

Tipo de artículo: Artículo original
Temática: Ingeniería y Gestión de Software
Recibido: 21/04/2015 | Aceptado: 04/09/2015

Generalización del Método Puntos de Función utilizando lógica difusa

Generalization of Function Points method using fuzzy logic

Liannis Soria Barreda ^{1*}, Manuel Ramón Almaguer Ochoa ¹, Julio Cesar Diaz Vera ¹, Lizandra Arza Perez ¹

¹ Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, km 2 ½ Torrens, Boyeros, La Habana, Cuba. C.P.: 19370.

* Autor para correspondencia: lsoria@uci.cu

Resumen

En los últimos años se ha incrementado la utilización del método Puntos de Función para la estimación del software. A pesar de los buenos resultados que se alcanzan con este método, aún existen dificultades que afectan la exactitud de las estimaciones, dentro de estas dificultades se destacan el "arranque en frío" y los "límites bruscos". En este trabajo se propone una generalización del método Puntos de Función mediante el uso de la lógica difusa que contribuye a atenuar el efecto de los "límites bruscos" y al mismo tiempo, es capaz de operar con eficacia en situaciones en las que no existen resultados históricos en la organización sobre desarrollos anteriores, por lo que no se ve afectado este proceso por el "arranque en frío". La efectividad de la propuesta es constatada mediante un experimento en el cual se obtiene la estimación utilizando el método de Puntos de Función clásico y la generalización propuesta en un caso de estudio. Los resultados alcanzados con la generalización mostraron ser más cercanos a la realidad que las estimaciones realizadas con el enfoque clásico.

Palabras clave: puntos de función, lógica difusa, límites bruscos, arranque en frío

Abstract

In the last years it has increased the use of Function Point method for estimating software. Despite the good results achieved with this method, there are difficulties that affect the accuracy of the estimates, among these difficulties highlight the "cold start" and "rough limits". In this paper a generalization of the method Function Point is proposed by using fuzzy logic, which helps to mitigate the effect of "rough limits" and at the same time is able to operate effectively in situations where the organization don't have historic results about previous developments, so it is not affected this process by the "cold start". The effectiveness of the proposal is proven by an experiment in which the estimate is

obtained using the classical method Function Points and the generalization proposed in a case study. The results achieved with the generalization showed to be closer to reality than estimates using the classical approach.

Keywords: *function point, fuzzy logic, rough limits, cold start*

Introducción

La estimación del tamaño del software resulta vital para los proyectos de software a la hora de planificar el esfuerzo y el costo asociado a su desarrollo. En la industria de desarrollo de software se utilizan una gran variedad de métodos para estimar; la elección del método de estimación dependerá de las características propias del proyecto a desarrollar. En (CMMI, 2010) ha sido incluido el método de estimación Puntos de Función como ejemplo para la estimación de proyectos de software, lo que ha permitido elevar la reputación de este método de estimación en los últimos años y por consiguiente su popularidad.

En el año 2012 el grupo consultor DCG, publicó un informe sobre la utilización de los métodos utilizados en la industria para medir el software, donde se evidencia el auge alcanzado por el método Puntos de Función en todo el mundo. Según este informe, el método Puntos de Función alcanzó un 34 % de utilización en las compañías encuestadas, mientras que el resto de los métodos de estimación oscilaron entre un 13 % y un 5 % de utilización (DCG, 2013). Además, en el último informe publicado a inicios del 2015 por la Asociación de Auditoría y Control de

Sistemas (ISACA, por sus siglas en inglés), sobre el uso de las métricas en los Estados Unidos, la métrica Puntos de Función logró ubicarse en la cima con un 39 % de utilización, seguido por las líneas de código con un 21 % (Henderson, et al. 2015).

A pesar de la popularidad alcanzada por el método Puntos de Función, todavía se pueden apreciar las siguientes insuficiencias:

- La clasificación de los componentes en simples, medianos y complejos, no refleja con exactitud la complejidad de los datos que son gestionados.
- Límites bruscos en la asignación de la cantidad de puntos de función a un componente que resulta engañosa para el proyecto. Por ejemplo, un archivo lógico interno con 19 elementos de datos y 6 subgrupos de elementos de datos es considerado de complejidad media y le son asignados 7 puntos de función al componente; en cambio a un archivo lógico interno con 20 elementos de datos y 6 subgrupos de elementos de datos es considerado de complejidad alta y le son asignados 10 puntos de función al componente.

La tendencia que existe hoy en el mundo para eliminar los límites bruscos en una situación dada, es la utilización de las denominadas técnicas de *soft computing*. Para el método Puntos de Función, los primeros pasos estuvieron encaminados a modelar probabilísticamente el comportamiento del método en (Yau and Tsoi, 1998) para proponer los valores de las 14 características del sistema que se utilizan para ajustar los puntos de función.

En (Lima, et al. 2003) se propone por primera vez la utilización de los conjuntos difusos para mejorar las estimaciones de este método con la utilización de un sistema que permite calcular los pesos de los componentes utilizando el modelo propuesto; pero en este trabajo se debe señalar que para la definición del modelo se utilizó una pequeña base de datos compuesta por sistemas heredados desarrollados en *Visual Basic* y *Microsoft Access* principalmente, lo que compromete la fiabilidad del modelo creado. En (Xia, et al. 2008) se continúa con esta tendencia y se propone la utilización de un enfoque *neuro-fuzzy*, la combinación de lógica difusa con redes neuronales, para calibrar los valores de los pesos del método utilizando los datos disponibles en el repositorio de proyectos de software del Grupo Internacional de Estándares de Medición de Software (ISBSG, por sus siglas en inglés). En este trabajo se refleja la tendencia de la industria en el desarrollo de software al aprender de los datos históricos del repositorio del ISBSG, lo que puede distorsionar los resultados al aplicarse en un entorno con características propias para el desarrollo de software. En esta propuesta además se mantienen las tres categorías para clasificar la variable relacionada con los elementos de datos, lo que implica continuar estimando con intervalos grandes que provocan límites bruscos.

Para mejorar la estimación del método Puntos de Función, también se he propuesto la combinación de la lógica difusa y los métodos de interpolación en (Chen, et al. 2010), pero en dicha investigación los autores reconocen que los parámetros de la función de membresía y los nodos de interpolación deben ser mejorados debido a que la cantidad de datos históricos utilizados en el estudio fue baja. Además según en el estudio realizado por (Ferreira, et al. 2014) sobre la utilización de técnicas de inteligencia artificial para la estimación del esfuerzo en proyectos de software, se puede concluir que la tendencia en los autores gira en torno a proponer nuevos modelos basados en la utilización de los datos históricos de estimaciones de proyectos anteriores, a los que se les aplica diferentes técnicas de extracción del conocimiento para realizar la estimación del nuevo proyecto a desarrollar.

Según el análisis de los trabajos anteriores y el estudio realizado por (Ferreira, et al. 2014), los modelos propuestos hasta el momento, que cuentan con una validación exitosa al utilizar una técnica de *soft computing* para mejorar las estimaciones utilizando el método Puntos de Función, han utilizado los registros históricos de estimaciones de proyectos anteriores para realizar los cálculos de los valores que se proponen. Por lo que estas propuestas no pueden ser utilizadas en las empresas de desarrollo de software que no tengan registrados una cantidad suficiente de datos históricos sobre

estimaciones anteriores que le permitan realizar un aprendizaje de sus características propias; ya que estas características pueden diferir en gran medida con las tendencias de la industria.

En el presente trabajo se propone generalizar el método Puntos de Función utilizando lógica difusa para eliminar los límites bruscos del método que permitan establecer estimaciones con intervalos más certeros en una empresa de desarrollo de software que no cuente con una base de datos histórica robusta sobre las estimaciones de proyectos ya desarrollados. Para esto se definen para cada una de las variables del método su dominio y las etiquetas lingüísticas que los representan y sus conjuntos borrosos; en el caso de las variables asociadas a los elementos de datos y los puntos de función, se propone extender a cinco las categorías para su clasificación. Luego se procede a aplicar el sistema de inferencia borroso y la concreción del conjunto borroso obtenido en cada una de las variables del método.

Materiales y métodos

En esta sección se enuncian los conceptos utilizados para la generalización realizada al método de estimación Puntos de Función, así como los pasos seguidos para obtener dicha generalización.

Lógica difusa

La lógica borrosa se desarrolló a partir de la teoría básica de los conjuntos borrosos establecidos por Lofti Zadeh en 1965 en la Universidad de Berkeley, donde cada conjunto borroso es un conjunto de elementos cuya pertinencia al mismo es gradual y no absoluta. Para cada conjunto borroso se define una función de pertenencia que indica la medida en que cada elemento forma parte de ese conjunto borroso. Un conjunto borroso A se define mediante notación matemática como $A = \{(x, \mu_A(x)) / x \in U\}$, así mismo su función de pertenencia se denota entonces como $\mu_A(x) \in [0, 1]$. (Maguiña, 2010)

Existen diferentes formas para representar la función de pertenencia de un conjunto borroso. La representación de las funciones de pertenencia de cada una de las etiquetas lingüísticas definidas en este trabajo se realizó a través de las funciones trapezoidales y triangulares, las cuales se muestran en la figura 1.

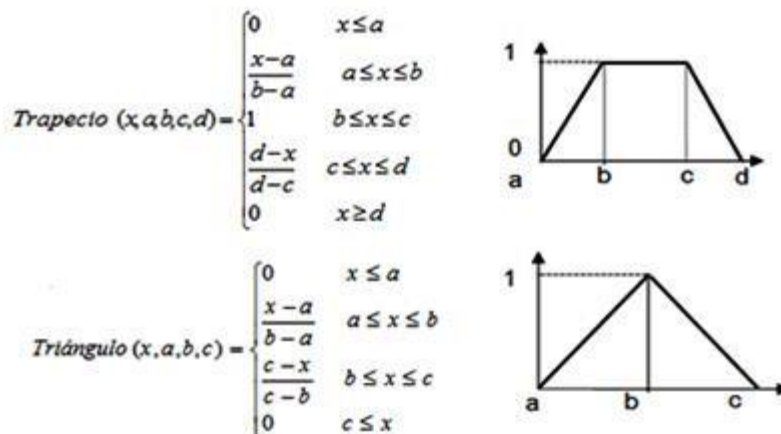


Figura 1. Funciones trapezoidal y triangular. (Piñero, 2005)

Sistemas de Inferencia Borrosos

Los sistemas de inferencia borrosos son los encargados de manipular los conjuntos borrosos mediante las reglas que describen el conocimiento del proceso en cuestión y los términos lingüísticos definidos, utilizando modelos de inferencia como los de Mamdani, Sugeno y Tsukamoto (Ferreira, et al. 2014). Estos sistemas son considerados sistemas expertos con un razonamiento aproximado para mapear un vector de entrada a una salida única basándose en la lógica difusa (Maguiña, 2010).

El modelo propuesto por Sugeno se adapta mejor a los análisis matemáticos y resulta eficiente para términos computacionales, técnicas lineales, de optimización y adaptativas; este modelo es también conocido como Takagi, Takagi-Sugeno, Kang o TSK. En este modelo la conclusión de las reglas borrosas definidas es representada como una función lineal de las variables de entrada. Además de no necesitar un proceso de concreción ya que cada regla tiene un valor exacto de salida a los cuales se les aplica un promedio o suma ponderada para obtener el resultado final.

(Maguiña, 2010)

El modelo propuesto por Tsukamoto es el menos utilizado debido a la poca transparencia que tiene respecto a los otros modelos. En este modelo la conclusión de las reglas borrosas definidas es representada a través de un conjunto borroso con una función de membresía monoatómica. Además de definir un valor exacto para cada regla inducido por la fuerza de activación de la regla, por lo que la salida final del sistema de inferencia está dada por el promedio ponderado de cada salida de las reglas, lo que implica no realizar un proceso de concreción. (Chuen, 1990)

El modelo propuesto por Mamdani ha sido más ampliamente aceptado al ser considerado más intuitivo y adaptable al lenguaje humano, además de poder ser convertido al tipo Sugeno (Jassbi, et al. 2007). Este método propone definir una base de reglas borrosas que contenga el siguiente formato: Si u_1 es A_1 y u_2 es A_2 y...y u_n es A_n entonces v es B , donde u_j y v son etiquetas lingüísticas, y A_j y B son los valores lingüísticos que dichas etiquetas pueden asumir. Para la realización de este trabajo se decidió utilizar este modelo. (Maguiña, 2010)

Método utilizado para generalizar el método Puntos de Función aplicando lógica difusa

Para definir la generalización del método Puntos de Función utilizando un sistema de inferencia borrosa del tipo Mamdani se tuvieron en cuenta los siguientes pasos:

1. Definición de las variables de entrada determinando su dominio y los números difusos que formarán las condicionantes en las reglas a partir de las subdivisiones del dominio que se realicen.

Las variables de entrada a utilizar se corresponden a cada uno de los componentes que se definen en el método Puntos de Función, las cuales son: Archivos internos lógicos (AIL), Archivos de interfaz externos (AIE), Entrada externa (EE), Salida externa (SE) y Consulta externa (CE). (Albrecht, 1979) En la tabla 1 se especifica para cada una de estas variables de entrada definidas, su dominio y las etiquetas lingüísticas que los representan y sus conjuntos borrosos.

Para las variables de entrada AIL y AIE se definieron tres etiquetas lingüísticas: Tipo de Elementos de Datos (TED), Tipo de Elementos de Registros (TER) y Puntos de Función (PF). Para las variables de entrada EE, SE y CE se definieron tres etiquetas lingüísticas: Tipo de Elementos de Datos (TED), Tipo de Archivo Referenciado (TAR) y Puntos de Función (PF). Para la etiqueta lingüística TED se propone su evaluación con los términos Muy Bajo, Bajo, Medio, Alto y Muy Alto debido al amplio rango de valores que pertenecen a dicho conjunto; el resto de las etiquetas lingüísticas se proponer evaluar con los términos Bajo, Medio y Alto según los mismos valores que propone el método Puntos de Función y que varían en cada una de las variables de entrada definidas.

El conjunto TED representa al número de campos únicos que son reconocidos por el usuario. El conjunto TER representa al número de subgrupos de elementos de datos que son reconocidos por el usuario. El conjunto TAR representa al número de AIL o AIE que se han sido leídos y mantenidos en una función transaccional. El conjunto PF representa el peso que se le otorgará para la estimación de acuerdo a la cantidad de TER y TED o TAR y TED de la variable de entrada que se analice. (IFPUG, 2010)

Tabla 1: Etiquetas lingüísticas definidas para las variables de entrada

TED	TER / TAR	PF
Archivos Internos Lógicos		
Muy Bajo: Trapecio (x, nulo, 1, 6, 10) Bajo: Trapecio (x, 3, 8, 12, 20) Medio: Trapecio (x, 10, 20, 36, 43) Alto: Trapecio (x, 28, 38, 44, 54) Muy Alto: Trapecio (x, 43, 50, nulo, nulo)	Bajo: Trapecio (x, nulo, 1, 2, 4) Medio: Trapecio (x, 1, 3, 5, 6) Alto: Trapecio (x, 3, 6, nulo, nulo)	Muy Bajo: Triángulo (x, 0, 1.8, 3.6) Bajo: Triángulo (x, 3.6, 5.4, 7.2) Medio: Triángulo (x, 7.2, 9, 10.8) Alto: Triángulo (x, 10.8, 12.6, 14.4) Muy Alto: Triángulo (x, 14.4, 16.2, 18)
Archivos de Interfaz Externos		
Muy Bajo: Trapecio (x, nulo, 1, 6, 10) Bajo: Trapecio (x, 3, 8, 12, 20) Medio: Trapecio (x, 10, 20, 36, 43) Alto: Trapecio (x, 28, 38, 44, 54) Muy Alto: Trapecio (x, 43, 50, nulo, nulo)	Bajo: Trapecio (x, nulo, 1, 2, 4) Medio: Trapecio (x, 1, 3, 5, 6) Alto: Trapecio (x, 3, 6, nulo, nulo)	Muy Bajo: Triángulo (x, 0, 1.3, 2.6) Bajo: Triángulo (x, 2.6, 3.9, 5.2) Medio: Triángulo (x, 5.2, 6.5, 7.8) Alto: Triángulo (x, 7.8, 9.1, 10.4) Muy Alto: Triángulo (x, 10.4, 11.7, 13)
Entradas Externas		
Muy Bajo: Trapecio (x, nulo, 1, 3, 7) Bajo: Trapecio (x, 1, 5, 7, 11) Medio: Trapecio (x, 6, 9, 11, 16) Alto: Trapecio (x, 10, 14, 18, 22) Muy Alto: Trapecio (x, 16, 22, nulo, nulo)	Bajo: Trapecio (x, nulo, 0, 1, 2) Medio: Trapecio (x, 0, 2, 3, 4) Alto: Trapecio (x, 1, 3, nulo, nulo)	Muy Bajo: Triángulo (x, 0, 0.8, 1.6) Bajo: Triángulo (x, 1.6, 2.4, 3.2) Medio: Triángulo (x, 3.2, 4, 4.8) Alto: Triángulo (x, 4.8, 5.6, 6.4) Muy Alto: Triángulo (x, 6.4, 7.2, 8)
Salidas Externas		
Muy Bajo: Trapecio (x, nulo, 1, 3, 7) Bajo: Bajo: Trapecio (x, 1, 5, 7, 9) Medio: Medio: Trapecio (x, 6, 9, 16, 20) Alto: Alto: Trapecio (x, 10, 16, 21, 28) Alto: Muy Alto: Trapecio (x, 21, 28, nulo, nulo)	Trapecio (x, nulo, 0, 1, 2) Muy Trapecio (x, 0, 2, 3, 4) Bajo: Trapecio (x, 2, 5, nulo, nulo) Triángulo (x, 5.4, 6.3, 7.2) Muy Alto: Triángulo (x, 7.2, 8.1, 9)	Bajo: Triángulo (x, 0, 0.9, 1.8) Triángulo (x, 1.8, 2.7, 4.5) Medio: Triángulo (x, 3.6, 4.5, 5.4)
Consultas Externas		
Muy Bajo: Trapecio (x, nulo, 1, 3, 7) Bajo: Trapecio (x, 1, 5, 7, 9) Medio: Trapecio (x, 6, 9, 16, 20) Alto: Trapecio (x, 10, 16, 21, 28) Muy Alto: Trapecio (x, 21, 28, nulo, nulo)	Bajo: Trapecio (x, nulo, 0, 1, 2) Medio: Trapecio (x, 0, 2, 3, 4) Alto: Trapecio (x, 2, 5, nulo, nulo)	Muy Bajo: Triángulo (x, 0, 0.8, 1.6) Bajo: Triángulo (x, 1.6, 2.4, 3.2) Medio: Triángulo (x, 3.2, 4, 4.8) Alto: Triángulo (x, 4.8, 5.6, 6.4) Muy Alto: Triángulo (x, 6.4, 7.2, 8)

2. Definición de las reglas que relacionan los datos de entrada con los datos de salida, determinando todas las posibles combinaciones entre los números difusos definidos abarcando todo el espacio definido en los dominios de las variables.

Las reglas borrosas consisten en la combinación de uno o más conjuntos borrosos de entrada, a los cuales se les llama premisas o antecedentes, a las que se le asocia un conjunto borroso de salida denominado conclusión o consecuente. La combinación de los conjuntos borrosos de entrada se realiza mediante expresiones lógicas, las cuales se traducen en una operación entre los combinados. Todas las reglas definidas forman parte entonces de la base de reglas que permite expresar el conocimiento de una forma completa. (Maguiña, 2010)

La base de reglas se representa como una memoria asociativa borrosa (MAB), donde se define la consecuencia para cada combinación de dos entradas en una matriz. En la tabla 2 se muestra la MAB definida para la generalización del método Puntos de Función.

Tabla 2: Memoria asociativa borrosa propuesta para la generalización del método Puntos de Función

Base de Reglas para los PF		TED				
		Muy Bajo	Bajo	Medio	Alto	Muy Alto
TER / TAR	Bajo	Muy Bajo	Muy Bajo	Medio	Alto	Alto
	Medio	Muy Bajo	Bajo	Medio	Alto	Muy Alto
	Alto	Bajo	Bajo	Medio	Muy Alto	Muy Alto

Según la MAB definida, se especifican 15 reglas a evaluar en el proceso de estimación utilizando el método propuesto, estas se muestran en la tabla 3. Se debe especificar que en todas las reglas será utilizada la expresión lógica “y”, lo cual deriva en la operación de intersección entre los conjuntos difusos de entrada, donde se calcula el mínimo para todos los casos.

Tabla 3: Relación de reglas resultantes para la generalización del método Puntos de Función

Número	TED	TER/TAR	PF
1.	Muy Bajo	Bajo	Muy Bajo
2.	Muy Bajo	Medio	Muy Bajo
3.	Muy Bajo	Alto	Bajo
4.	Bajo	Bajo	Muy Bajo
5.	Bajo	Medio	Bajo
6.	Bajo	Alto	Bajo
7.	Medio	Bajo	Medio
8.	Medio	Medio	Medio

9.	Medio	Alto	Medio
10.	Alto	Bajo	Alto
11.	Alto	Medio	Alto
12.	Alto	Alto	Muy Alto
13.	Muy Alto	Bajo	Alto
14.	Muy Alto	Medio	Muy Alto
15.	Muy Alto	Alto	Muy Alto

3. Definición de los datos de salida para cada regla definida, a través de la unión de los conjuntos de números difusos resultantes en un conjunto final para su concreción.

Teniendo en cuenta la MAB definida se debe aplicar el operador T-normas para obtener el conjunto borroso de salida de cada regla. Las T-normas más utilizadas son: el mínimo, el producto algebraico, el producto acotado, el producto drástico, el producto de Hamacher y el producto de Einstein. La T-norma elegida para este trabajo fue el producto lógico o del mínimo, el cual tiene la siguiente representación: $T_{\min}(a, b) = \min(a, b) = A \cap B$. (Peregrín, 2000)

Luego de obtener todos los conjuntos borrosos para cada una de las reglas que permanecen activas, se debe formar un único conjunto borroso que debe ser transformado en un valor nítido que pueda ser utilizado para la estimación del esfuerzo del proyecto. Para esto el modelo de Mamdani propone los métodos: Centro de área, Bisector del área, Media de máximo, Máximo más chico y Máximo más grande. (Bojórquez y Bojórquez, 2014)

Para el proceso de concreción en este trabajo se propone utilizar el método Centro de área, que se encuentra definido por la expresión mostrada en la figura 2, donde $\mu_A(z)$ es el conjunto borroso resultante de evaluar todas las reglas definidas y z representa la media del peso asociado a los PF en cada una de las reglas.

$$COA = \frac{\sum_{z=a}^b \mu_A(z) \cdot z}{\sum_{z=a}^b \mu_A(z)}$$

Figura 2. Fórmula para calcular el Centro del área. (Bojórquez y Bojórquez, 2014)

Para exponer los resultados que se obtienen en este paso se mostrará a continuación el cálculo realizado para el caso presentado en la introducción de este trabajo que refleja los límites bruscos del método; donde a un archivo interno

lógico que contiene 19 TED y 6 TER le son asignados 7 puntos de función y otro con 20 TED y 6 TER le son asignados 10 puntos de función. Además, se muestra un tercer caso para un archivo interno lógico al cual se le asigna, en el método sin generalizar, la misma cantidad de puntos de función que el segundo caso presentado sin tener en cuenta la diferencia significativa que existe entre los TED de ambos casos.

1. AIL con 19 TED y 6 TER
 $\mu_{TED\ BAJO} = (20-19/20-12) = 0,13$, $\mu_{TED\ MEDIO} = (19-10/20-10) = 0,9$, $\mu_{TER\ ALTO} = 1$, Reglas activas: 6 y 9
 $PF(19, 6) = (0.13*6 + 0.9*10) / 0.13 + 0.9 = 8.56$
2. AIL con 20 TED y 6 TER
 $\mu_{TED\ MEDIO} = 1$, $\mu_{TER\ ALTO} = 1$, Regla activa: 9
 $PF(20, 6) = (1*10) / 1 = 9$
3. AIL con 30 TED y 5 TER
 $\mu_{TED\ MEDIO} = 1$, $\mu_{TED\ ALTO} = (30-28/38-28) = 0.2$, $\mu_{TER\ MEDIO} = 1$, $\mu_{TER\ ALTO} = (5-3/6-3) = 0.67$ Reglas activas: 8, 9, 11 y 12
 $PF(30, 5) = (1*10 + 0.67*10 + 0.2*14 + 0.2*18) / (1 + 0.67 + 0.2 + 0.2) = 10.05$

Resultados y discusión

Para validar la generalización realizada al método Puntos de Función utilizando lógica difusa, se utilizó como caso de estudio el módulo de Selectividad del Sistema Único de Aduana que fue desarrollado para la Aduana General de la República de Cuba en la Universidad de las Ciencias Informáticas. El análisis realizado se basó en las variaciones que se obtuvieron al estimar utilizando el método Puntos de Función clásico y con el método generalizado propuesto en este trabajo, contra el tiempo real transcurrido para el desarrollo de este módulo.

El módulo Selectividad contiene once funcionalidades, que fueron clasificadas en los componentes del método Puntos de Función de la manera en que se muestran en la tabla 4. Luego fueron calculados los puntos de función y el esfuerzo, para comparar los resultados obtenidos contra el tiempo real en el que fue desarrollado el módulo. Se debe especificar que el factor de ajuste que se utilizó en el caso de estudio para ajustar los puntos de función que responde a las características propias del módulo desarrollado fue de 1.07; así como el valor de 8 horas/PF para la tasa de productividad ya que es el valor recomendado en (Salazar-B, 2009) para calcular el esfuerzo en organizaciones que no cuenten con un repositorio de métricas que les permita actualizar dicho parámetro.

Tabla 4: Componentes del método Puntos de Función identificados para el módulo Selectividad

Variab	Cantid	TED	TER / TAR
Entradas	1	1	1
Externas	1	2	1

	1	6	1
	2	8	1
	1	10	1
	2	16	1
Consultas Externas	1	4	1
	1	6	1
	1	8	1
	1	9	1
	1	10	1
Archivos de interfaz lógicos	1	52	12

El módulo Selectividad contiene once funcionalidades, que fueron clasificadas en los componentes del método Puntos de Función de la manera en que se muestran en la tabla 4. Luego fueron calculados los puntos de función y el esfuerzo, para comparar los resultados obtenidos contra el tiempo real en el que fue desarrollado el módulo. Se debe especificar que el factor de ajuste que se utilizó en el caso de estudio para ajustar los puntos de función que responde a las características propias del módulo desarrollado fue de 1.07; así como el valor de 8 horas/PF para la tasa de productividad ya que es el valor recomendado en (Salazar-B, 2009) para calcular el esfuerzo en organizaciones que no cuenten con un repositorio de métricas que les permita actualizar dicho parámetro.

En la figura 3 se muestran las horas que fueron invertidas en el desarrollo del módulo Selectividad, junto a las horas que fueron estimadas por el método Puntos de Función tradicional y la propuesta de generalización realizada en la investigación. Con los resultados obtenidos se comprueba que con la propuesta de generalización utilizando lógica difusa se obtienen estimaciones más precisas que con el método Puntos de Función tradicional.

En ambos casos las variaciones de horas obtenidas se encuentran en un rango de hasta un 5 % por encima o por debajo de las horas dedicadas al desarrollo, lo cual permite considerar a ambas estimaciones como factibles según la tendencia de la industria. Es necesario resaltar que el módulo Selectividad contiene solamente once funcionalidades, por lo que la variación de las horas obtenidas será mayor en un sistema de mayor complejidad. Además, se debe señalar que la estimación obtenida con el método Puntos de Función tradicional estuvo por debajo de lo sucedido en la realidad, lo que significa que al utilizarse este método no se hubieran planificado todos los recursos necesarios para su desarrollo y que por tanto a estos recursos le fueran planificadas otras tareas, las cuales se verían afectadas al no terminarse el sistema en el tiempo planificado y necesitarse por un tiempo superior a lo planificado.

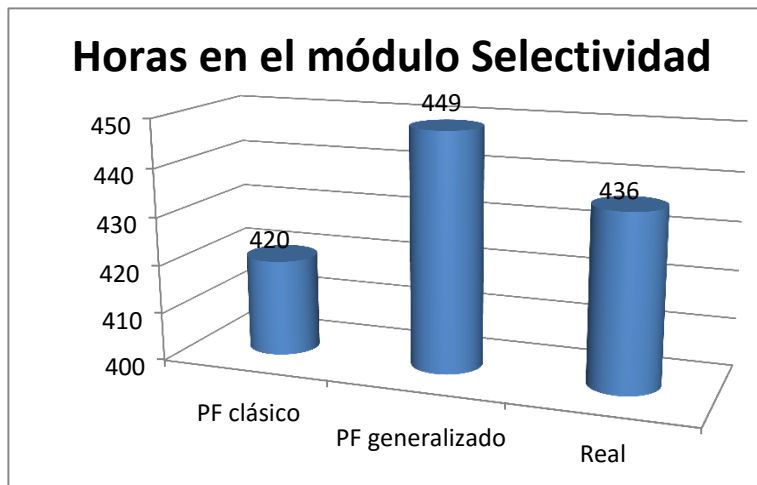


Figura 3. Comparación del método Puntos de Función tradicional con el generalizado y el tiempo real de desarrollo

Por último, se considera que un factor fundamental para garantizar el correcto funcionamiento de la generalización propuesta en la investigación, lo constituye el conteo de las funciones de datos y transaccionales del software que se estima. Para esto deben ser estudiadas con profundidad todas las aclaraciones, reglas, guías y criterios que contiene el manual de prácticas de conteo de la IFPUG. Al garantizarse este factor se podrán ir creando registros históricos consistentes con todos los sistemas que se estimen al realizarse un conteo correcto y homogéneo, lo que permitirá calibrar los conjuntos utilizados en la propuesta para lograr una mayor precisión en la estimación.

Conclusiones

El método de estimación Puntos de Función ha ganado popularidad en los últimos años, pero continúa presentando deficiencias a la hora de clasificar los componentes de un sistema y asignar la cantidad de puntos de función a estos componentes. En este trabajo se ha presentado una generalización para este método utilizando la lógica difusa para eliminar los límites bruscos y minimizar el efecto del arranque en frío. Para esto fueron definidas las variables de entrada y sus etiquetas lingüísticas utilizando los modelos matemáticos funciones trapezoidales y triangulares.

La utilización del operador T-normas del mínimo para obtener la salida en cada una de las reglas, permite obtener resultados con menor sensibilidad ante los pequeños cambios que se introduzcan. Además de utilizarse el método Centro del área para la concreción de los datos de salida obtenidos en cada una de las reglas al ser considerado el método más potente de su tipo y por tanto es el más utilizado.

La propuesta presentada fue aplicada en uno de los módulos de un sistema desarrollado en la Universidad de las Ciencias Informáticas. Los resultados obtenidos en el experimento demostraron que con la generalización se pueden obtener estimaciones más cercanas a la realidad que al utilizar el enfoque clásico del método. El proceso de calibración de los números difusos de cada una de las variables definidas para reflejar las características propias de la empresa donde se aplique, constituye un área de trabajo futuro.

Referencias

- ALBRECHT, A. Measuring Application Development Productivity. Proceedings of the IBM Application Development Symposium. Monterey, CA: IBM Corporation, 1979, p. 83-92.
- BOJÓRQUEZ, G.; BOJÓRQUEZ, J. Metodología para la implementación de sistemas difusos tipo Mamdani en lenguajes de programación de propósito general. En: Congreso Internacional en Ingeniería Electrónica. Memoria Electro. México: Instituto Tecnológico de Chihuahua, 2014, p. 318-323.
- CHEN, Q.; CHENG, R.; et al. Study of Function Points Analysis Based on Fuzzy-Interpolation. Journal of Computational Information Systems, 2010, 6 (5): p. 1369-1375.
- CHUEN, L. Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II. Systems, Man and Cybernetics, IEEE Transactions, 1990, 20(2): p. 419-435.
- CMMI Product Team. CMMI for Development, Version 1.3. [En línea]. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-033, 2010, [Consultado el: 14 de enero de 2015]. Disponible en: [<http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>].
- DCG, David Consulting Group. Informe 2012 Métodos Utilizados en la Industria para Medir el Software. Laboratorios de las Tecnologías de la Información, 2013. [Consultado el: 3 de febrero de 2015]. Disponible en: [<http://www.laboratorioti.com/2013/02/18/informe-2012-metodos-utilizados-en-la-industria-para-medir-el-software/>].
- FERREIRA, G. Estimación del esfuerzo en proyectos de software utilizando técnicas de inteligencia artificial. Revista Cubana de las Ciencias Informáticas, 2014, 8 (4): p. 1-20.
- HENDERSON, D.; SHEETZ, S.; et al. Understanding Software Metric Use. ISACA Journal, 2015, volume 1. Disponible en: [<http://www.isaca.org/Journal/archives/2015/Volume-1/Pages/Understanding-Software-MetricUse.aspx>].

IFPUG, International Function Point Users Group. Function Point Counting Practices Manual. New Jersey: 2010, Release 4.3.1. ISBN 978-0-9753783-4-2.

JASSBI, J.; ALAVI, S. H.; et al. Transformation of a Mamdani FIS to first order sugeno FIS. En: IEEE international conference on Fuzzy systems. FUZZ-IEEE 2007. London, United Kingdom: IEEE International, 2007, p. 1-6.

LIMA, O.; MUNIZ, P.; et al. Fuzzy modeling for function points analysis. Software Quality Journal, 2003, 11: p. 149–166.

MAGUIÑA, R. A. Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio. Revista de Investigación de Sistemas de Informática, 2010, 7 (1): p. 91-104.

PEREGRÍN, A. Integración de operadores de implicación y métodos de defuzzificación en sistemas basados en reglas difusas. Implementación, análisis y caracterización. Tesis Doctoral, Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial, Granada, 2000.

PIÑERO, P. Un modelo para el aprendizaje y la clasificación automática basado en técnicas de soft computing. Tesis Doctoral, Universidad Central Marta Abreu de las Villas, Departamento de Ciencias de la Computación, Villa Clara, 2005.

SALAZAR-B, G. Estimación de proyectos de software: un caso práctico. Revista de Ingeniería y Ciencia, 2009, 5 (9): p. 123-143.

XIA, W.; HO, D.; et al. A Neuro-Fuzzy Model for Function Point Calibration. WSEAS Transactions on Information Science & Applications, 2008, Issue 1, Volume 5, ISSN: 1790-0832.

YAU, C.; TSOI, H.-L. Modelling the probabilistic behaviour of function point analysis. Information and Software Technology, 1998, 40: p. 59–68.