



Uniciencia
ISSN: 1011-0275
ISSN: 2215-3470

Universidad Nacional, Costa Rica

Garita-González, Gabriela; Lizano-Madriz, Fulvio
Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO
Uniciencia, vol. 32, núm. 1, 2018, Enero-Junio, pp. 118-133
Universidad Nacional, Costa Rica

DOI: <https://doi.org/10.15359/ru.32-1.8>

Disponible en: <https://www.redalyc.org/articulo.oa?id=475958170008>

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

Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO

Software Cost Estimation: A Pedagogical Proposal to Teach COCOMO

Gabriela Garita-González

ggarita@uned.ac.cr

Dirección de Producción de Materiales Didácticos

Universidad Nacional a Distancia UNED

San José, Costa Rica

Fulvio Lizano-Madriz

fлизано@una.cr

Escuela de Informática

Universidad Nacional

Heredia, Costa Rica

Recibido-Received: **8/jun/2016** / Corregido-Corrected: **23/mar/2017**.

Aceptado-Accepted: **20/abr/2017** / Publicado-Published: **31/ene/2018**.

Resumen

Los planes de estudio de ciencias de la computación y carreras afines evidencian una brecha importante en torno a la temática de estimación de costos de proyectos de software. En general, en los modelos curriculares relacionados con el desarrollo de software, se recomienda la temática de la estimación en el desarrollo del software sin especificar ningún modelo de estimación en concreto. En abono a lo anterior, dichos modelos tienen la particularidad de que son difíciles de entender y los ejemplos existentes en la bibliografía suelen ser muy vagos sin considerar contextos cercanos a la realidad. En este artículo se presenta una propuesta de aplicación pedagógica del modelo de estimación de costos COCOMO en las modalidades básica e intermedia orientada a la magnitud del producto final. El modelo incluye seis apartados necesarios para un adecuado abordaje pedagógico. Lo primero que se define es su fundamento teórico esquematizado de forma resumida. Además, se presenta un caso real de estudio, ejemplos resueltos, la propuesta pedagógica y el análisis respectivo. La relevancia de este trabajo se fundamenta en el hecho de que se facilita la enseñanza del tema de estimación del costo del software, de forma práctica y contextualizada en un caso real.

Palabras claves: COCOMO básico; COCOMO intermedio; estimación de costos del software; ingeniería de software; propuesta pedagógica de enseñanza de COCOMO.

Abstract

The computer science curriculum and other related academic programs show a relevant gap on software cost estimation. Usually, curriculum models related to software development recommend the inclusion of software estimation topics without specifying any particular estimation model. In addition, software estimation models are difficult to understand and the existing examples are often vague, and this without considering actual contexts. In this paper we present a pedagogical proposal to teach COCOMO in its basic and intermediate modalities, oriented to the magnitude of

the final product. The model includes six sections required for a proper pedagogical approach. First, the theoretical background is explained in a very simple way. Then, there is a presentation of an actual case study, several examples, a pedagogical approach, and the respective analysis. This work is important because it is now possible to teach the software cost estimation topic by using a practical model contextualized in an actual case.

Keywords: basic COCOMO; intermedia COCOMO; software cost estimation; software engineering; pedagogical proposal.

Resumo

Os planos de estudo de ciências da computação e carreiras afins evidenciam uma brecha importante em torno da temática de estimativa de custos de projetos de software. Em geral, nos modelos curriculares relacionados com o desenvolvimento de software, recomenda-se a temática da estimativa no desenvolvimento do software sem especificar nenhum modelo de estimativa de maneira concreta. Reforçando o anterior, tais modelos têm a particularidade de que são difíceis de entender e os exemplos existentes na bibliografia tendem a ser muito vagos sem considerar contextos próximos à realidade. Neste artigo é apresentada uma proposta de aplicação pedagógica do modelo de estimação de custos COCOMO nas modalidades básica e intermediária orientadas à magnitude do produto final. O modelo inclui seis seções necessárias para uma adequada abordagem pedagógica. O primeiro que se define é seu fundamento teórico esquematizado de forma resumida. Além disso, são apresentados um caso real de estudo, exemplos solucionados, a proposta pedagógica e a análise respectiva. A relevância deste trabalho se fundamenta no fato de facilitar o ensino do tema de estimação do custo do software de forma prática e contextualizada em um caso real.

Palavras-chaves: COCOMO básico; COCOMO intermediário; estimativa de custos do software; engenharia de software; proposta pedagógica de ensino de COCOMO.

Las razones para utilizar modelos algorítmicos de estimación de costo en proyectos de software son muchas. [Boehm et al. \(2000\)](#) indican que estos modelos son utilizados para, entre otras cosas, por razones presupuestarias, ayudar en la gestión de riesgos, planificar y controlar el proyecto, enriquecer el análisis de inversión de los proyectos de software.

Los modelos algorítmicos de estimación de costo en proyectos de software tienen un alcance específico a contextos limitados. No hay uniformidad en la aplicación afectiva de un determinado modelo sobre otro. La comparación de los resultados obtenidos en varios modelos parece ser la forma adecuada de obtener estimaciones realistas ([Boehm et al., 2000](#)).

Otras investigaciones han ahondado en las limitaciones de los modelos algorítmicos de estimación de costo. En un estudio que evaluó cuatro de los modelos algorítmicos más populares utilizados para estimar los costos de software (SLIM, COCOMO, puntos de función, y ESTIMACS), [Kemerer \(1987\)](#) comprobó la exactitud de la estimación del esfuerzo a posteriori. En este estudio, se concluyó que los modelos no desarrollados en entornos de procesamiento de datos empresariales mostraron gran necesidad de calibración. Los autores del estudio argumentan que como modelos del proceso de desarrollo de software, los modelos fallaron para reflejar con suficiencia los factores subyacentes que afectan la productividad. En abono a lo anterior, [Leung y Fan \(2002\)](#) explican que estas limitaciones están basadas en el gran número de factores interrelacionados que influyen en el proceso de desarrollo de software de un equipo de desarrollo dado y el gran número de atributos del proyecto. Además, el entorno de desarrollo está en constante evolución y existe una evidente falta de medición de la complejidad real de un determinado sistema de software.

Si bien es cierto que los modelos de estimación de costo en proyectos de software han sido utilizados desde hace mucho tiempo en la industria, la exactitud de los resultados del uso de estos modelos ha sido objeto de constante debate. Al respecto, [Lewis \(2001\)](#) ha introducido el tema de la complejidad algorítmica como una indicación de los límites en la estimación de software. En concreto, según Lewis, la complejidad algorítmica es una definición adecuada de la complejidad de un proyecto de programación. En este sentido, esta complejidad hace imposible la estimación objetiva de la complejidad del proyecto, el tiempo de desarrollo y la productividad de los programadores.

Otro referente interesante de la aplicación de estos modelos de estimación de costo en la industria lo proporciona [Yang et al. \(2008\)](#). En su estudio hecho en 112 proyectos de software en China, se determinó que los grandes proyectos de software eran más propensos a los costos y sobrecostos de calendarización, que la mayoría de gerentes de TI y profesionales practicantes no se mostraban ni satisfechos con la estimación de proyectos, que solo 15% de las organizaciones utilizaban métodos basados en modelos, y que el alto costo de la adopción versus el beneficio mínimo después de la adopción fueron las principales causas de la baja utilización de métodos basados en modelos.

No obstante lo anterior, a pesar de todas estas críticas, parecía existir un consenso respecto a que el mayor beneficio de los modelos de estimaciones de costos en software es proveer una clara y consistente argumentación para manejar los problemas de ingeniería de software que se presentan a lo largo del ciclo de vida de este ([Boehm, 1981](#)).

En el caso específico de COCOMO, algunos estudios han realizado validaciones empíricas sobre el particular y han obtenido buenos resultados sobre las posibilidades del modelo ([Basavaraj y Shet, 2008](#)).

Adicionalmente, COCOMO no solamente es un modelo de estimación de costos relativamente bien conocido y utilizado, sino también está en plena evolución. [Sharma \(2013\)](#) propuso recientemente una nueva extensión de COCOMO llamada E-COCOMO. Esta extensión ataca el problema del retrabajo en los proyectos de software por medio de la incorporación de nuevos manejadores de costo.

Considerando estos aspectos negativos y positivos de modelos como COCOMO, se podría especular que entre otros factores que entran en juego en el debate, el tema de cómo se enseña COCOMO tiene algo que ver. No obstante lo conocido que el modelo COCOMO es en la industria, este es un tópico considerado de forma marginal en los procesos formales de enseñanza. Es difícil encontrar en la bibliografía algunos ejemplos de abordaje de COCOMO en la academia. [McDonald \(2001\)](#) ha propuesto algunas alternativas interesantes en el tema de estimación con la utilización de COCOMO y otras técnicas o modelos desde una perspectiva de capacitación en la gestión de proyectos de software. [Mehmood y Saeed \(2009\)](#) agregaron, a esta perspectiva de gestión de proyectos, la arista de la gestión de la calidad para utilizar COCOMO y otras técnicas o modelos. No obstante, el uso específico de COCOMO no es protagonista en los procesos de formación. El tema sugerido es el de estimación en general.

Así las cosas, por ejemplo, en la ACM-IEEE currícula de Ciencias de la Computación para nivel de bachillerato universitario, en el curso Gestión de Proyectos de Software, se incluye el tópico técnicas de medición y estimación de software con un objetivo de aprendizaje orientado

a la preparación de planes de proyectos que incluyan, entre otros aspectos, estimaciones del tamaño y esfuerzo ([Cassel et al., 2008](#)). Más ampliamente, las guías de currículo también para bachillerato universitario en Ingeniería del Software incluyen el tema de estimación de costos en los cursos de Estadística y Métodos Empíricos para Computación, Gestión de Proyectos de Software, Gestión de Procesos de Software y Proyecto Final de Ingeniería del Software ([LeBlanc et al., 2006](#)). Las guías de currículo para Bachillerato Universitario en Ingeniería de Sistemas ([Topi et al., 2010](#)) solo incluyen el tema de forma general como estimación de tanto costos como de beneficios en el curso de Fundamentos de Sistemas de Información. El tema de estimación de costos también se incluye a nivel de posgrados ([Pyster, 2009](#); [Gorgone et al., 2006](#)). Aunque el tema de estimación de costos está presente en esta propuestas de lineamientos guía para planes de estudio, existe una carencia significativa de propuestas concretas de abordaje de la enseñanza del tema de los modelos de estimación de costos en los proyectos de software.

Especificamente en el caso de COCOMO es aún más difícil encontrar ejemplos tropicalizados de cómo enseñar este importante tema. Entre otros pocos, el trabajo de [Garita \(2014\)](#) presenta varios modelos para el cálculo y la estimación de proyectos de software integrados a las disciplinas de la ingeniería de software. Considerando lo anterior, en este artículo se muestra una propuesta con enfoque pedagógico, de abordaje de los modelos algorítmicos de estimación de costo de software, enfocándose en COCOMO.

La estructura del artículo tiene la finalidad de facilitar la aplicación del modelo pedagógico con la modalidad híbrida ([Doering y Veletsianos, 2008](#)) y virtual ([Dabbagh, 2005](#)). Por esta razón, en el apartado 2 se detalla el caso de estudio: Pel ® S.A el que será de gran utilidad en el modelo pedagógico propuesto. El apartado 3 incluye el fundamento teórico y contenidos del modelo COCOMO básico e intermedio.

En el apartado 4 de este artículo, el público lector encontrará 4 ejemplos prácticos de aplicación del modelo COCOMO que utilizan como base el caso de estudio y el fundamento teórico previamente presentado. En el apartado 5 se encuentra el modelo pedagógico que detalla la audiencia, modalidad, tiempo estimado, descripción general y diseño de la actividad. Este diseño incluye el objetivo de aprendizaje, los contenidos y la estrategia.

El apartado 6 de este artículo presenta un breve análisis de los contenidos del artículo, el cual finaliza en el apartado 7 con las conclusiones y trabajos futuros. Los apartados 2, 3 y 5 guiarán a estudiantes a identificar, analizar y resolver un problema propio de la ingeniería de software: la estimación de un proyecto de software. Por su parte, los apartados 1, 4, 6 y 7 apoyan el trabajo de docentes y profesionales en la aplicación práctica del modelo COCOMO.

El caso de estudio: PEL ® S.A

En el 2009, se funda la empresa de Pagos Electrónicos PEL S.A con el objetivo de desarrollar soluciones de software para transacciones financieras, gestiones de pagos y cobros electrónicos, todo esto a través de un proceso de recaudación en línea interconectado con las diferentes entidades financieras del país.

El 4 de enero, después de una negociación, ingresa una petición de un nuevo desarrollo, en el cual se incluirán las funcionalidades de gestiones de pagos, para que la clientela pueda realizar de forma segura y recurrente recaudaciones en línea de los diversos servicios que ofrece. La gerencia

de Pel ® la ejerce el arquitecto o arquitecta funcional de la empresa, controla los proyectos y cambios de sus productos principales. El tipo de desarrollo que enfrenta su equipo de trabajo contendrá unas 100.000 líneas de código (100 KLDC). Entre las principales especificaciones: el cliente necesita un software de alta fiabilidad, pues el tipo de transacciones del nuevo sistema involucra pérdidas fácilmente recuperables, hasta grandes pérdidas financieras; se requiere que un alto porcentaje del equipo de trabajo tenga al menos 1 año de experiencia en el desarrollo de aplicaciones similares, por lo que cada salario mensual de quienes lo desarrollan se promedia en \$4.000. Este tipo de desarrollo hace necesario el uso de una herramienta CASE que permita aumentar la productividad, a su vez, esta empresa vigila la calidad de los desarrollos al reducir el costo en términos de tiempo y dinero. La herramienta CASE integra todo el proceso de desarrollo desde el análisis hasta la implementación.

El proceso de desarrollo del sistema deberá iniciar en el 2017; la tabla 1 detalla las principales características a tomar en cuenta para la estimación del costo del proyecto.

Tabla 1

Esquema de modos de desarrollo de software PEL ® S.A

La cantidad de requisitos	Tamaño	Complejidad	Cantidad de personas requeridas	Experiencia
Se considera un número promedio de requisitos	100 KLDC	Media	Se considera un número promedio de personas	Media

Nota: Elaboración propia.

COCOMO: Un algoritmo de estimación de costo de software

A continuación se muestran los principales conceptos concernientes al modelo constructivo de costos (COCOMO, por su acrónimo del idioma inglés COnstructive COst MOdel) ([Boehm, 1981](#)). COCOMO es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software ([Garita, 2014](#)). En la Figura 1 se muestran los principales conceptos relacionados con el modelo COCOMO. Este modelo, propuesto por Barry W. Boehm, fue introducido a finales de los años 70 y comienzos de los 80 del siglo pasado en su trabajo, Software Engineering Economics ([Boehm, 1981](#)). Entre otras características, el modelo COCOMO está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el “tamaño” del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo.

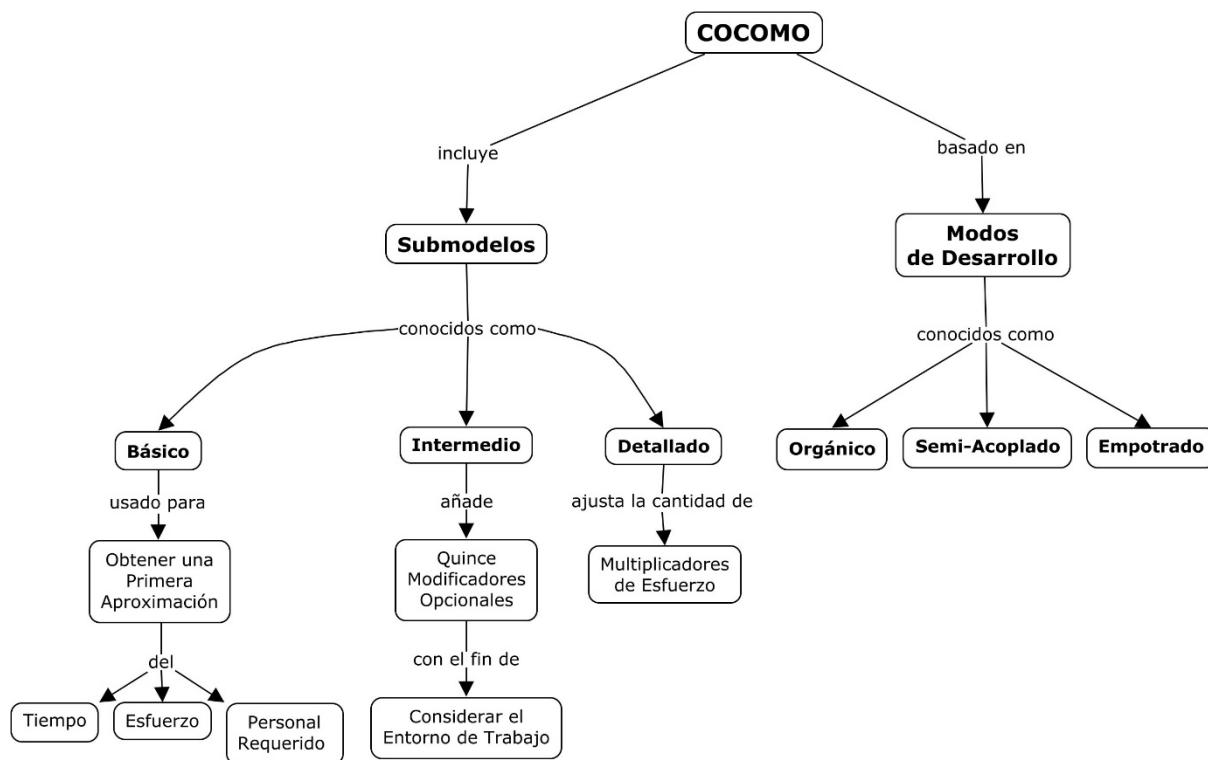


Figura 1. Conceptualización básica de COCOMO. Elaboración propia.

Los submodelos son tres: básico, intermedio y detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado. En la Tabla 2 se muestra el esquema de modos de desarrollo de software con sus principales características que ayudan a elegir el tipo de modo de desarrollo para un proyecto en particular. En esta Tabla 2, en el caso del tamaño, se consideran las líneas de código fuente del software en unidades de miles de líneas de código (KLDC, por sus siglas en inglés).

Tabla 2
Esquema de modos de desarrollo de software

Modo de desarrollo	Requisitos	Tamaño	Complejidad	Personas	Experiencia
Orgánico	Poco rígidos	Pequeño (<50KLDC)	Pequeña	Pocas	Mucha
Semiacoplado	Poco/medio	Medio (50 a 300KLDC)	Medio	Medio	Medio
Empotrado	Alto	Grande (>300KLDC)	Alta	Alta	Poca

Nota: ([Boehm, 1981](#)).

Estos modos de desarrollo permiten utilizar cuatro valores constantes. En la Tabla 3 se muestran los modos de desarrollo y los valores constantes respectivos. Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo.

Tabla 3
Valores constantes por modo de desarrollo

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Nota: ([Boehm, 1981](#)).

Las ecuaciones incluidas en este artículo, son las utilizadas para los submodelos básico e intermedio. Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P). No se incluyen las ecuaciones para el submodelo detallado, por razones de espacio dentro del desarrollo de la propuesta del enfoque pedagógico descrita en este trabajo. En la Tabla 4, se muestran las ecuaciones para esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P) así como los multiplicadores de esfuerzo (ME), utilizados solo en la ecuación de esfuerzo del submodelo intermedio.

Tabla 4
Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Ecuación	Submodelo básico	Submodelo intermedio
Esfuerzo (E)	$(E) = a * (KLDC)b$	$(E) = a * (KLDC)b * ME$
Tiempo (T)	$(T) = c * (E)d$	$(T) = c * (E)d$
Personal (P)	$(P) = E/T$	$(P) = E/T$

Nota: (Boehm, 1981).

Los multiplicadores de esfuerzo, utilizados en la ecuación de esfuerzo del submodelo intermedio, son quince agrupados en cuatro grandes categorías: atributos de producto, atributos de computador, atributos personales y atributos del proyecto. En la Tabla 5 se muestran los multiplicadores de esfuerzo. Cada uno de estos multiplicadores de esfuerzo, tiene una valoración que se clasifica en una escala de 6 valores desde “muy bajo”, “bajo”, “nominal”, “alto”, “muy alto” y “extraordinariamente alto”. Estos multiplicadores de esfuerzo ajustan el valor real del esfuerzo. Dos ejemplos nos ayudan a entender el objeto de los multiplicadores de esfuerzo. Primero, en caso de que no se desee ajustar el esfuerzo nominal, se debe utilizar el valor nominal (es decir, el

valor “1”) para cualquier multiplicador de esfuerzo. Segundo; si, por el contrario, se desea hacer algún ajuste en particular al esfuerzo nominal por razones de, en este caso, poca experiencia en la aplicación, se podría utilizar el multiplicador AEXP en valor nominal “muy bajo” (1,29).

Tabla 5
Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Multiplicadores de esfuerzo (ME)			Valoración					
			Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Atributos del producto								
1.	RELY	Fiabilidad requerida del software	0,75	0,88	1,00	1,15	1,40	
2.	DATA	Tamaño de la base de datos		0,94	1,00	1,08	1,16	
3.	CPLX	Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de la computadora								
4.	TIME	Restricciones del tiempo de ejecución			1,00	1,11	1,30	1,66
5.	STOR	Restricciones del almacenamiento princ.			1,00	1,06	1,21	1,56
6.	VIRT	Inestabilidad de la máquina virtual		0,87	1,00	1,15	1,30	
7.	TURN	Tiempo de respuesta del computador		0,87	1,00	1,07	1,15	
Atributos del personal								
8.	ACAP	Capacidad del analista	1,46	1,19	1,00	0,86	0,71	
9.	AEXP	Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
10.	PCAP	Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	
11.	VEXP	Experiencia en S.O. utilizado	1,21	1,10	1,00	0,90		
12.	LEXP	Experiencia en el lenguaje de progr.	1,14	1,07	1,00	0,95		
Atributos del proyecto								
13.	MODP	Uso de prácticas de programación modernas	1,24	1,10	1,00	0,91	0,82	
14.	TOOL	Uso de herramientas software	1,24	1,10	1,00	0,91	0,83	
15.	SCED	Restricciones en la duración del proy.	1,23	1,08	1,00	1,04	1,10	

Nota: ([Boehm, 1981](#)).

Aplicación

En este apartado se inicia la aplicación del modelo COCOMO basado en las especificaciones del caso de estudio: PEL ® S.A.

Se incluyen a continuación cuatro ejemplos de aplicación del modelo COCOMO. Cada uno de estos ejemplos se basa en el caso de estudio PEL ® S.A.

1. En el primer ejemplo, se desea estimar el costo del proyecto, de forma básica, utilizando el modelo COCOMO básico. El desarrollo de este ejemplo se muestra en la Tabla 6.

Tabla 6
Solución ejemplo #1. COCOMO básico modo semi-acoplado

Ecuación	Submodelo básico	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)b$	$= 3 * (100)^{1.12}$ $= 3 * 173.78$ $= 521,34$
Tiempo (T)	$(T) = c * (E)d$	$= 2.5 * (521.34)^{0.35}$ $= 2.5 * 8,93$ $= 22,33$
Personal (P)	$(P) = E/T$	$= 521.34 / 22.3$ $= 23.34$

Nota: Elaboración propia.

En este ejemplo, el valor de las variables “a”, “b”, “c” y “d” se obtiene de la Tabla 3 COCOMO básico, para un modo de desarrollo semiacoplado, el cual es el modo de desarrollo que responde a las características del caso de estudio PEL ® S.A. para el tamaño, la complejidad, las personas y su respectiva experiencia (que en el caso de estudio es igual a medio en todos los casos). Así las cosas, para el cálculo del tiempo (T), el resultado sería T = 22.3 que indica los meses estimados para desarrollar el proyecto. En cuanto al personal requerido para desarrollar el proyecto (P), la estimación es P = 23. Finalmente, el resultado del esfuerzo (E), que se utiliza para determinar el costo del proyecto, corresponde a 521,34 unidades persona-mes. En el caso de estudio PEL ® S.A. se observa que el salario promedio de cada recurso utilizado en el proyecto es de \$4000. Esto implica que el costo en dólares del proyecto es de \$2.085.360 (dos millones ochenta y cinco mil, trescientos sesenta dólares).

2. En el segundo ejemplo, se desea estimar el costo del proyecto, utilizando el modelo COCOMO intermedio, ajustado con la característica de la *fiabilidad requerida del software* (RELY) con una valoración de “alto”. El desarrollo de este ejemplo se muestra en la Tabla 7.

Tabla 7
Solución ejemplo #2. COCOMO intermedio, modo semiacoplado con ME = RELY “alto”

Ecuación	Submodelo intermedio	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)b * ME$	$= 3 * (100)^{1.12} * 1.15$ $= 3 * 173.78 * 1.15$ $= 599.54$
Tiempo (T)	$(T) = c * (E)d$	$= 2.5 * (599.541)^{0.35}$ $= 2.5 * 9.38$ $= 23.45$
Personal (P)	$(P) = E/T$	$= 599.541 / 23.45$ $= 25.57$

Nota: Elaboración propia.

En este ejemplo, el valor de las variables “a”, “b”, “c” y “d” se obtiene de la tabla 3 COCOMO intermedio, para un modo de desarrollo semiacoplado, el cual es el modo de desarrollo que responde a las características del caso de estudio PEL ® S.A. para el tamaño, la complejidad, las personas y su respectiva experiencia (que en el caso de estudio es igual a medio en todos los casos) se ajustó al esfuerzo (E) con el multiplicador RELY con una valoración de “alto” utilizando el indicador respectivo de 1.15, que se encuentra en la Tabla 5. Así las cosas, para el cálculo del tiempo (T), el resultado sería $T = 23.45$ que indica los meses estimados para desarrollar el proyecto. En el caso del personal requerido para desarrollar el proyecto (P), la estimación es $P = 26$. Finalmente, el resultado del esfuerzo (E), que se utiliza para determinar el costo del proyecto, corresponde a 599,54 unidades persona-mes. En el caso de estudio PEL ® S.A., se observa que el salario promedio de cada uno de los recursos utilizados en el proyecto es de \$4000. Esto implica que, el costo en dólares del proyecto es de \$ 2.398.160 (dos millones trescientos noventa y ocho mil, ciento sesenta dólares).

3. En el tercer ejemplo, se desea estimar del costo del proyecto, utilizando el modelo COCOMO intermedio, ajustado con las características *fiabilidad requerida del software (RELY)* con una valoración de “alto”, y *experiencia en la aplicación (AEXP)* con una valoración de “bajo”. El desarrollo de este ejemplo se muestra en la Tabla 8.

Tabla 8

Solución ejemplo #3. COCOMO intermedio, modo semiacoplado los ME RELY “alto” y AEXP “bajo”

Ecuación	Submodelo intermedio	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)b * ME$	$= 3 * (100)^{1.12} * 1.15 * 1.13$ $= 3 * 173.78 * 1.299$ $= 677.22$
Tiempo (T)	$(T) = c * (E)d$	$= 2.5 * (677.22)^{0.35}$ $= 2.5 * 9.78$ $= 24.45$
Personal (P)	$(P) = E/T$	$= 677.22 / 24.45$ $= 27.69$

Nota: Elaboración propia.

En este ejemplo, el valor de las variables “a”, “b”, “c” y “d” se obtiene de la Tabla 3 COCOMO intermedio, para un modo de desarrollo semiacoplado, el cual es el modo de desarrollo que responde a las características del caso de estudio PEL ® S.A. para el tamaño, la complejidad, las personas y su respectiva experiencia (que en el caso de estudio es igual a medio en todos los casos), se ajustó el esfuerzo (E) con los multiplicadores RELY con una valoración de “alto” utiliza el indicador de 1.15 y AEXP con una valoración de “bajo” utiliza el indicador de 1.13, ambos se encuentran en la Tabla 5. Así las cosas, para el cálculo del tiempo (T), el resultado sería $T = 24.45$ que indica los meses estimados para desarrollar el proyecto. En el caso del personal requerido para desarrollar el proyecto (P), la estimación es $P = 28$. Finalmente, el

resultado del esfuerzo (E), que se utiliza para determinar el costo del proyecto, corresponde a 667,22 unidades persona-mes. En el caso de estudio PEL ® S.A., se observa que el salario promedio de cada uno de los recursos utilizados en el proyecto es de \$4000. Esto implica que, el costo en dólares del proyecto es de \$ 2.708.880 (dos millones setecientos ocho mil, ochocientos ochenta dólares).

4. Para el ejemplo final, consideraremos un ligero ajuste en las principales características del caso de estudio. Las nuevas características sugeridas se muestran en la Tabla 9.

Tabla 9
Nuevas características para la estimación del proyecto

<i>La cantidad de requisitos</i>	<i>Tamaño</i>	<i>Complejidad</i>	<i>Cantidad de personas requeridas</i>
Se considera pocos requisitos	50 KLDC	Baja	Baja

Nota: Elaboración propia.

Una vez considerado este cambio, en este cuarto ejemplo se desea estimar el costo del proyecto utilizando COCOMO intermedio con modo de desarrollo orgánico que es el que correspondería a las nuevas características mostradas en la Tabla 9. Adicionalmente, en este ejemplo se desea ajustar ciertas características adicionales. Primero, se desea considerar la *fiabilidad requerida del software (RELY)* con una valorización de “alto”. Segundo, la *experiencia en la aplicación (AEXP)*, con una valorización de “bajo”. Finalmente, se quiere incluir el *uso de herramientas de software (TOOL)* con una valorización de “bajo”, tomando en cuenta que el ejemplo supone el poco uso de una herramienta CASE. Así las cosas, el desarrollo de este ejemplo se muestra en la Tabla 10.

Tabla 10
Solución ejemplo 4. COCOMO intermedio, modo orgánico, con los ME RELY “alto”, AEXP “bajo” y TOOL “bajo”

Ecuación	Submodelo intermedio	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)b * ME$	$= 3.2 * (50)^{1.05} * 1.15 * 0.94 * 1.10$ $= 3.2 * 60.80 * 1.1891$ $= 231.35$
Tiempo (T)	$(T) = c * (E)d$	$= 2.5 * (231.35)^{0.38}$ $= 2.5 * 7.91$ $= 19.78$
Personal (P)	$(P) = E/T$	$= 231.35 / 19.78$ $= 11.69$

Nota: Elaboración propia.

En este ejemplo, el valor de las variables “a”, “b”, “c” y “d” se obtiene de la Tabla 3 COCOMO intermedio, para un modo de desarrollo orgánico.

Ahora, en el tema del ajuste deseado, de la Tabla 9 obtenemos las nuevas características que permiten identificar, para el cálculo del esfuerzo (E), los ME requeridos. Tal como se indicó en el enunciado del ejemplo, la valoración de los ME utilizados, que se obtienen de la Tabla 5, son:

- RELY, con una valoración de “alto”, utiliza el indicador 1.15.
- AEXP que tiene una valoración de “bajo”, utiliza el indicador 0.94.
- TOOL que tiene una valoración de “bajo”, utiliza el indicador 1.10.

Considerando lo anterior, para el cálculo del tiempo (T), el resultado sería $T = 19.78$ que indica los meses estimados para desarrollar el proyecto. En el caso del personal requerido para desarrollar el proyecto (P), la estimación es $P = 12$. Finalmente, el resultado del esfuerzo (E), que se utiliza para determinar el costo del proyecto, corresponde a 231,32 unidades persona-mes. Finalmente, considerando que el salario promedio de cada uno de los recursos utilizados en el proyecto es de \$4000, el costo en dólares del proyecto es de \$ 925.400 (novecientos veinticinco mil, cuatrocientos dólares).

Ruta de aprendizaje del modelo COCOMO: La propuesta de aplicación pedagógica

1. **Dirigido a:** Personal docente y de tutorías, estudiantes y profesionales en el área de computación, sistemas de información o informática.
2. **Modalidad:** hibrida o virtual.
3. **Tiempo estimado:** 2 -3 horas.
4. **Descripción general**

Propuesta teórica-práctica para orientar el proceso de enseñanza y aprendizaje, el cual pretende mejorar las habilidades de estudiantes en la técnica de estimación de proyectos de software modelo COCOMO, disciplina propia de la ingeniería de software, la cual implica asumir compromisos, interpretar resultados en contraste de las principales características que determinan el gestionar un proyecto: el esfuerzo, el tiempo y los recursos. Entre las principales actividades a realizar están: desarrollar ecuaciones matemáticas para describir las relaciones entre el tamaño del software y la estimación; ajustar el modelo COCOMO con excepción propias del mercado y la interpretación de los resultados obtenidos.

5. Diseño

- a. **Objetivo de aprendizaje:** Aplicar buenas prácticas de ingeniería de software en la estimación de proyectos de software por medio del modelo COCOMO.
- b. **Contenidos:** Modelo COCOMO básico y modelo COCOMO intermedio. (Apartado 3)
- c. **Estrategia:** Tabla 11. Propuesta de aplicación pedagógica de COCOMO.

Tabla 11
Propuesta de aplicación pedagógica de COCOMO

Consigna	Recursos
Cada estudiante debe:	
1. Leer los contenidos que le permitirán conocer el fundamento teórico del modelo COCOMO. Se encuentran en el apartado 3. COCOMO: Un algoritmo de estimación de costo de software.	Ingresar a la plataforma Moodle. Curso activo.
2. Leer el caso apartado 2. El caso de estudio: PEL ® S. A	
3. Resolver el ejercicio 1 ¿Cuál es la estimación del costo del proyecto para el caso de estudio PEL ® S. A, si se utiliza el modelo COCOMO básico?	Uso de tablas 1, 2, 3 y 4
4. Resolver el ejercicio 2 ¿Cuál es la estimación del costo del proyecto para el caso de estudio PEL ® S. A, si se utiliza el modelo COCOMO intermedio y se ajusta la característica fiabilidad requerida del software (RELY) con una valoración de “alto”, de acuerdo con lo indicado por el cliente en este aspecto?	Uso de tablas 1, 2, 3, 4 y 5
5. Resolver el ejercicio 3 ¿Cuál es la estimación del costo del proyecto para el caso de estudio PEL ® S. A, si se utiliza el modelo COCOMO intermedio, y se ajustan las características fiabilidad requerida del software (RELY) con una valoración de “alto”, y experiencia en la aplicación (AEXP) con una valoración de “bajo”?	Uso de tablas 1, 2, 3, 4 y 5
6. Resolver el ejercicio 4 considerando lo siguiente (no utilice el caso de estudio): <ul style="list-style-type: none">• Características<ul style="list-style-type: none">• Cantidad de requisitos: pocos.• Tamaño: 50 KLDC.• Complejidad: baja.• Cantidad de personas: baja• Multiplicadores de esfuerzo (ME)<ul style="list-style-type: none">• Fiabilidad del software (RELY): alta,• Experiencia del equipo de desarrollo (AEXP): baja.• Herramientas de software (TOOL): bajo (se asume un bajo uso de la herramienta CASE),	Uso de tablas 2, 3, 4 y 5
7. Al finalizar los ejercicios puede solicitar los solucionarios.	Tablas 6, 7, 8, 9 y 10

Nota: Elaboración propia.

Análisis

Las razones expuestas por [Boehm et al. \(2000\)](#) para utilizar modelos algorítmicos de estimación del costo en los proyectos de software hacen obligatorio considerar modelos como COCOMO durante el proceso de formación académica de profesionales en ciencias de la computación, ingeniería del software, ingeniería de sistemas, y campos afines. No obstante que el limitado alcance de los modelos algorítmicos de estimación de costos y otros problemas de su aplicación ([Boehm et al., 2000](#); [Kemerer, 1987](#); [Leung y Fan, 2002](#)) hayan llevado a algunos estudios a sugerir que la correcta práctica de estas técnicas pasa por dominar varias técnicas y comparar resultados ([Boehm et al., 2000](#)), el dominar al menos una de esas técnicas permitirá

a los grupos practicantes novatos del desarrollo de software desarrollar una correcta práctica de estimación de costos que incluya algunos elementos formales matemáticos de sustentación de una determinada estimación.

La aplicación correcta del modelo COCOMO hace uso de un número importante de fórmulas y otros aspectos como los multiplicadores de esfuerzo, los cuales hacen difícil de entender el modelo en sí. En adición a esto, el gran número de factores interrelacionados que influyen en el proceso de desarrollo de software de un equipo de desarrollo dado y el gran número de atributos del proyecto reportados por [Leung y Fan \(2002\)](#) podrían hacer cuestionable el uso de estos modelos. Sin embargo, el afán constante de incrementar el desarrollo de software haciendo uso de buenas prácticas de ingeniería que permitan incrementar la calidad del producto final, irremediablemente hacen pensar que es necesario estimar mejor. En este sentido, el aprendizaje del uso del modelo COCOMO proveerá a nuevas generaciones de profesionales de la computación con algunos elementos de argumentación en el manejo de los problemas de ingeniería de software tal como lo sugiere [Boehm \(1981\)](#).

La utilización de un modelo pedagógico, como el presentado en este trabajo, debe permitir, vía una mejor comprensión de los modelos de estimación por parte de las nuevas generaciones de profesionales del desarrollo del software, reducir el alto costo de adopción reportado por [Yang et al. \(2008\)](#). COCOMO se presenta como una alternativa interesante para incrementar la “cultura de estimación” en el desarrollo del software, no solo por el hecho de que es uno de los más conocidos sino porque está en plena evolución ([Sharma, 2013](#)).

La brecha existente en los modelos curriculares comunmente utilizados como base de programas de formación en ciencias de la computación, ingeniería del software, ingeniería de sistemas, y campos afines ([Cassel et al., 2008; Gorgone et al., 2006; LeBlanc et al., 2006; Pyster, 2009; Topi et al., 2010](#)) abre espacio para utilizar modelos pedagógicos como el propuesto en este trabajo.

Conclusiones y trabajos futuros

En este artículo se propone un enfoque pedagógico del modelo algorítmico de estimación de costo de software COCOMO. El objetivo de este trabajo es generar un proceso estándar de enseñanza y aprendizaje del modelo mismo que permita al profesorado enfocar, de forma práctica y basada en un caso real, la enseñanza del tema entre estudiantes. Por su parte, el estudiantado podrá, de forma clara y guiada, abordar los conceptos del modelo para lograr una comprensión adecuada de su aplicación. Finalmente, los grupos profesionales practicantes del desarrollo de software, así como las organizaciones que pretender incluir en sus programas de formación interna el tema de la estimación del costo del software, podrán utilizar este material para comprender y practicar COCOMO.

A futuro, esperamos implementar el modelo y medir sus resultados. Pensamos desarrollar tanto estudios de alcance exploratorio-descriptivo-correlacional ([Hernández, Fernández y Baptista, 2006](#)) como otros longitudinales con el fin de medir el impacto inmediato de la aplicación del modelo y el impacto en el mediano y largo plazo del modelo en la forma de estimar el costo del software. El diseño de estos estudios debería incluir áreas geográficas diversas, tanto nacionales como internacionales, con el fin de explorar si, entre otras cosas, el uso del lenguaje utilizado en el abordaje teórico y los ejemplos incluidos son adecuados para el objetivo final de incrementar la ‘cultura de estimación’.

Referencias

- Basavaraj, M. J., y Shet, K. C. (2008). Empirical validation of Software development effort multipliers of Intermediate COCOMO Model. *Journal of Software*, 3(5), 65-71. Recuperado de <http://www.jsoftware.us/show-60-641-1.html>
- Boehm, B. (1981). *Software engineering economics*. Vol. 197. Recuperado de <http://csse.usc.edu/TECHRPTS/1984/usccse84-500/usccse84-500.pdf>
- Boehm, B., Abts, C., y Chulani, S. (2000). *Software development cost estimation approaches—A survey*. *Annals of software engineering*, 10(1), 177-205. doi: <https://doi.org/10.1023/A:1018991717352>
- Cassel, L., Clements, A., Davies, G., Guzdial, M., McCauley, R., McGetrick, A., Sloan B., Snyder L., Tymann, P., y Weide, B. (2008). *Computer science curriculum 2008: An interim revision of CS 2001*. Recuperado de <https://www.acm.org/education/curricula/ComputerScience2008.pdf>
- Dabbagh, N. (2005). Pedagogical models for E-Learning: A theory-based design framework. *International Journal of Technology in Teaching and Learning*, 1(1), 25-44. Recuperado de <http://sicet.org/web/journals/ijttl/issue0501/DabbaghVol1.Iss1.pp25-44.pdf>
- Doering, A., y Veletsianos, G. (2008). Hybrid online education: Identifying integration models using adventure learning. *Journal of Research on Technology in Education*, 41(1), 23-41. doi: <https://doi.org/10.1080/15391523.2008.10782521>
- Garita, G. (2014). *Métodos analíticos y métricas de calidad del software*. San José. Costa Rica: Editorial EUNED.
- Gorgone, J., Gray, P., Stohr, A., Valacich, J., y Wigand, R. (2006). MSIS 2006: model curriculum and guidelines for graduate degree programs in information systems. *ACM SIGCSE Bulletin*, 38(2), 121-196. doi: <https://doi.org/10.1145/1138403.1138448>
- Hernández, R., Fernández-Collado, P., Baptista, L. (2006). *Metodología de la investigación*. México: Mc. Graw Hill.
- Kemerer, C. (1987). An empirical validation of software cost estimation models. *Communications of the ACM*, 30(5), 416-429. doi: <https://doi.org/10.1145/22899.22906>
- LeBlanc, J., Sobel, A., Diaz-Herrera, L., y Hilburn, B. (2006). *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. Miami: IEEE Computer Society.
- Leung, H., & Fan, Z. (2012). Software Cost Estimation. In *Handbook of Software Engineering and Knowledge Engineering* (pp. 307-324). World Scientific Publishing Company. Recuperado de https://doi.org/10.1142/9789812389701_0014
- Lewis, J. (2001). Limits to software estimation. *ACM SIGSOFT Software Engineering Notes*, 26(4), 54-59. doi: <https://doi.org/10.1145/505482.505490>
- McDonald, J. (2001). Why is software project management difficult? And what that implies for teaching software project management. *Computer Science Education*, 11(1), 55-71. doi: <https://doi.org/10.1076/csed.11.1.55.3845>
- Mehmood, Z., y Saeed, S. (2009). *Teaching quality assurance and project management to undergraduate computing students in Pakistan*. Recuperado de http://www.academia.edu/19786065/Teaching_Quality_Assurance_and_Project_Management_to_Undergraduate_Computing_Students_In_Pakistan

- Pyster, Art. (2009). *Graduate Software Engineering 2009 (GSwE2009) Curriculum Guidelines for Graduate Degree Programs in Software Engineering*. Stevens Institute of Technology. Recuperado de <https://www.acm.org/binaries/content/assets/education/gsew2009.pdf>
- Sharma, H. K. (2013). E-COCOMO: The Extended COSt Constructive MOdel for Cleanroom Software Engineering. *Database Systems Journal*, 4(4). 3-11. Recuperado de http://www.dbjournal.ro/archive/14/14_1.pdf
- Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker, J., Sipior, J., de Vreede, G. (2010). *IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*. Association for Computing Machinery (ACM) & Association for Information Systems (AIS). Recuperado de <https://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf>
- Yang, D., Wang, Q., Li, M., Yang, Y., Ye, K. y Du, J. (2008). A survey on software cost estimation in the chinese software industry. *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM. doi: <https://doi.org/10.1145/1414004.1414045>



Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO (Gabriela Garita-González y Fulvio Lizano-Madriz) por [Revista Uniciencia](#) se encuentra bajo una [Licencia CreativeCommons Atribución-NoComercial-SinDerivadas 3.0 Unported](#).