores.
- Identificación de las necesidades del cliente.
- Realización de auditorías y revisiones conjuntas.
- Entrega e instalación del *software*.
- Apoyo en la operación del *software*.
- Prestación de servicios al cliente.
- Valoración de la satisfacción del cliente.

2.5.2 Procesos de ingeniería

Esta categoría agrupa aquellos procesos relacionados con el desarrollo y mantenimiento de sistemas de *software*, así como la documentación para los usuarios. Se divide como sigue:

- Desarrollo de requisitos y diseño del sistema.
- Desarrollo de requisitos del *software*.
- Diseño del *software*.
- Implementación del *software*.
- Integración y pruebas del *software*.
- Mantenimiento del *software*.

2.5.3 Procesos de gestión de proyectos

Consiste en procesos relacionados con el establecimiento del proyecto y la coordinación y gestión de los recursos necesarios para producir un producto o prestar un servicio de *software* que satisfaga al cliente. En esta categoría están los siguientes subprocesos:

- Planificación del ciclo de vida del proyecto.
- Elaboración del plan del proyecto.
- Organización de los equipos de trabajo que ejecutarán el proyecto.
- Gestión de requisitos.
- Gestión de la calidad.
- Gestión de riesgos.
- Gestión de recursos y del cronograma del proyecto.
- Gestión de subcontratistas.

2.5.4 Procesos de soporte

Incluye aquellos procesos que pueden ser empleados por otros procesos de *software* de cualquier categoría (incluyendo esta). Pueden emplearse en otro punto del ciclo de vida del *software*, ya sea por los equipos de trabajo, el cliente o una organización independiente fuera del contexto del proyecto. Se dividen en:

- Desarrollo de la documentación.
- Gestión de la configuración del *software*.
- Aseguramiento de la calidad.
- Resolución de problemas.
- Revisiones de *software*.

2.5.5 Procesos organizacionales

Agrupa procesos que están relacionados con la organización donde se elaboran los productos de *software*. Están vinculados con los objetivos de esta organización y se encargan de elaborar procesos, productos o activos que ayuden a alcanzar estos objetivos. Esta categoría está dividida en:

- Alineación de los procesos con la organización.
- Establecimiento de procesos de *software*.
- Evaluación de procesos de *software*.
- Mejora de procesos de *software*.
- Gestión de recursos humanos.
- Gestión de la infraestructura.
- Reutilización.

Además de estos procesos fundamentales, los ingenieros de *software* realizan procesos que son ortogonales o transversales, es decir, están presentes en cada uno de los procesos arriba mencionados. Son los siguientes:

- **Procesos de comunicación**. La mayoría de los procesos, tanto fundamentales como ortogonales, son ejecutados por equipos de trabajo en los cuales participan varios actores con diferente formación, experiencia y conocimiento. Estos actores requieren comunicarse continuamente. La comunicación verbal y escrita es esencial para discutir, expresar y transferir los resultados de la ejecución de estos procesos.
- **Procesos de toma de decisiones**. Durante la ejecución de cualquier proceso es común encontrar situaciones en las que

los equipos de trabajo deben escoger entre varias alternativas de solución. Los procesos de toma de decisiones contribuyen a resolver estas situaciones.

- **Procesos de modelado**. La naturaleza abstracta del *software* hace que el ingeniero de *software* se mueva en un mundo de conceptos, símbolos, sistemas abstractos y modelos. El modelado es uno de los procesos que el ingeniero ejecuta con mayor frecuencia durante el desarrollo y mantenimiento. Para modelar un sistema, el ingeniero de *software* requiere el uso de lenguajes artificiales, particularmente, lenguajes gráficos y/o formales. El lenguaje de modelado unificado UML es considerado, actualmente, como la lengua franca del desarrollo de *software*. Usando estos lenguajes, el ingeniero elabora modelos que capturan diferentes aspectos de un sistema. Estos modelos se elaboran, por lo general, usando herramientas de *software* que automatizan el proceso de creación de modelos y su transformación.

Como puede apreciarse, en la clasificación dada anteriormente, los procesos en los que un ingeniero de *software* puede participar son muy amplios. Para llevar a cabo las actividades que estos procesos requieren, los ingenieros de *software* deben aplicar el conocimiento científico que proveen la computación, las ciencias gerenciales y otras ciencias básicas como la lógica y las matemáticas. Este conocimiento es aplicado a través de los procesos y enfoques de resolución de problemas de la ingeniería, por ejemplo, los enfoques de sistemas y el pensamiento sistémico que caracterizan a la Ingeniería de Sistemas. La aplicación del conocimiento propio de las ciencias gerenciales es, también, esencial para el ejercicio de esta profesión. Mediante la aplicación de los conceptos, principios, métodos y técnicas gerenciales, los ingenieros de *software* pueden controlar las tres variables fundamentales de la ingeniería: costos, tiempos y calidad de la solución.

2.6 Instrumentos que emplean los ingenieros de *software*

Durante la resolución de sus problemas, los ingenieros de *software* emplean un conjunto de instrumentos conceptuales, tecnológicos y metodológicos que incluye marcos conceptuales, principios, métodos, modelos, procesos, técnicas, lenguajes, patrones, herramientas y mejores prácticas.

Una analogía interesante, que nos permite caracterizar al ingeniero de *software*, es la de un médico de familia. Estos profesionales atienden a sus pacientes usando los instrumentos y medicamentos contenidos en su maletín médico. De manera análoga, los ingenieros de *software* resuelven los problemas de información y automatización de sus clientes usando un maletín virtual que contiene los instrumentos arriba señalados.

Cada situación amerita el uso de instrumentos específicos. Son las características del problema y de su solución las que determinan los instrumentos que el ingeniero debe utilizar en cada situación. Por ejemplo, durante los procesos de análisis y diseño de *software* es común utilizar un método de desarrollo de *software* en conjunto con el lenguaje de modelado UML y una herramienta de *software* que automatice la aplicación de este lenguaje. De igual manera, durante los procesos de programación, integración y pruebas se utilizan los lenguajes de programación, los ambientes integrados de programación y los marcos de trabajo.

Acerca de los autores

JONÁS ARTURO MONTILVA CALDERÓN es profesor titular jubilado del Departamento de Computación de la Facultad de Ingeniería de la Universidad de Los Andes (ULA) en Mérida, Venezuela. Es ingeniero de Sistemas egresado de la ULA. Obtuvo el grado de M. Sc. en Computación y Ciencias de la Información de la Universidad Case Western Reserve en Cleveland, Ohio, EE. UU., y el grado de Ph. D. en Estudios de la Computación en la Universidad de Leeds, Leeds, Inglaterra. Fue fundador del postgrado en Computación y cofundador del Consejo de Computación Académica y de la Coordinación de Estudios Interactivos a Distancia de la ULA. Fue profesor invitado del Departamento de Ciencias e Ingeniería de la Computación de la Universidad del Sur de Florida, Tampa, EE. UU. Por cuatro años consecutivos fue visitante distinguido de la Sociedad de Computación de la IEEE, región 9. Fue director académico del Centro de Excelencia en Ingeniería de Software (CEISOFT) del parque tecnológico de la ULA. Es socio fundador y CEO de BIOSOFT C. A., empresa de consultoría, desarrollo de *software* y capacitación profesional ubicada en Mérida, Venezuela. Es miembro correspondiente estadal de la Academia de Mérida. Sus principales líneas de investigación son ingeniería del *software*, arquitecturas empresariales, educación virtual, ontologías informáticas y tecnologías emergentes.

JUDITH BARRIOS ALBORNOZ es profesora titular jubilada del Departamento de Computación de la Facultad de Ingeniería de la Universidad de Los Andes, Mérida Venezuela. Es ingeniera de Sistemas de la Universidad de Los Andes (ULA-Venezuela), tiene una maestría en Sistemas de Información del ITESM de Monterrey (México), un diploma de Profundización DEA en Teoría e Ingeniería de Bases de Datos y un doctorado en Informática de la Universidad de Paris 1 Panteón-Sorbona (Francia). Fue coordinadora del postgrado en Computación de la ULA en varias oportunidades. Es profesora e investigadora invitada anualmente por la Universidad de Paris 1 (desde el 2004) y por el Conservatorio Nacional de Artes y Oficios (CNAM París) desde el 2017. Es socia y consultora de BIOSOFT C.A. Sus áreas de trabajo principales son ingeniería del software, ingeniería de requisitos, ingeniería de métodos, sistemas de información, gestión de cambios organizacionales, modelado de negocios y arquitectura empresarial

Adquiéralo en amazon.com