

# CUANTIFICACIÓN ECONÓMICA DEL APORTE DE LA UNIVERSIDAD HACIA LA SOCIEDAD MEDIANTE EL DESARROLLO DE SOFTWARE

## ECONOMIC QUANTIFICATION FROM THE CONTRIBUTION OF THE UNIVERSITY TO THE SOCIETY THROUGH SOFTWARE DEVELOPMENT

María Nerexi Loor Zamora, Marlon Navia Mendoza, Gustavo Molina Garzón

Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, Campus Politécnico, km 2.7 vía  
Calceta – El Morro – El Limón, Manabí, Ecuador

Email: mnlz16loor1994@hotmail.com

---

### Información del artículo

*Tipo de artículo:*  
Artículo original

*Recibido:*  
11/01/2019

*Aceptado:*  
22/05/2019

*Licencia:*  
CC BY-NC-SA 4.0

*Revista*  
*ESPAMCIENCIA*  
10(1):8-13

### Resumen

La estimación del costo del software es el proceso de cuantificar el esfuerzo requerido para desarrollar un sistema de software. Hay varios métodos y modelos para estimar este costo, y los principales factores que toman en cuenta son el tamaño del software, las interfaces, el tiempo de desarrollo, entre otros. Este trabajo presenta una estimación, desde el punto de vista económico, del aporte del software elaborado en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López (ESPAM MFL) hacia la sociedad. Para esto, se analizaron los sistemas computacionales desarrollados por los estudiantes de la Carrera de Informática durante cinco años (del 2012 al 2016), basados en la operatividad y cuantificación económica. Como metodología, se aplicaron los modelos COCOMO II y COSYSMO, de forma posterior al desarrollo; aunque con limitaciones en la obtención de los datos para la estimación. Mediante investigación de campo se determinó que el 33,6% del software desarrollado fue instalado. Aplicando los modelos mencionados, se estimó un valor de entre USD \$344 033,00 y \$504 194,00 que comprende el software instalado, como beneficio entregado a la sociedad a través de su desarrollo; y entre USD \$1 023 907,00 y \$1 500 578,00 el valor estimado de todo el software desarrollado.

*Palabras clave:* cuantificación, estimación de software, COCOMO II, COSYSMO.

### Abstract

The valuation of software consists of the process of quantifying the effort involved to develop a software system. There are several methods and models to estimate this cost, and the main factors to take into consideration are the software size, interfaces, developing time, among others. This study shows an estimation from the economical point of view of the contribution of the software developed at the Agricultural Polytechnic of Manabí (ESPAM MFL) toward the society. For the purpose, the computer systems developed by the students at the computing school during five years were analyzed (from 2012 to 2016), based on the operability and the economic quantification. As a methodology, models COCOMO II and COSYSMO were carried out, after the development; although with limitations in the collecting of data for the valuation. By means of the field research it was determined that 33,6% of the software developed was installed. Carrying out the models above mentioned, it was estimated that about USD \$344 033,00 and \$504 194,00 with regard to the installed software, as a benefit handed to the society through its development; and between USD \$1 023 907,00 and \$1 500 578,00 the estimated cost of all the software developed.

*Keywords:* Quantification, software estimation, COCOMO II, COSYSMO.

## INTRODUCCIÓN

Uno de los objetivos de toda universidad es aportar a su entorno social, principalmente a través de la transferencia de conocimientos y/o tecnología, tal como lo resalta la Ley Orgánica de Educación Superior del Ecuador (Registro Oficial 298, 2018). Los proyectos con los que se aporta a la sociedad pueden ser realizados por profesores, estudiantes, o equipos conformados por ambos. En Ecuador, este aporte no era tan visible hasta hace unos pocos años, no porque no se lo hiciera, sino porque no se lo evidenciaba o cuantificaba.

En la ESPAM MFL los proyectos de año son trabajos que hacen los estudiantes cada dos semestres, y consisten básicamente en aplicar los conocimientos adquiridos durante un año académico en la resolución de un problema, utilizando un tipo concreto de herramientas o conocimientos. Por otra parte, los trabajos de titulación son el requisito para obtener el título universitario de grado, realizando un trabajo de forma más profesional. En ambos casos se puede obtener como producto un software. Además, la ESPAM MFL cuenta con un programa extracurricular denominado Semillero de Investigadores, a través del cual los estudiantes adquieren información sobre el proceso de investigación, y al final generan y ejecutan un proyecto (ESPAM MFL, 2012). Si bien todos estos proyectos contemplan un presupuesto de materiales y equipos para su ejecución, prácticamente ninguno establece el costo que debería tener el software elaborado.

Con el fin de valorar el costo del software desarrollado por los estudiantes de la Carrera de Informática-Computación de la ESPAM-MFL, se aplicaron dos métodos de estimación. Esta estimación se ha realizado principalmente sobre los proyectos de curso y trabajos de titulación. Dentro de esta valoración, se ha considerado el porcentaje de proyectos entregado y el total desarrollado. Además, este trabajo puede servir de referencia para realizar estimaciones similares en otras universidades.

En el área de las tecnologías de la información, la estimación de los costos de desarrollo de software es un campo que se viene estudiando desde hace poco más de dos décadas, tal como lo demuestran trabajos como el de Briand *et al.* (1998) o de Briand *et al.* (1999). Estos modelos utilizan distintas variables para estimar el costo del software, como pueden ser el tamaño del software, la funcionalidad, la cantidad de interfaces, algoritmos o puntos de función, características de la organización a quien está dirigido, la experiencia en proyectos previos, e incluso el criterio de expertos; dando como resultado valores económicos o de esfuerzo en términos de personas-mes (Jones, 2007). Estos métodos pueden utilizar algoritmos, que emplean como entrada varios de los criterios antes mencionados; o no utilizarlos, sino funcionar con base en la comparación analítica e

inferencia, aplicando incluso mecanismos de inteligencia artificial en su funcionamiento (Khatibi y Jawawi, 2010).

Existen muchos métodos de estimación de costo de software. Como ejemplo se puede citar los propuestos por Briand *et al.* (1998); Zia *et al.* (2011); Hihn *et al.* (2015); Gharehchopogh y Pourali (2015) o Saif (2016); por mencionar algunos. Una de las principales dificultades en la aplicación de estos métodos de estimación es que no todos los proyectos de software son iguales, por lo que los criterios utilizados en la estimación de costos no necesariamente van a ser equivalentes, e incluso se podrían aplicar diferentes modelos de acuerdo al tipo de proyecto o metodología de desarrollo utilizada (Boehm, 2017). Por otra parte, el estimar los costos al inicio de un proyecto puede dar una buena aproximación a los costos reales, pero no toma en cuenta los posibles imprevistos que se den en el camino. En cualquier caso, aunque estos métodos no sean del todo exactos, tal como lo indican Kumari y Pushkar (2013), son útiles al momento de estimar con cierta precisión los costos relacionados al desarrollo de software.

Los métodos de estimación tomados en cuenta para este trabajo fueron el Modelo Constructivo de Costo II (*COConstructive COSt MOdel II*, denominado desde aquí como COCOMO II por sus siglas en inglés) y el Modelo Constructivo de Costo para Ingeniería de Sistemas (*COConstructive Systems Engineering Cost Model*, llamado COSYSMO por su nombre en inglés). El primero, COCOMO II, es la segunda versión un modelo que tiene ya varios años, y que entre sus métricas contempla características del producto (tamaño de código y base de datos, complejidad, reutilización de código, documentación, entre otros), del personal (capacidad y experiencia del personal en el tipo de aplicación y la plataforma utilizada, continuidad del mismo), de la plataforma, del proyecto, y el costo por persona-mes para el desarrollo (Boehm *et al.*, 2000). Aunque no es un modelo relativamente nuevo o novedoso, se sigue utilizando como referencia al momento de hacer estimaciones o comparar modelos.

Por otra parte, COSYSMO es un método posterior con un enfoque más completo. Este modelo contiene dieciocho parámetros: cuatro relacionados con el tamaño del software y catorce relacionados con el esfuerzo. Estos parámetros se agrupan respecto al tamaño del sistema (requisitos, interfaces, algoritmos, escenarios operativos), y métricas de costo (comprensión de la arquitectura, migración, documentación, capacidad y experiencia del equipo, entre otros); así como el costo por persona-mes (Valerdi, 2005). COSYSMO utiliza un marco de trabajo similar al de COCOMO II, pero integra estándares aceptados por la industria.

Por lo tanto, basados en estos antecedentes se tuvo como objetivo cuantificar económicamente el aporte de la universidad hacia la sociedad mediante el desarrollo de software.

## MATERIALES Y MÉTODOS

El tipo de investigación estuvo dividida en dos partes: la primera de campo, y la segunda de tipo bibliográfica no experimental. Tipo de estudio: transversal, descriptivo exploratorio. Técnicas de recolección: encuestas, aplicación de modelos de estimación de costo de software.

Para la primera parte de la investigación (determinar el porcentaje de software entregado en relación al total desarrollado) se tomó una muestra completamente aleatoria de los trabajos realizados. Entre el año 2012 y el 2016 se encontró que en la carrera de computación de la ESPAM MFL se habían ejecutado 94 trabajos de año, 54 trabajos de titulación (Tesis de grado), y 5 proyectos del Programa Semillero de Investigadores; todos estos con la entrega de un producto final de software. Esto da un total de 153 proyectos desarrollados. Para determinar la muestra se utilizó la ecuación [1], teniendo  $N$  (población) con un valor de 153,  $Z$  (confianza) al 95% tiene un valor de 1,96,  $p$  (probabilidad de ser seleccionado) un 0,5, una  $q$  (probabilidad de fracaso) de 0,5, y el valor de  $d$  (error esperado) al 0.05. Estos parámetros dan un tamaño de la muestra  $n$  de 110.

$$n = \frac{N * Z_{\alpha}^2 * p * q}{d^2 * (N - 1) + Z_{\alpha}^2 * p * q} \quad [1]$$

Una vez determinada la muestra, y previo a la aplicación de los modelos de estimación de costo, se realizó una encuesta para determinar la situación del software desarrollado. Mediante esta encuesta se determinó si el software fue entregado e instalado, o si solo se quedó en desarrollo. Estos valores sirven para estimar la transferencia real y el total desarrollado, independientemente de que se haya entregado o no.

Se buscó información sobre el software elaborado en los proyectos realizados, para estimar su costo con base en los modelos mencionados. Considerando la similitud en la formación de los estudiantes y su experiencia, se configuraron los parámetros de estimación de forma similar para todos los sistemas y aplicaciones desarrollados. El cuadro 1 muestra estos parámetros de estimación. El costo persona-mes se estableció como el promedio durante el tiempo analizado de una remuneración a un estudiante que está realizando pasantías, esto es la mitad del salario básico unificado (SBU) más los costos de afiliación a la seguridad social, de acuerdo a las leyes ecuatorianas.

**Cuadro 1.** Configuración de parámetros de estimación

Parámetro	COCOMO II	COSYSMO
Dimensionamiento o del tamaño del software	Líneas de código	Número requerimientos, de interfaces y de algoritmos
Controladores de escala	Nominal	n/a
Aspectos del sistema	Bajo	Bajo
Aspectos de Personal	Nominal	Nominal
Aspectos de Plataforma	Nominal	Nominal
Aspectos de Proyecto	Nominal	Nominal
Mantenimiento	No	No

Se realizó la estimación del software desarrollado en 10 proyectos distintos, puesto que, de los trabajos correspondientes al tiempo del estudio, fueron los únicos de los que se pudo encontrar el código fuente para poder aplicar COCOMO II. Para realizar la estimación se utilizó una herramienta web que dispone de varios modelos (Madachy, n.d.), en donde se pueden configurar los parámetros de cada uno de ellos. En el caso del modelo COCOMO II, el parámetro individual de cada proyecto, relacionado al tamaño del mismo, fue el número de líneas de código. Para obtener el valor de este parámetro se utilizó la aplicación Cloc versión 1,70 para Windows 7 (Durán, 2016). En cambio para COSYSMO la diferencia para cada desarrollo fueron el número de requisitos, número de interfaces y de algoritmos, todos estos en nivel nominal. Esta información se obtuvo mediante una búsqueda bibliográfica en los mencionados proyectos.

Una vez estimado el costo de cada proyecto, se determinó la media aritmética del costo de los mismos, y con este valor se realizó una inferencia por extrapolación lineal para determinar tanto el aporte real (el software entregado) como el total desarrollado en estos trabajos. Este procedimiento se hizo para ambos modelos de estimación.

Las limitaciones de este estudio son las siguientes: solo se pudo obtener el código fuente (necesario para aplicar COCOMO II) de 10 de los proyectos de software desarrollados, lo que puede causar inexactitud en la estimación del costo total. En muchos casos solo le había dejado el programa de software ejecutable, y en algunos ni eso. Por otra parte, no fue posible comprobar si el código facilitado efectivamente se correspondía con la funcionalidad declarada en los informes de los trabajos realizados, lo que podría causar inconsistencia en los resultados obtenidos. Además, solo se han considerado dos modelos de estimación, uno basado en el tamaño en líneas de código, y otro basado en puntos de función.

## RESULTADOS Y DISCUSIÓN

Después de realizar la encuesta a la muestra de las organizaciones para las que se desarrolló software, se procesaron los datos obtenidos. El cuadro 2 muestra la cantidad de software que se instaló y entregó, así como la cantidad y porcentaje que no fue implementado. Apenas una tercera parte (33,6%) del software desarrollado pudo ser instalado y cumplir con el objetivo para el que se desarrolló. El restante 66,4% no fue instalado por diferentes motivos, lo cual no compete al presente trabajo.

**Cuadro 2:** Número de software instalado y no instalado

	Frecuencia	Porcentaje
Se instaló el software desarrollado	37	33,6%
No se instaló el software	73	66,4%
Total	110	100,0%

Una vez aplicados los modelos mediante la herramienta antes indicada, se obtuvo el costo de 10 sistemas seleccionados. En el cuadro 3 se muestran los costos obtenidos aplicando COCOMO II, así como el costo promedio. Para el cálculo se utilizó el código escrito, procurando no tomar en cuenta el código generado automáticamente por los entornos de desarrollo. Se puede apreciar que el costo estimado más bajo entre los trabajos analizados, según el modelo COCOMO II, fue de USD \$1349; mientras que el más alto fue de USD \$19696, con una relación de casi 10 a 1 entre ambos valores.

**Cuadro 3.** Costo de software desarrollado de acuerdo a COCOMO II

No.	Líneas de código	Costo estimado (USD)
1	3882	1349
2	30585	13143
3	5473	1970
4	27478	11679
5	28121	11980
6	19393	7952
7	44136	19696
8	31274	13470
9	14098	5594
10	26548	11244
Costo medio		<b>9808</b>

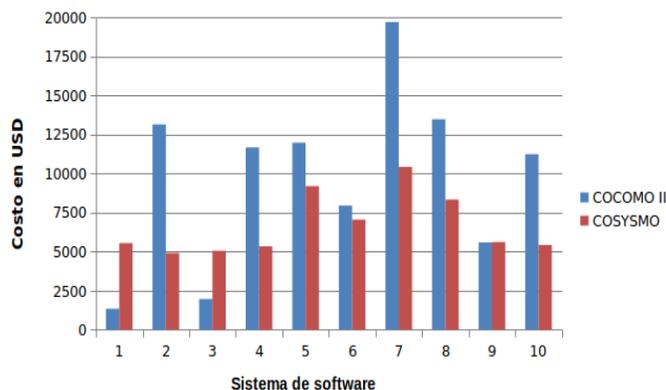
De la misma forma, con los 10 sistemas seleccionados, se estimó su costo aplicando COSYSMO. El cuadro 4 muestra estos valores, incluyendo el costo promedio. Aquí el costo estimado de software más bajo fue de USD \$4917; mientras que el más alto fue de USD \$10437. A diferencia del modelo anterior, en este se puede observar que la variación de costos es menor, aunque el promedio también es más bajo.

**Cuadro 4.** Costo de software desarrollado de acuerdo a COSYSMO

No.	No. de Requisitos	No. de Interfaces	Número de Algoritmos	Costo Estimado (USD)
1	18	12	16	5548
2	12	10	15	4917
3	14	12	14	5058
4	14	13	15	5336
5	36	14	32	9197
6	27	22	16	7049
7	30	18	38	10437
8	15	17	30	8337
9	15	12	16	5616
10	21	13	15	5427
Costo medio				<b>6692</b>

Se puede observar que el costo estimado varía de acuerdo al modelo aplicado. En la figura 1 se aprecia mejor esta variabilidad. En la mayoría de los casos la tendencia del costo de cada sistema es diferente en cada modelo, y en apenas dos de los casos analizados el costo es aproximadamente el mismo.

Según lo establecido en el cuadro 2, el 33,6% de los sistemas de software fueron entregados. Por lo tanto, teniendo un total de 153 proyectos, se deduce que aproximadamente 51 de estos terminaron en un producto entregado. Con esta cifra, y la media del costo de los Cuadros 3 y 4, en el cuadro 5 se muestra un estimado del costo total del software entregado, así como del costo total desarrollado, que equivaldría al potencial aporte de la institución si se hubieran entregado todos los sistemas elaborados.



**Figura 1.** Costo de cada sistema de acuerdo al modelo aplicado

**Cuadro 5:** Estimación de aporte a través del software.

	COCOMO II	COSYSMO
Costo software entregado	504 194	344 033
Costo total de software desarrollado	1 500 578	1 023 907

Con base en estos valores se puede estimar la cuantificación del aporte del software, durante los cinco años del análisis, de entre USD \$344 033,00 (el menor valor obtenido) y \$504 194,00 (el mayor valor). De igual forma, el costo estimado del total de software desarrollado estaría entre USD \$1 025 907,00 y \$1 500 578,00.

Para una adecuada transferencia tecnológica o de conocimientos, es necesario que todos los involucrados (investigadores, administradores, académicos) entiendan las necesidades del entorno, y tengan la preparación adecuada. Es una oportunidad para aprovechar el potencial científico disponible en las instituciones de educación superior (Wang y Cao, 2010).

En la práctica, la forma de hacerlo no siempre es la adecuada, y en la mayoría de los casos no se llega a efectuar dicha transferencia, tal como se ha analizado en este estudio, donde se encontró que apenas una tercera parte de los sistemas de software desarrollados fueron entregados. En el estudio hecho por Diebold *et al.* (2015) en Alemania, se encontró varios factores que limitan una adecuada transferencia tecnológica, como por ejemplo no entender bien las necesidades de la industria, o que no se encuentran definidos los procesos para realizar dicha transferencia.

Por lo tanto, se hace evidente la necesidad de un marco de trabajo, que incluya políticas y procedimientos claros, para facilitar la transferencia de tecnología a nivel de sistemas de software por parte de la universidad (Hallam *et al.*, 2014). En este sentido, la propuesta de Schuh y Aghassi (2013), un modelo de portal para transferencia de tecnología, puede servir de referencia en la propuesta de soluciones para la situación analizada.

Por otra parte, la aplicación de los modelos de estimación debe hacerse de forma adecuada, tomando en cuenta la naturaleza de cada proyecto. Una de las posibles causas de la diferencia en los costos encontrados en cada modelo sea justamente que solo se tomó en cuenta indicadores generales (líneas de código, número de interfaces o de algoritmos), sin tomar en consideración aspectos propios de cada proyecto. Es así que para una mejor estimación se podría tomar en cuenta el tipo de proyecto y/o método de desarrollo empleado, para decidir qué modelo utilizar (Boehm, 2017). Además, no se encontraron estudios que indiquen qué modelos de estimación puedan ser los más adecuados para los trabajos de software realizados por estudiantes. Como trabajo a futuro se podría analizar qué

modelo de estimación de costo es el más adecuado para aplicar en este caso.

## CONCLUSIONES

En este documento se presentó la aplicación de dos modelos de estimación de costo de software, posterior al desarrollo de los mismos, para determinar el monto aproximado de los sistemas elaborados por estudiantes de la ESPAM MFL en el lapso de 5 años. Los modelos aplicados dieron como resultado costos diferentes para casi todos los trabajos analizados. A pesar de esto, son un buen método para determinar, de forma aproximada, costos de proyectos de software.

La variación en el costo estimado de creación de un software puede deberse a varios factores. Uno de ellos es la diferencia del tipo de métrica que cada modelo utiliza (Cuadro 2). Debido a que no todos los trabajos de software utilizados para la estimación son similares en su naturaleza, no es posible determinar una equivalencia en número de líneas de código con los parámetros de COSYSMO (número requerimientos, de interfaces y de algoritmos). Mientras que con COCOMO II se muestra una mayor variabilidad en los costos estimados, con COSYSMO la variación es relativamente menor (Figura 1). Al respecto, una futura investigación en esta vía sería aplicar varios métodos a posteriori a los programas y sistemas elaborados, para comparar los resultados obtenidos, y determinar el o los que mejor se ajusten a los desarrollos de tipo estudiantil.

El costo estimado del software desarrollado y entregado en el lapso de tiempo analizado (5 años) está entre un millón y un millón y medio de dólares, dependiendo del modelo de estimación aplicado. Es decir, que en promedio al año se ha elaborado aproximadamente entre USD \$200 000 y \$300 000 en software.

Se encontró que apenas una tercera parte del software desarrollado en la universidad suele entregarse a la organización beneficiada. De acuerdo a investigaciones realizadas por otros autores, son varios los factores que pueden incidir en este problema. Por lo tanto, es necesario establecer estrategias y procedimientos para que la transferencia tecnológica sea más significativa.

Aunque no es un requisito al momento de que los estudiantes elaboren software, el realizar la estimación de su costo daría un valor agregado a su trabajo. El modelo o método que apliquen en su estimación debe estar fundamentado y justificado, preferiblemente con un estudio como el que se propone en la sección anterior.

## LITERATURA CITADA

- Boehm, B., Clark, B., Devnani-Chulani, S., Horowitz, E., Madachy, R., Reifer, D., ... Steece, B. 2000. COCOMO II Model Definition Manual. University of Southern California (2.1). Center for Software

- Engineering, USC.  
<http://doi.org/10.1525/nr.2000.4.1.6>
- Boehm, B. W. 2017. Software cost estimation meets software diversity. Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017, 495–496. <http://doi.org/10.1109/ICSE-C.2017.159>
- Briand, L. C., El Emam, K., & Bomarius, F. 1998. COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. In Proceedings of the 20th International Conference on Software Engineering, 24: 90-399. IEEE Comput. Soc. <http://doi.org/10.1109/ICSE.1998.671392>
- Briand, L. C., El Emam, K., Surmann, D., Wiczorek, I., & Maxwell, K. D. 1999. An assessment and comparison of common software cost estimation modeling techniques. In Proceedings of the 21st international conference on Software engineering - ICSE '99 (pp. 313–322). New York, New York, USA: ACM Press. <http://doi.org/10.1145/302405.302647>
- Diebold, P., Vetro, A., & Mendez F., D. 2015. An Exploratory Study on Technology Transfer in Software Engineering. In 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) 2015:1–10. IEEE. <http://doi.org/10.1109/ESEM.2015.7321189>
- Durán, J. 2016. CLOC la herramienta para contar líneas de código. Retrieved December 15, 2017, from <https://www.somosbinarios.es/cloc-herramienta-contar-lineas-codigo/>
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López). 2012. Manual del Sistema de Investigación (2da ed.).
- Gharehchopogh, F. S., & Pourali, A. 2015. A new approach based on continuous genetic algorithm in software cost estimation. Journal of Scientific Research and Development, 2(4):87–94.
- Hallam, C., Wurth, B., & Flannery, W. 2014. Understanding the system dynamics of the university-industry technology transfer process and the potential for adverse policy creep. In PICMET 2014 - Portland International Center for Management of Engineering and Technology, Proceedings: Infrastructure and Service Integration (pp. 1129–1136). Kanazawa, Japan: IEEE. Retrieved from <https://ieeexplore.ieee.org/abstract/document/692128>
- Hihn, J., Juster, L., Menzies, T., Mathew, G., & Johnson, J. 2015. NASA software cost estimation model: an analogy based estimation model. In 2015 ICEAA Professional Development & Training Workshop. San Diego, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration. Retrieved from <https://trs.jpl.nasa.gov/handle/2014/45724>
- Jones, C. 2007. Estimating Software Costs: Bringing Realism to Estimating (Second Ed.). New York, NY (US): McGraw-Hill.
- Khatibi, V., & Jawawi, D. N. A. 2010. Software Cost Estimation Methods: A Review. Journal of Emerging Trends in Computing and Information Sciences, 2(1): 21–29.
- Kumari, S., & Pushkar, S. 2013. Comparison and Analysis of Different Software Cost Estimation Methods. International Journal of Advanced Computer Science and Applications, 4(1):153–157.
- Madachy, R. (n.d.). Suite of Constructive Cost Models. Retrieved December 10, 2017, from <http://csse.usc.edu/tools/COCOMOSuite.php>
- Registro Oficial 298. Ley Orgánica de Educación Superior. 2018. Ecuador.
- Saif, H. 2016. Software Cost Estimation. Global Sci-Tech, 8(1):15–21.
- Schuh, G., & Aghassi, S. 2013. Technology transfer portals: A design model for supporting technology transfer via social software solutions. In 2013 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 43–47). IEEE. <http://doi.org/10.1109/IEEM.2013.6962371>
- Valerdi, R. 2005. The Costructive Systems Engineering Cost Model (COSYSMO). University of Southern California.
- Wang, J., & Cao, H. R. 2010. Improve the university technology transfer: Factors and framework. In 2010 Second International Conference on Communication Systems, Networks and Applications, 2:216–219.
- Zia, Z., Rashid, a., & uz Zaman, K. 2011. Software cost estimation for component-based fourth-generation-language software applications. IET Software, 5(1):103