

# **INSTITUTO POLITÉCNICO NACIONAL**

---

## **CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**

**MODELO PARA LA ESTIMACIÓN DEL ESFUERZO DE DESARROLLO  
EN PROYECTOS DE SOFTWARE A NIVEL PERSONAL APLICANDO  
LÓGICA DIFUSA**

**QUE PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS COMPUTACIONALES**

**PRESENTA:**

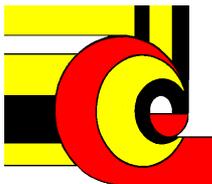
**CUAUHTÉMOC LÓPEZ MARTÍN**

**DIRECTOR DE TESIS:**

**DR. AGUSTÍN FRANCISCO GUTIÉRREZ TORNÉS**

**CODIRECTOR DE TESIS:**

**DR. CORNELIO YÁÑEZ MÁRQUEZ**



**MÉXICO, D.F.**

**ABRIL 2007**

**Agradecimientos  
en el entorno familiar**

*A Citlalli<sup>1</sup> ... mi estrella alcanzable.*

*A Silvia, esposa y excelente madre.*

*Ambas fuente inspiradora para  
la culminación de esta obra.*

*A Teresa y Jaime, de quienes heredé sus  
genes y seguí su ejemplo que guió  
mi moral e intelecto durante el  
desarrollo de este trabajo.*

*A Teresa, Mario y Jaime, mis hermanos.*

---

<sup>1</sup> *Citlalli en lengua náhuatl significa estrella*

**Agradecimientos  
en el entorno institucional**

*Al Centro de Investigación en Computación  
del Instituto Politécnico Nacional (CIC-IPN).*

*Al Consejo Nacional de Ciencia y Tecnología  
(CONACYT).*

## **Agradecimientos en el entorno académico**

*Al Dr. Agustín Gutiérrez Tornés y al  
Dr. Cornelio Yáñez Márquez,  
directores de esta tesis, por su adecuada,  
continua e ingente orientación.*

*Al Dr. Grigori Sidorov y al  
Dr. Igor Bolshakov,  
quienes vaticinaron la culminación  
de esta obra.*

*Al Dr. Jérôme Leboeuf Pasquier,  
del Centro Universitario de Ciencias Exactas  
e Ingenierías (CUCEI) de la Universidad de  
Guadalajara, de quien se recibió el primer  
curso de la apasionante lógica difusa.*

*Al M.C. Porfirio Gutiérrez y al  
M.C. Osvaldo Camacho  
del Laboratorio de Estadística del CUCEI,  
donde fueron adquiridos los  
conocimientos estadísticos  
que fundamentan  
esta tesis.*

*Al Dr. Carlos Montes de Oca y al  
Dr. Miguel Ángel Serrano,  
del Centro de Investigación en Matemáticas (CIMAT)  
de quienes se recibió el curso del  
Personal Software Process.*

*A los Doctores  
Aurora Pérez Rojas,  
Edgardo Felipe Riverón,  
Sergio Suárez Guerra y  
Jesús Olivares Ceja  
por sus valiosas aportaciones  
para el perfeccionamiento  
de esta tesis.*

## Resumen

Esta investigación está fundamentada en los siguientes hechos: (1) La estimación del esfuerzo de desarrollo de software ha sido identificada como uno de los tres grandes desafíos de la ciencia computacional; (2) La estimación del esfuerzo de desarrollo de software es una de las actividades más críticas en la administración de proyectos de software; (3) Debido a que una sola técnica para estimar el esfuerzo de desarrollo de software no es la mejor para todas las situaciones, una cuidadosa comparación de los resultados de diversos enfoques es lo más adecuado para generar estimaciones más realistas; (4) El 90% de las empresas desarrolladoras de software mexicanas no tienen procesos formales para registrar, dar seguimiento y controlar aspectos mensurables durante el proceso de desarrollo de software (incluyendo la estimación del esfuerzo de desarrollo) y (5) Los ingenieros de software no podrán apoyar apropiadamente a sus equipos de manera consistente y confiada para generar productos con calidad, a menos que tengan las capacidades provistas por un entrenamiento personal.

En esta investigación, 364 programas desarrollados por cincuenta y dos programadores integraron la población de datos. A partir de esta población, se seleccionó una muestra de 105 programas de treinta desarrolladores de software y a partir de esta muestra fueron generados tres Sistemas de Lógica Difusa (SLD) para estimar el esfuerzo de desarrollo de veinte programas desarrollados por otro grupo distinto de siete desarrolladores. Todos los programas fueron desarrollados con base en las mismas prácticas personales. Los SLD fueron comparados con una ecuación de regresión lineal. Estas comparaciones fueron principalmente basadas en la Magnitud del Error Relativo a la estimación de cada programa (MER) y su media (MMER), así como en la dependencia entre las MER y el esfuerzo (análisis de heterocedasticidad). En este estudio, la clásica Magnitud Relativa del Error (MRE) por programa y su media (MMRE), no son consideradas como principales criterios de evaluación debido a que algunos investigadores han demostrado recientemente sus debilidades; sin embargo las MRE, MMRE y otras medidas como la mediana de las MRE y MER (MdMRE y MdMER, respectivamente) así como el Pred(25) son calculados para poder ser comparados con otras investigaciones.

Hay dos pasos principales para usar un modelo de estimación: (1) debe determinarse si el modelo es adecuado para describir los datos (reales) observados, es decir, la comprobación de la adecuación del modelo (o verificación); si éste resultara adecuado, entonces (2) el modelo de estimación es validado haciendo uso de nuevos datos. Esta investigación considera estos dos pasos para los modelos. En el paso uno, el valor de la MMER de la regresión lineal fue de 0.25, mientras que de los modelos difusos fueron los siguientes: triangular = 0.25, gaussiana = 0.27 y trapezoidal = 0.28. El análisis de varianza (ANOVA) de las MER de los cuatro modelos no mostró una diferencia estadísticamente significativa. En el segundo paso (validación del modelo), el SLD de tipo triangular obtuvo mejor resultado cuando la heterocedasticidad no fue considerada, mientras que cuando fue considerada, en la primera submuestra fue mejor y en la segunda submuestra sólo fue igualada por el tipo trapezoidal. En los dos casos de la validación por submuestras, el tipo triangular obtuvo una MMER=0.23.

Tres hipótesis fueron formuladas y probadas. Esta investigación siguió una rigurosa metodología de investigación. Esto sugiere concluir que la lógica difusa puede ser una alternativa para estimar el esfuerzo de desarrollo de software a nivel personal cuando programas son desarrollados con base en prácticas personales.

# Índice

	Página
Siglas y acrónimos	1
Tablas y figuras	2
Glosario	3
Capítulo 1. Introducción	6
1.1 Antecedentes	6
1.1.1 El origen de la ingeniería del software	6
1.1.2 La industria del software en el extranjero	6
1.1.3 La industria del software en México	7
1.1.4 La estimación del esfuerzo de desarrollo en la academia internacional y nacional	9
1.1.5 El entrenamiento personal previo a la integración del equipo	10
1.2 Planteamiento del problema	11
1.3 Justificación	12
1.4 Objetivos	14
1.5 Método	14
1.5.1 Enfoque de la investigación	14
1.5.2 Definición del alcance de la investigación	15
1.5.3 Formulación de hipótesis	15
1.5.4 Selección del diseño apropiado para la investigación	17
1.5.5 Selección de la población y muestra	19
1.6 Organización del documento de tesis	20
Capítulo 2. Marco teórico	22
2.1 El proceso de estimación	22
2.2 Proceso personal de software	23
2.3 Técnicas de estimación	25
2.3.1 Necesidad de comparar técnicas	27
2.3.2 Criterios de evaluación para comparación de técnicas	27
2.4 Medición del software	29
2.5 Estadística	29
2.5.1 Correlación y coeficiente de determinación	29
2.5.2 Regresión lineal simple y múltiple	30
2.5.3 Valores atípicos y de influencia	32
2.5.4 Pruebas para comparación de medias	32
2.6 Lógica difusa	34
2.6.1 Conjuntos difusos	35
2.7 Estado del arte	38

## Índice (continuación)

	Página
Capítulo 3. Modelo propuesto	41
3.1 Proceso para la generación del modelo difuso	41
3.2 Recolección de datos	42
3.3 Generación de modelos	54
3.3.1 Modelos de regresión	54
3.3.1.1 Regresión lineal múltiple	54
3.3.1.2 Regresión lineal simple	56
3.3.2 Modelos difusos	57
3.3.2.1 Modelo difuso con funciones de membresía triangulares	59
3.3.2.2 Modelo difuso con funciones de membresía gaussianas	60
3.3.2.3 Modelo difuso con funciones de membresía trapezoidales	62
3.4 Análisis de datos	63
3.4.1 Comprobación de la adecuación de los modelos (verificación)	63
Capítulo 4. Resultados	69
4.1 Procedimiento para la validación de los modelos	69
4.2 Obtención de las MRE y MER	69
Capítulo 5. Conclusiones, recomendaciones, trabajo futuro y legado de la investigación	76
5.1 Conclusiones	76
5.2 Recomendaciones	78
5.3 Trabajo futuro	79
5.4 El legado de la investigación	79
Referencias	81
Anexos	87
A. Formas de registro	87
B. Datos de desarrolladores	90
C. Datos para la generación de los modelos de estimación	92
D. Datos para la validación de los modelos de estimación	98
E. Predicción del esfuerzo mediante el modelo de regresión ( $10 \leq NyC \leq 182$ )	100
F. Predicción del esfuerzo mediante el modelo difuso con funciones de membresía triangulares ( $10 \leq NyC \leq 182$ )	101
G. Predicción del esfuerzo mediante el modelo difuso con funciones de membresía gaussianas ( $10 \leq NyC \leq 182$ )	102
H. Predicción del esfuerzo mediante el modelo difuso con funciones de membresía trapezoidales ( $10 \leq NyC \leq 182$ )	103

## Siglas y Acrónimos

AMITI	Asociación Mexicana de la Industria de Tecnologías de Información
ANIEI	Asociación Nacional de Instituciones de Educación en Informática
CME	Cuadrado medio del error
CMM	Modelo de Madurez de la Capacidad ( <i>Capability Maturity Model</i> )
CMMi	Integración del Modelo de Madurez de la Capacidad ( <i>Capability Maturity Model Integration</i> )
CMR	Cuadrado medio debido a la regresión
CMT	Cuadrado medio debido a los tratamientos
CMU	Carnegie Mellon University
COCOMO	Modelo Constructivo de Costo ( <i>Constructive Cost Model</i> )
CONACYT	Consejo Nacional de Ciencia y Tecnología
KPA	Área clave de proceso del CMM ( <i>Key Process Area</i> )
LSD	Diferencia mínima significativa de Fisher ( <i>least significant difference</i> )
MdMER	Mediana de las MER
MdMRE	Mediana de las MRE
MER	Magnitud del Error Relativo a la Estimación ( <i>Magnitude of Error Relative to the Estimate</i> )
MMER	Media de las MER
MMRE	Media de las MRE
MoProSoft	Modelo de Proceso de Software de México
MRE	Magnitud Relativa del Error ( <i>Magnitud of Relative Error</i> )
NyC	Líneas de código adicionado (nuevo) + modificado (cambiado).
PDIS	Programa para el Desarrollo de la Industria de Software de México
PND	Plan Nacional de Desarrollo de México 2001–2006
Pred(25)	Porcentaje de programas que fueron estimados con una MRE (o MER) menor o igual a 0.25
PSP	Proceso de Software Personal ( <i>Personal Software Process</i> )
SCE	Suma de cuadrados debidos al error
SCR	Suma de cuadrados de la regresión
SCT	Suma de cuadrados total (en regresión lineal) y Suma de cuadrados debido a los tratamientos
SEI	Instituto de Ingeniería de Software ( <i>Software Engineering Institute</i> ) ubicado en la universidad de <i>Carnegie Mellon (CMU)</i>
SPICE	Mejoramiento del Proceso de Software y Determinación de la Capacidad, ISO 15504 ( <i>Software Process Improvement and Capability dEtermination</i> )
STC	Suma total de cuadrados (en ANOVA)
SWEBOK	Cuerpo de Conocimiento de la Ingeniería de Software ( <i>Software Engineering Body of Knowledge</i> )
SWEBOK-SEI	Cuerpo de Conocimiento de la Ingeniería de Software Software del Instituto de Ingeniería de Software ( <i>Software Engineering Body of Knowledge - Software Engineering Institute</i> )
TSP	Proceso de Software en Equipo ( <i>Team Software Process</i> )

# Tablas y figuras

## Tablas

- 1 Metas de crecimiento para el 2010 en la industria de software en México
- 2 Proceso para la generación, verificación y validación de un modelo de estimación difuso
- 3 Estándar de cuantificación de código
- 4 Programas desarrollados dentro de la investigación
- 5 Datos de desarrolladores para la generación de modelos aplicando criterio A
- 6 Programas dentro de la muestra con valor  $z$  más pequeño
- 7 Programas atípicos excluidos para generación de modelos
- 8 Resumen de programas excluidos para generación de modelos
- 9 Datos de desarrolladores para la validación de modelos aplicando criterio A
- 10 Programas menores que 10 NyC excluidos para validación de modelos
- 11 Programas atípicos excluidos para validación de modelos
- 12 Resumen de programas excluidos para validación de modelos
- 13 Análisis de parámetros del modelo de regresión lineal múltiple
- 14 ANOVA del modelo de regresión lineal múltiple
- 15 ANOVA del modelo de regresión lineal simple considerando sólo programas con reutilización de código
- 16 Análisis de parámetros del modelo de regresión lineal simple
- 17 ANOVA del modelo de regresión lineal simple
- 18 Características de las funciones de membresía triangulares
- 19 Características de las funciones de membresía gaussianas
- 20 Características de las funciones de membresía trapezoidales
- 21 MRE y MER de cada programa para la comprobación de la adecuación de modelos
- 22 ANOVA de la MRE y MER de modelos
- 23 Prueba de rangos múltiples de la MRE y MER
- 24 Muestra de programas para validación de los modelos
- 25 MRE y MER por programa para validación de modelos
- 26 Comparación de medias entre submuestras A y B
- 27 MMRE y MMER por submuestras A y B
- 28 Resumen de criterios de evaluación y comparación de modelos

## Figuras

- 1 La investigación en la economía de la ingeniería del software
- 2 Proceso para la estimación del software
- 3 Proceso del PSP
- 4 Diagrama de dispersión de la Tabla 5
- 5 Diagrama dispersión de la muestra de programas para esta investigación
- 6 Funciones de membresía triangulares del sistema difuso
- 7 Funciones de membresía gaussianas del sistema difuso
- 8 Funciones de membresía trapezoidales del sistema difuso
- 9 Gráficas de varianzas constantes y de normalidad
- 10 Gráficas de medias de MRE y MER
- 11 Gráficas de dispersión (MRE y MER versus esfuerzo) y sus valores de correlación ( $r$ )
- 12 Gráficas de dispersión de submuestras A
- 13 Gráficas de dispersión de submuestras B

## Glosario

Alcance de la investigación	Es el grado de profundidad de una investigación, puede tener los siguientes niveles: exploratoria, descriptiva, correlativa y hasta explicativa.
Código reutilizado	Código tomado de alguna biblioteca y usado sin modificación alguna por algún programa.
Coefficiente de determinación ( $r^2$ )	Proporción de la variación en los valores observados en la variable de respuesta que es explicada por la regresión.
Comprobación de la adecuación del modelo	Determinación de si el modelo de estimación es adecuado para describir los valores observados. También se le conoce como verificación del modelo.
Conjunto difuso	Conjunto con valores de pertenencia dentro del segmento $[0,1]$ , y no solo 0 (no pertenece) ó 1 (pertenece).
Correlación ( $r$ )	Grado en que dos conjuntos de datos están relacionados. El valor de la correlación $r$ varía de $-1.0$ a $+1.0$ .
Defuzificación	Construir un sistema de inducción de reglas reemplazando los hechos abruptos ( <i>crispy</i> ) con entradas difusas. Luego una máquina de inferencia usa las reglas para corresponder la entrada con la salida difusa.
Desarrollo de software	Fases del proceso genérico de software que involucran al diseño, codificación y prueba, además de aquellas protectoras consideradas.
Diseño de investigación	Plan o estrategia concebida para obtener la información que se desea. Existen los tipos siguientes: experimental (preexperimental, puro y cuasiexperimental) y no experimental.
Enfoque cualitativo de la investigación	Se utiliza primero para descubrir y refinar preguntas de investigación. A veces, pero no necesariamente, se prueban hipótesis.
Esfuerzo de desarrollo de software	Cantidad de tiempo requerido por un conjunto de personas para diseñar, codificar, probar y ejecutar otras actividades de apoyo relacionadas directamente con la obtención de un producto de software.
Fuentes de invalidación interna	Factores a considerar dentro de un experimento para valorar su proceso. Entre ellos la historia, maduración, inestabilidad, administración de las pruebas, instrumentación, regresión estadística, selección, mortalidad experimental y la interacción entre selección y maduración.
Funciones de membresía	Representación de conjuntos difusos dentro de un sistema difuso. Existen entre otras funciones de membresía las triangulares, gaussianas y trapezoidales.
Heterocedasticidad	Datos distribuidos dentro de una gráfica de dispersión no mostrándose dentro de una banda horizontal centrada y simétrica.
Hipótesis	Indican lo que se está tratando de probar y se definen como explicaciones tentativas del fenómeno investigado; son formuladas a manera de proposiciones y sujetas a comprobación empírica. Existen de investigación (o de trabajo), nulas, alternativas y estadísticas.
Homocedasticidad	Datos distribuidos similarmente dentro de una gráfica de dispersión mostrándose aproximadamente en una banda horizontal centrada y simétrica.
Línea de código	Instrucción escrita en algún lenguaje de programación, se clasifican en base, adicionado (nuevo), modificado (cambiado), eliminado, nuevo y cambiado (NyC) y reusado.
Lenguaje de programación imperativo	Lenguajes con facilidades para la asignación de valores a localidades de memoria (ejemplos: C, Pascal, Basic, COBOL, PHP y JAVA).

## **Glosario** *(continuación)*

Lenguaje de programación declarativo	Lenguajes basados en relaciones y funciones, en los que el programador no considera la asignación de valores de localidades de almacenamiento de memoria (ejemplos: PROLOG, LISP, ZIM y SQL).
Modelo difuso	Modelo que opera a nivel de términos lingüísticos (conjuntos difusos representados por funciones de membresía). Asimismo, el modelo representa incertidumbre.
Muestra	Subconjunto de una población, existen muestras no probabilísticas (o dirigidas) y probabilísticas.
Población	Conjunto de todos los casos que concuerdan con determinadas especificaciones.
Proceso de estimación	Serie de actividades a seguir para las diversas estimaciones de un proyecto de software involucrando análisis de requerimientos, estimación del tamaño, estimación del esfuerzo, estimación de costos, estimación de la calendarización, estimación de recursos de computadora, evaluación de riesgos, inspección y aprobación de estimaciones, así como calibración y mejoramiento de este proceso.
Proceso de estimación del esfuerzo	Actividades específicas para estimar el esfuerzo de desarrollo, incluyendo la obtención de datos a partir de proyectos anteriores, generación de modelos de estimación y evaluación de qué tan efectivos con los modelos con base en su precisión.
Proceso del PSP	Fases que el PSP considera para el software: planeación, diseño, revisión de diseño, codificación, revisión de código, compilación, pruebas y postmortem.
Regresión lineal múltiple	Considera la relación lineal que existe entre una variable dependiente y más de una independiente.
Regresión lineal simple	Considera la relación lineal que existe entre una variable dependiente y una independiente.
Residual	Diferencia entre la observación y la media de la muestra que la contiene.
Supuesto de independencia	Cada dato de la muestra debe corresponder a una observación independiente de las otras.
Supuesto de normalidad	Datos de muestra mostrando una distribución normal. Una gráfica normal de residuales debería ser aproximadamente lineal.
Supuesto de varianza constante	Datos de muestra con varianzas iguales. En una gráfica de este tipo los residuales deberían mostrarse aproximadamente en una banda horizontal centrada y simétrica.
Validación del modelo	Se concentra en la determinación de si el modelo funcionará bien en su ambiente pretendido de operación.
Valor atípico	Observación que no se ajusta a la tendencia que muestran los datos restantes.
Valor de influencia	Puede ser también un valor atípico. Es el alejamiento extremo del valor de la observación de la variable independiente.

## Resumen

Esta investigación está fundamentada en los siguientes hechos: (1) La estimación del esfuerzo de desarrollo de software ha sido identificada como uno de los tres grandes desafíos de la ciencia computacional; (2) La estimación del esfuerzo de desarrollo de software es una de las actividades más críticas en la administración de proyectos de software; (3) Debido a que una sola técnica para estimar el esfuerzo de desarrollo de software no es la mejor para todas las situaciones, una cuidadosa comparación de los resultados de diversos enfoques es lo más adecuado para generar estimaciones más realistas; (4) El 90% de las empresas desarrolladoras de software mexicanas no tienen procesos formales para registrar, dar seguimiento y controlar aspectos mensurables durante el proceso de desarrollo de software (incluyendo la estimación del esfuerzo de desarrollo) y (5) Los ingenieros de software no podrán apoyar apropiadamente a sus equipos de manera consistente y confiada para generar productos con calidad, a menos que tengan las capacidades provistas por un entrenamiento personal.

En esta investigación, 364 programas desarrollados por cincuenta y dos programadores integraron la población de datos. A partir de esta población, se seleccionó una muestra de 105 programas de treinta desarrolladores de software y a partir de esta muestra fueron generados tres Sistemas de Lógica Difusa (SLD) para estimar el esfuerzo de desarrollo de veinte programas desarrollados por otro grupo distinto de siete desarrolladores. Todos los programas fueron desarrollados con base en las mismas prácticas personales. Los SLD fueron comparados con una ecuación de regresión lineal. Estas comparaciones fueron principalmente basadas en la Magnitud del Error Relativo a la estimación de cada programa (MER) y su media (MMER), así como en la dependencia entre las MER y el esfuerzo (análisis de heterocedasticidad). En este estudio, la clásica Magnitud Relativa del Error (MRE) por programa y su media (MMRE), no son consideradas como principales criterios de evaluación debido a que algunos investigadores han demostrado recientemente sus debilidades; sin embargo las MRE, MMRE y otras medidas como la mediana de las MRE y MER (MdMRE y MdMER, respectivamente) así como el Pred(25) son calculados para poder ser comparados con otras investigaciones.

Hay dos pasos principales para usar un modelo de estimación: (1) debe determinarse si el modelo es adecuado para describir los datos (reales) observados, es decir, la comprobación de la adecuación del modelo (o verificación); si éste resultara adecuado, entonces (2) el modelo de estimación es validado haciendo uso de nuevos datos. Esta investigación considera estos dos pasos para los modelos. En el paso uno, el valor de la MMER de la regresión lineal fue de 0.25, mientras que de los modelos difusos fueron los siguientes: triangular = 0.25, gaussiana = 0.27 y trapezoidal = 0.28. El análisis de varianza (ANOVA) de las MER de los cuatro modelos no mostró una diferencia estadísticamente significativa. En el segundo paso (validación del modelo), el SLD de tipo triangular obtuvo mejor resultado cuando la heterocedasticidad no fue considerada, mientras que cuando fue considerada, en la primera submuestra fue mejor y en la segunda submuestra sólo fue igualada por el tipo trapezoidal. En los dos casos de la validación por submuestras, el tipo triangular obtuvo una MMER=0.23.

Tres hipótesis fueron formuladas y probadas. Esta investigación siguió una rigurosa metodología de investigación. Esto sugiere concluir que la lógica difusa puede ser una alternativa para estimar el esfuerzo de desarrollo de software a nivel personal cuando programas son desarrollados con base en prácticas personales.

## Abstract

This research is founded on the following facts: (1) Software development effort estimation has even been identified as one of the three great challenges for half-century-old computer science, (2) Software development effort estimation is one of the most critical activities in managing software projects, (3) Given that no single software development estimation technique is best for all situations, a careful comparison of the results of several approaches is most likely to produce realistic estimates, (4) the 90% of software Mexican enterprises do not have formal processes to record, track and control measurable issues during the development process (including software effort estimation), and (5) unless engineers have the capabilities provided by personal training, they cannot properly support their teams or consistently and reliably produce quality products.

In this research, 364 programs developed by fifty-two programmers integrated the data population. From this population, a sample of 105 programs of thirty programmers three Fuzzy Logic Systems (FLS) were generated for estimating the effort of twenty programs developed by other group of seven developers. All programs were based upon the same personal practices. With a linear regression equation those four FLS were compared. These comparisons were mainly based on the Magnitude of Error Relative to the estimate of each program (MER) and their mean (MMER), and in considering the dependability between the MER and effort (analysis of heteroscedasticity). In this study, the classical Magnitude of Relative Error by program (MRE) and their mean (MMRE) were not considered as main evaluation criterion because of some researches have recently demonstrated their weakness; however MRE, MMRE and other measures as median of MRE as well as MER (MdMRE and MdMER respectively) and Pred(25) are calculated to be compared with other researches.

There are two main stages for using an estimation model: (1) it must be determined whether the model is adequate to describe the observed (actual) data, that is, the model adequacy checking (verification); if it resulted adequate then (2) the estimation model is validated using new data. This research considers these two stages for models (linear regression and those FLS). In the first stage, the MMER value of linear regression was equal to 0.25, while MMER of those fuzzy models were the following: triangular = 0.25, gaussian = 0.27 and trapezoidal = 0.28. A analysis of variance (ANOVA) of the MER of four models did not show a statistically significant difference among their means. Afterwards the second stage (validation model), the triangular FLS obtained better values than other three models when the heteroscedasticity was not considered. When heteroscedasticity was considered, in the first subsample the triangular type had the better value, while in the second subsample triangular type was only equal to trapezoidal type. In the two cases of validation by subsamples, triangular type had a MMER=0.23.

Three hypotheses were both formulated and tested. This research followed a rigorous methodology for the researching. It allows suggesting that the fuzzy logic can be used as an alternative for estimating the development effort at personal level when programs are developed based upon personal practices.

# Capítulo 1

## Introducción

### 1.1 Antecedentes

#### 1.1.1 El origen de la ingeniería del software

*“NATO report also contains sections reporting on discussions which will be of interest to a much wider audience. This holds for subjects like the difficulties of meeting schedules”*

*Peter Naur, NATO Conference, 1968 [98]*

En octubre de 1968, convocados por el Comité de Ciencia de la Organización del Tratado del Atlántico Norte (OTAN), se reunieron en Garmisch, Alemania, poco más de cincuenta personas de la industria, academia y gobierno de once países [98]. La finalidad de este grupo era discutir diversos temas relacionados con el desarrollo de software. Es en este evento que el concepto de la ingeniería de software surge, siendo actualmente definida por la IEEE como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software, esto es, la aplicación de la ingeniería al software [49].

Un tema relevante en esta reunión fue la dificultad en el cumplimiento de la planeación de proyectos de software. Una actividad previa a la planeación es la estimación del esfuerzo de desarrollo de software, definida como la cantidad de tiempo requerido por un conjunto de personas para diseñar, codificar, probar y ejecutar otras actividades de apoyo relacionadas directamente con la obtención de un producto de software.

#### 1.1.2 La Industria del software en el extranjero

*“El mercado de las tecnologías de información (hardware, software y comunicaciones) representa el 6.6% del valor de la producción económica mundial”*

*Digital Planet: The Global Information Economy.  
WITSA, 2000 [28]*

Durante la década pasada, la mayor parte de los países, aún los que enfrentaron crisis financieras y recesiones económicas, incrementaron su gasto en tecnologías de información y comunicación [99].

En los años recientes, el gasto en tecnologías de información ha trasladado su énfasis del hardware al software, provocando que la relación entre el segundo y el primero suba de 32.5% en 1996 a 40% en 1999. El mercado mundial de productos de software rebasa los 153,000 millones de dólares anuales. Estados Unidos es el principal consumidor, con un gasto superior a los 75,000 millones de dólares anuales y una participación de 48.8% en el total mundial [28].

India ha maquilado software para Estados Unidos durante 15 años de política gubernamental ininterrumpida de apoyo. El gobierno indio ha invertido más de 2.5 billones de dólares en apoyo a la capacitación, certificación, promoción e infraestructura [7].

Irlanda se especializó en el desarrollo de aplicaciones y basó el desarrollo de su industria de software en la atracción de empresas extranjeras por medio de programas de apoyo sumamente agresivos [112]. La industria de software de Canadá basa su producción en las necesidades del mercado externo, con productos enfocados a nichos específicos como: animación, gráficos, administración de documentos, extracción inteligente de datos y administración de escuelas [112]. La industria de software australiana es considerada como sublíder mundial en provisión de servicios de aplicación tales como: contenidos de internet y servicios de administración; servicios en línea que se ofrecen desde Australia a otros países (auditoría, facturación, publicidad) [122].

De acuerdo con la Conferencia de las Naciones Unidas sobre Comercio y Desarrollo, el desarrollo de la industria del software en los países emergentes se puede llevar a cabo a través de dos alternativas de políticas estratégicas [122]:

La primera estrategia se basa en el uso de software libre como una herramienta para el fortalecimiento de la infraestructura económica a fin de soportar el desarrollo de otros sectores de la economía y, la segunda estrategia que se orienta a considerar a la industria del software como sector prioritario por la generación de nuevos empleos y el crecimiento industrial basado en la creación de oferta de software para exportación.

Una opción alternativa para el desarrollo de la industria del software consiste en integrar ambas estrategias bajo un orden secuencial; el uso de software libre como herramienta para el fortalecimiento de la infraestructura económica sería reemplazada gradualmente por la estrategia de desarrollo de software para exportación [112].

### **1.1.3 La industria del software en México**

*“El Plan Nacional de Desarrollo 2001–2006 plantea el objetivo de elevar y extender la competitividad del país, mediante la estrategia de promover el uso y aprovechamiento de la tecnología y de la información”*

*Secretaría de Economía, 2002 [112]*

El Plan Nacional de Desarrollo (PND) 2001–2006 señala que la incorporación y aprovechamiento de los últimos avances científicos y tecnológicos debe basarse en la aplicación de una estrategia coherente que incluya los siguientes puntos: divulgación entre la

---

sociedad de la cultura tecnológica; fomento de la tecnología local y adaptación de la tecnología extranjera; incorporación de las tecnologías de la información (computación, telecomunicación y telemática) al desarrollo de la cultura empresarial; impulso al desarrollo de la industria de tecnologías de la información; fomento de la reconversión digital de procesos dentro de las empresas, especialmente las pequeñas; promoción del encadenamiento digital de proveedores; así como fomentar y difundir la industria de desarrollo de software [112]. El Programa para el Desarrollo de la Industria de Software (PDIS) es uno de los medios que el ejecutivo federal utiliza para definir las líneas de acción con las que llevará a la práctica esa estrategia del PND.

México cuenta con una posición favorable para convertirse en una potencia en este ramo, gracias a su ubicación geográfica, perfil demográfico y estado de desarrollo tecnológico. Para ello, una de las siete estrategias del PDIS (además de (1) promover las exportaciones y la atracción de inversiones, (2) educación y formación de personal competente en el desarrollo de software, en cantidad y calidad convenientes, (3) contar con un marco normativo y promotor de la industria, (4) desarrollar el mercado interno, (5) fortalecer a la industria local y (6) promover la construcción de infraestructura física y de telecomunicaciones) propone trabajar en el alcance de niveles internacionales en capacidad de procesos de software [112].

Existen diversos modelos de proceso, tales como el CMM (*Capability Maturity Model*, Modelo de Madurez de la Capacidad) [102] del Instituto de Ingeniería de *Software* (*SEI*, por sus siglas en inglés *Software Engineering Institute*) ubicado en *Carnegie Mellon University* (*CMU*), el ISO 15504 (o *SPICE: Software Process Improvement and Capability dEtermination*) [116] y el reciente mexicano MoProSoft (Modelo de proceso de *Software*) [95]. En estos tres modelos es considerada la estimación del esfuerzo de desarrollo de software. Esto es que, tácitamente, entre otras prácticas, existe la necesidad por la estimación del esfuerzo de desarrollo de software dentro de empresas.

El CMM se compone de cinco niveles de madurez. Con la excepción del nivel uno, cada nivel se compone de diversas áreas clave del proceso (o KPA, *Key Process Area*). Cada KPA está organizada en las siguientes cinco características comunes (*common features*): (1) compromisos, (2) habilidades, (3) actividades, (4) medición y análisis y (5) verificación, que conjuntamente buscan el logro de metas de la KPA. La planeación de proyectos de software es una KPA del nivel dos y una de sus metas es que “las estimaciones del software sean documentadas para su uso dentro de la planeación y seguimiento del proyecto”. Esta KPA indica dentro de su actividad número nueve que “las estimaciones del tamaño de los productos de trabajo (o cambios a ellos) sean derivados de acuerdo a un procedimiento documentado” incluyendo a las líneas de código como ejemplo de unidad de tamaño; mientras que en su actividad once indica que “las estimaciones para el esfuerzo y costos del proyecto de software son derivados de acuerdo a un procedimiento documentado”. El CMM ha evolucionado al CMMi, versión que igualmente considera a la estimación del esfuerzo de desarrollo de software [23].

### 1.1.4 La estimación del esfuerzo de desarrollo en la academia internacional y nacional

*“It has placed an increasing demand for software developers equipped not only to deal with the scientific and technical aspects of computing, but for those who have professional education and preparation for the practice of software engineering”.*

*SEI, CMU 1999 [6]*

En enero de 1994, Laurel Keleda and Gwenn Bell encabezaron un comité para el proyecto SWEBOK (*Software Engineering Body of Knowledge*). Este comité tenía como finalidad establecer un criterio y una norma para la práctica profesional de la ingeniería del software bajo la cual pudieran ser basadas las decisiones dentro de la industria, certificación profesional y currícula educativa. Los patrocinadores del SWEBOK son la ACM, IEEE *Computer Society* y el SWECC (*Computer Society and Software Engineering Coordinating Committee*). La estimación del esfuerzo de desarrollo de software es considerada dentro del capítulo 8 del SWEBOK (área de conocimiento: Administración de la Ingeniería del Software, *Software Engineering Management*), parte B (Administración del Proceso/Proyecto, *Process/Project Management*), sección 2 Planeación, inciso 4 [48].

El SEI de CMU (donde el CMM fue creado) y la Administración Federal de Aviación de los EE. UU. (FAA), iniciaron un proyecto para mejorar las competencias de ingeniería de software de esta última en el personal técnico y administrativo. Entonces fue generado el *Software Engineering Body of Knowledge* (SWEBOK-SEI) donde la estimación del esfuerzo de desarrollo de software es considerada dentro del área de conocimiento 3.1 (Administración del Proyecto de Software), unidad 3.1.3 (*Project Forecasting*) [37].

En México, dentro de los modelos curriculares a nivel licenciatura de Informática y Computación elaborados por la Asociación Nacional de Instituciones de Educación en Informática (ANIEI) y el Instituto Nacional de Estadística, Geografía e Informática (INEGI) se incluye a la “Estimación de costo y tiempo global del desarrollo” dentro de la sexta (Programación e Ingeniería de Software, PI) de siete áreas de conocimiento en su sección 6.3 (Sistemas de Software) de la que se desprende una sección 6.3.1 (Análisis y diseño) y una subsección PI23 [4].

De igual forma, la importancia de la estimación del esfuerzo de desarrollo de software se refleja en dos libros de ingeniería de software de amplia circulación entre las universidades del país [106] [117].

### 1.1.5 El entrenamiento personal previo a la integración del equipo

*“Unless engineers have the capabilities provided by personal training, they cannot properly support their teams or consistently and reliably produce quality products”*

*Watts Humphrey, 2000 [42]*

Implícitamente, en el desempeño de un empleado dentro de una empresa se involucra un nivel aceptable en sus aspectos físicos, intelectuales y morales. Con estas características, puede entonces aspirar a integrarse a equipos con ese mismo nivel y una vez que han logrado ser un grupo sinérgico, entonces este equipo puede aspirar a colaborar con otros equipos dentro de la organización.

Esta secuencia individuo-equipo-organización está reflejada en la secuencia PSP-TSP-CMM (*Personal Software Process - Team Software Process - CMM*, Proceso Personal de Software - Proceso de Software en Equipo - CMM) que el SEI ha desarrollado.

Sin embargo, curiosamente esta secuencia de tres modelos de proceso no corresponde a su secuencia cronológica, es decir, primero apareció el CMM en 1987, luego el PSP en 1995 y posteriormente el TSP en 1997 [42][43]. Esto sucedió cuando Watts Humphrey decidió en 1989 aplicar los principios del CMM para escribir pequeños programas con la finalidad de convencer a los ingenieros de software de adoptar las prácticas que el proceso dictaba. Posteriormente fue contratado de tiempo completo por el SEI para este fin. Al cabo de tres años y tras al menos 15 versiones (con el uso de lenguajes como Pascal y C++, y alrededor de 25,000 líneas de código) Humphrey logró concluir que los principios de administración de procesos de Deming y Juran era aplicables al trabajo de ingenieros de software individualmente como lo eran a otros campos de la tecnología. De 1993 a 1994 el PSP fue enseñado en CMU y en el Instituto Tecnológico de Massachussets y en 1995 Humphrey publicó la versión final del PSP [42].

Asimismo, el SEI de CMU ha producido un modelo curricular para la ingeniería del software [37]. Este modelo incorpora dos niveles de educación: (1) ingeniería del software en lo pequeño (*software engineering in the small*) y (2) ingeniería del software en lo grande (*software engineering in the large*). El primer punto se refiere a la aplicación de principios de ingeniería de software al desarrollo de un producto en lo individual, mientras que el segundo punto el desarrollo en equipo.

Puede entonces inferirse que dentro de ese 90% de empresas emergentes que de acuerdo al PDIS existen en México, la estimación del esfuerzo de desarrollo de software debería comenzar desde un nivel personal desarrollando pequeños programas para que los desarrolladores puedan integrarse con menor problema a equipos de trabajo y posteriormente a organizaciones altamente maduras en sus procesos.

De acuerdo con la sección 1.1.3, el CMM es una descripción disponible de metas, métodos y prácticas necesarias para la industria de la ingeniería de software; por ello, las empresas deberían comprender el CMM y saber cómo aplicar sus principios. Los desarrolladores primero deberían conocer dichos principios individualmente. La pregunta sería, ¿cómo comprender el CMM a nivel personal dentro de un entorno de investigación? La

respuesta es haciendo uso del PSP. Doce de las dieciocho áreas de proceso clave del CMM están al menos parcialmente tratadas por el PSP. El proceso del PSP incluye planeación, desarrollo (diseño del algoritmo, revisión de diseño, codificación, revisión de código, compilación, y pruebas), además de la etapa final denominada *postmortem*. Por esto puede deducirse del presente párrafo, que las prácticas del PSP podrían auxiliar en la experimentación dentro de laboratorios de investigación en los que se deseen comparar técnicas de estimación a nivel personal.

## 1.2 Planteamiento del problema

*Software Effort Estimation is one of the three Great Challenges for Half-Century-Old Computer Science*

*Frederic Brooks, Jr. 2003 [17]*

Desde el Congreso convocado por la OTAN en 1968, evento en que nació la ingeniería de software como tal, la planeación de proyectos de software fue señalada como una de sus prioridades. Una de las actividades de la planeación es la estimación del esfuerzo de desarrollo, que es una de las actividades más críticas en la administración de proyectos de software [46].

Países como la India, Irlanda, Canadá, Australia y Singapur, sirven como ejemplo para el proyecto del gobierno y ubicar a México como líder en la industria del software de latinoamérica para el año 2010. El criterio de crecimiento del PDIS es con base en niveles del CMM.

Para desarrollar software de mejor calidad y ser competitivos al interior y exterior de nuestro país, las empresas se deben someter a la cultura de procesos; diversos modelos de procesos existen en el mundo; tres de los más representativos para México (CMM, SPICE y MoProSoft) consideran a la estimación del esfuerzo de software como una de las actividades necesarias. Asimismo, criterios y normas tanto internacionales como nacionales para la práctica profesional de la ingeniería del software en las que se basa la industria, las certificaciones profesionales y los modelos curriculares académicos (SWEBOK, ANIEI) consideran a la estimación del esfuerzo de desarrollo de software.

Por otro lado, con base en estudios que por años hizo Humphrey, se demostró que al tener un entrenamiento personal los ingenieros de software, podrían integrarse a equipos que de manera más confiable desarrollaran software de mejor calidad. Ese entrenamiento debería incluir prácticas de estimación del esfuerzo de sus programas.

De acuerdo con el CONACYT [112], más del 90% de las empresas de software mexicanas (catalogadas como *emergentes*) no tienen procesos formales para administrar proyectos.

En 1968, Alfred M. Pietrasanta del Instituto de Investigación de Sistemas de IBM mencionó lo siguiente: “*Quien espera una rápida y fácil solución para el problema de la polifacética estimación se va a decepcionar*”, esta predicción sigue siendo una realidad ahora como lo era entonces [51]. Incluso se ha mencionado que el tiempo de desarrollo no puede ser objetivamente predicho [61], pero al mismo tiempo es una realidad que existe la necesidad de

ello, por lo que encontrar alternativas se mantiene como una ocupación constante dentro del ámbito de la investigación.

Una cuidadosa comparación de los resultados de diversos enfoques es lo más adecuado para generar estimaciones más realistas, dado que ninguna técnica de estimación del esfuerzo de desarrollo de software es mejor para todas las situaciones, [12]. Expertos han sugerido que es apropiado usar más de un método para predecir el esfuerzo de desarrollo de software [114]. De hecho, el consenso de diversos enfoques de estimación usualmente produce mejores resultados que un solo modelo [123].

Se concluye entonces lo siguiente:

- a) Desde el origen de la ingeniería del software en 1968, la estimación del esfuerzo de desarrollo de software ha sido una actividad relevante. Incluso, esta ha sido referenciada por un reconocido investigador, como uno de los tres grandes desafíos de la ciencia computacional.
- b) La estimación del esfuerzo de desarrollo de software es considerada tanto en normas nacional (MoProSoft) e internacionales (CMM, SPICE) de procesos como en cuerpos de conocimiento para certificaciones y modelos curriculares (SWEBOK, ANIEI).
- c) El Programa para el Desarrollo de la Industria del Software, derivado del Plan Nacional de Desarrollo de México, tiene como referencia de crecimiento al CMM.
- d) Más del 90% de las empresas desarrolladoras de software en México carece de prácticas para administrar proyectos.
- e) Antes de integrarse a equipos, un profesionista del desarrollo de software debería tener un entrenamiento personal.
- f) El Proceso de Software Personal permite aplicar a nivel individual, al menos 12 de las 18 áreas clave de proceso del CMM.
- g) Con el antecedente de que ninguna técnica de estimación del esfuerzo de desarrollo de software es mejor para todas las situaciones, es importante proponer nuevos modelos que permitan comparar sus resultados y entonces generar estimaciones más realistas.

### 1.3 Justificación

Con base en la AMITI (Asociación Mexicana de la Industria de Tecnologías de Información), la meta del PDIS para el año 2010 (abastecer un mercado de 5,000 millones de dólares anuales, similar al nivel que alcanzó India en el año 2000) se fundamenta en niveles del CMM (Tabla 1) [112].

**Tabla 1.** Metas de crecimiento para el 2010 en la industria de *software* en México

Tipo	Número de empresas	Tamaño promedio (Empleados)	Personal ocupado Total	Facturación (miles de dólares)
CMM 5	3	4,000	12,000	576,000
CMM 4	10	1,500	15,000	720,000
CMM 3	37	750	27,750	1,332,000
CMM 2	120	250	30,000	1,440,000
CMM 1	300	70	21,000	1,008,000
Total	470		105,750	5,076,000

Por otro lado, además de la planeación y control del proyecto, las técnicas de estimación se usan para los siguientes propósitos dentro de las empresas: (a) presupuestación, (b) administración de riesgos y (c) análisis de mejoramiento de inversión [12] [121]. Las consecuencias de un esfuerzo deficientemente estimado dentro de las empresas son, entre otras: (a) deficiencia en la calidad de los productos, (b) clientes insatisfechos y (c) desarrolladores frustrados [50].

Barry Boehm, creador del modelo de estimación del esfuerzo de software COCOMO 81 y II, (que rara vez es omitido dentro de las referencias de cualquier artículo relacionado con la estimación del esfuerzo), muestra en uno de sus trabajos (Figura 1, ángulo inferior izquierdo), cómo él considera la necesidad de la investigación sobre mejores modelos para la estimación del software dentro del marco de la economía de la ingeniería del software [11].

La propuesta de un modelo de estimación de esfuerzo de software a nivel personal basado en lógica difusa, permitirá un recurso más para la comparación entre más modelos; asimismo, el modelo propuesto servirá para el entrenamiento individual que un ingeniero de software debería tener antes de aspirar a integrarse a equipos de desarrollo.

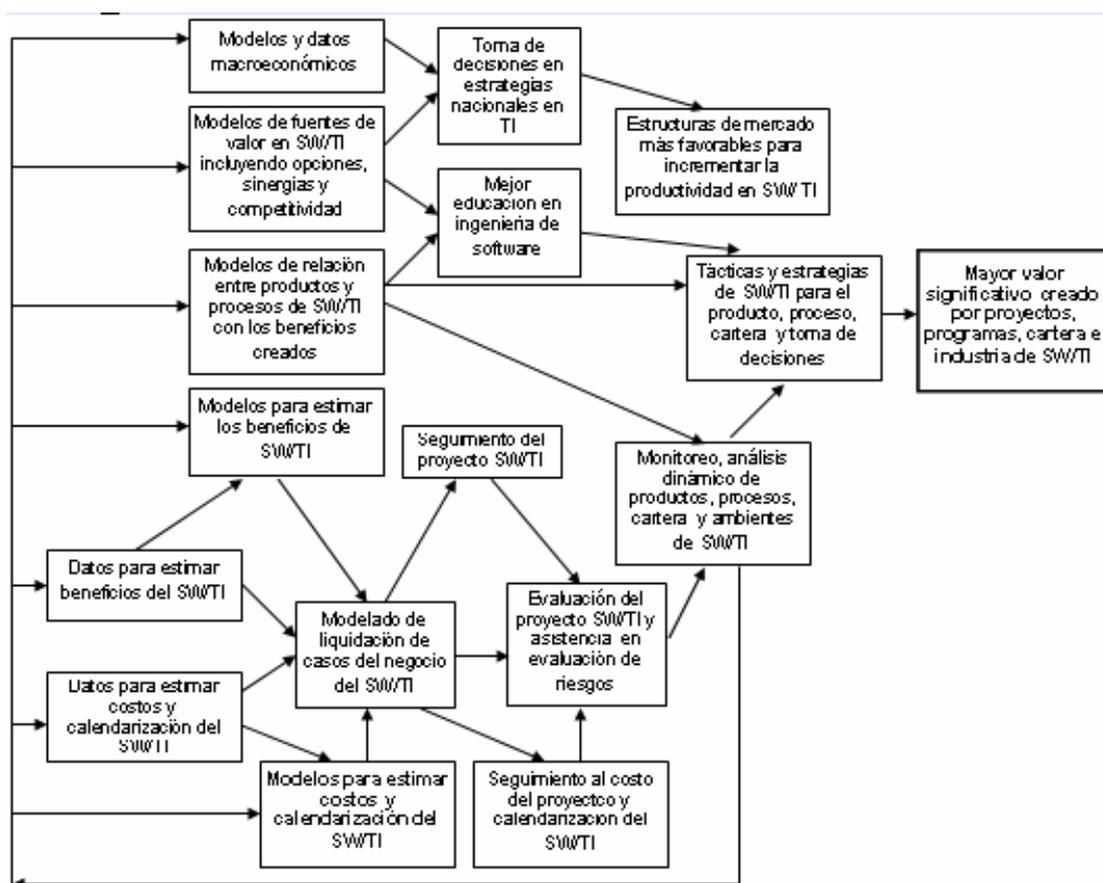


Figura 1. La investigación en la economía de la ingeniería del software [11]

## 1.4 Objetivos

### General:

Generar un modelo difuso con una precisión aceptable ( $MMER \leq 0.25$ ) integrado por reglas, conjuntos y funciones de membresía, derivadas tanto del tamaño como del tiempo de desarrollo, de programas elaborados con base en un proceso común y dentro de un diseño cuasiexperimental, con la finalidad de aportar tanto a las empresas desarrolladoras de software como al ámbito académico, una alternativa para la estimación del esfuerzo de desarrollo de software a nivel personal.

### Específicos:

- Proponer un modelo para la estimación del esfuerzo de desarrollo de software a nivel personal basado en lógica difusa, extensible a otras instancias del ciclo de vida del software.
- Proveer a la industria y a la academia de un modelo que permita iniciar la práctica de la estimación del esfuerzo a nivel personal y, en lo sucesivo, un mejor desempeño en equipos de trabajo dentro de una organización.
- Proponer un modelo que permita comparar sus resultados con otros modelos,
- Demostrar que la aplicación de la lógica difusa a la estimación del esfuerzo de desarrollo tiene resultados iguales o mejores que el modelo más empleado hasta ahora: la regresión estadística.
- Proponer un proceso para la generación de modelos basados en lógica difusa para la estimación del esfuerzo de desarrollo de software.

## 1.5 Método

### 1.5.1 Enfoque de la investigación

Existen dos enfoques de investigación: cualitativo y cuantitativo [35]:

**Cuantitativo.** Utiliza la recolección y el análisis de datos para contestar preguntas de investigación y probar hipótesis establecidas previamente, y confía en la medición numérica, el conteo y frecuentemente en el uso de la estadística para establecer con exactitud patrones de comportamiento en una población.

**Cualitativo.** Se utiliza primero para descubrir y refinar preguntas de investigación. A veces, pero no necesariamente, se prueban hipótesis. Con frecuencia se basa en métodos de recolección de datos sin medición numérica, como las descripciones y las observaciones. Por lo regular, las preguntas e hipótesis surgen como parte del proceso de investigación, el cual es flexible, y se mueve entre los eventos y su interpretación, entre las respuestas y el desarrollo de la teoría. Su propósito es reconstruir la realidad, tal y como la observan los actores de un sistema social previamente definido.

La investigación de esta tesis corresponde al enfoque cuantitativo respetando los siguientes diez pasos generales [35]: (1) concepción de la idea a investigar, (2) planteamiento del problema de investigación, (3) elaboración del marco teórico, (4) definición del alcance de

la investigación, (5) establecimiento de hipótesis, (6) selección del diseño apropiado para la investigación, (7) selección de la muestra, (8) recolección de datos, (9) análisis de datos y (10) presentación de resultados. En lo particular, esta tesis se basa en directrices de investigadores del área de la ingeniería del software [57] [107] [115]. Los pasos 1 y 2 han sido ya mencionados en las secciones 1.1 y 1.2 de este capítulo. A continuación se describe del cuarto al séptimo punto. El tercer punto es descrito en el capítulo 2, mientras que el octavo y noveno puntos se incluyen en el capítulo 3; el punto décimo se describe en el capítulo 4.

### 1.5.2 Definición del alcance de la investigación

El alcance de una investigación puede tener hasta cuatro niveles [35]:

**Exploratoria:** Se efectúan cuando el objetivo es examinar un tema o problema de investigación poco estudiado, del cual se tienen muchas dudas o no se ha abordado antes. Es decir, los estudios exploratorios sirven para familiarizarse con fenómenos relativamente desconocidos.

**Descriptiva:** Buscan especificar las propiedades, las características y los perfiles importantes de personas, grupos, comunidades o cualquier otro fenómeno que se someta a un análisis. Pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren.

**Correlativa:** Tienen como propósito evaluar la relación que exista entre dos o más conceptos, categorías o variables en un contexto particular (miden el grado de relación entre dos o más variables). Su utilidad es saber cómo se puede comportar un concepto o una variable conociendo el comportamiento de otras variables relacionadas. Se le atribuye un valor explicativo (siguiente nivel), pero parcial.

**Explicativa:** Están dirigidos a responder a las causas de los eventos y fenómenos físicos y sociales.

El alcance de esta tesis es exploratorio porque tras un extenso estudio de la aplicación de la lógica difusa a la estimación del esfuerzo de desarrollo basado en un proceso personal, no fue encontrado y además se infirió que resultaría útil para los planes de nuestro país.

Esta tesis es descriptiva, dado que se recolectaron datos de 52 desarrolladores que generaron sus 364 programas con base en un proceso de software personal.

Asimismo, este estudio es correlativo pues se midió y analizó la correlación entre las líneas de código y el esfuerzo de desarrollo de los programas.

Por último, esta investigación alcanza un nivel explicativo dado que se identifican las causas generales y especiales de los resultados de la aplicación del modelo difuso.

### 1.5.3 Formulación de hipótesis

Las hipótesis indican lo que se está tratando de probar y se definen como explicaciones tentativas del fenómeno investigado, formuladas a manera de proposiciones (sujetas a comprobación empírica); involucran dos o más variables, que son propiedades que pueden variar y cuya variación es susceptible de medirse. Existen cuatro tipos de hipótesis [35]:

1. **De investigación (o de trabajo):** Proposición tentativa acerca de las posibles relaciones entre dos o más variables.

2. **Nulas:** Inverso de las hipótesis de investigación.
3. **Alternativas:** Indican posibilidades alternas ante la hipótesis de investigación y nulas (sólo pueden formularse cuando efectivamente haya al menos otra posibilidad).
4. **Estadísticas:** Son exclusivas del enfoque cuantitativo y representan la transformación de las hipótesis de investigación, nulas y alternativas en términos estadísticos. Hay tres tipos de hipótesis estadísticas:
  - 4.1) **De estimación:** Evalúan la suposición de un investigador respecto al valor de alguna característica en una muestra de individuos u objetos y en una población.
  - 4.2) **De correlación:** traducen una correlación entre dos o más variables.
  - 4.3) **De diferencia de medias:** Compara una estadística entre dos o más conjuntos.

Con base en el planteamiento del problema de este estudio se derivan las siguientes hipótesis estadísticas (se colocan en el orden de esta investigación):

**a) De correlación:**

A partir de una población, la muestra extraída posee una correlación ( $r$ ) mayor o igual a 0.71 entre las líneas de código ( $x$ ) y el esfuerzo ( $y$ ), es decir, con un  $r^2 \geq 0.50$  (valor aceptable para un modelo de predicción [40]):

$$H_i: r_{xy} \geq 0.71$$

A partir de una población, la muestra extraída no posee una correlación ( $r$ ) mayor o igual a 0.71 entre las líneas de código ( $x$ ) y el esfuerzo ( $y$ ), es decir, con un  $r^2 \geq 0.50$ :

$$H_o: r_{xy} < 0.71$$

**b) De diferencia de medias:**

Tras la verificación de los modelos lineal ( $X_1$ ) y difusos ( $X_2, X_3, X_4$ ), existe diferencia significativa entre las medias de las MER de los cuatro modelos:

$$H_i: \bar{X}_1 \neq \bar{X}_2 \neq \bar{X}_3 \neq \bar{X}_4$$

Tras la verificación de los modelos lineal ( $X_1$ ) y difusos ( $X_2, X_3, X_4$ ), no existe diferencia significativa entre las medias de las MER de los cuatro modelos:

$$H_o: \bar{X}_1 = \bar{X}_2 = \bar{X}_3 = \bar{X}_4$$

**c) De estimación:**

Al término de la validación de los modelos, en la predicción de programas a nivel personal haciendo uso de un modelo difuso se obtiene una  $MMER \leq 0.25$ :

$$H_i: MMER \leq 0.25$$

Al término de la validación de los modelos, en la predicción de programas a nivel personal haciendo uso de un modelo difuso se obtiene una  $MMER > 0.25$ :

$$H_o: MMER > 0.25$$

### 1.5.4 Selección del diseño apropiado para la investigación

El término diseño se refiere al plan o estrategia concebida para obtener la información que se desea. Existen los dos diseños de investigación siguientes [35]:

1) **Experimental:** Un experimento es un estudio en el que se manipulan intencionalmente una o más variables independientes para analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes, dentro de una situación de control para el investigador.

Durante un experimento se debe cumplir el control o la validez interna. Deben entonces considerarse las fuentes de invalidación interna siguientes [35]:

- a) **Historia:** Acontecimientos que ocurren durante el desarrollo del experimento que pudieran afectar la variable dependiente y llegan a confundir los resultados.
- b) **Maduración:** Procesos internos de los participantes que operan como consecuencia del tiempo y que afectan los resultados del experimento tales como cansancio, hambre, aburrimiento o aumento en la edad.
- c) **Inestabilidad:** Poca o nula confiabilidad en las mediciones, fluctuaciones en las personas seleccionadas, o inestabilidad autónoma de las mediciones.
- d) **Administración de las pruebas:** Efecto que puede tener la aplicación de una prueba sobre las puntuaciones de pruebas consecuentes.
- e) **Instrumentación:** Cambios en los instrumentos de medición.
- f) **Regresión estadística:** Efecto provocado por una tendencia que los sujetos seleccionados, sobre la base de puntuaciones extremas, muestran al regresar, en pruebas posteriores, a un promedio en la variable en la que fueron seleccionados
- g) **Selección:** Elección de sujetos para los grupos del experimento, de tal manera que los grupos no sean equiparables.
- h) **Mortalidad experimental:** Diferencias en la pérdida de participantes.
- i) **Interacción entre selección y maduración:** Efecto de maduración que no es igual en los grupos del experimento, debido a algún factor de selección.

Los diseños experimentales son propios de la investigación cuantitativa y se dividen en:

- a) **Preexperimental:** Su grado de control es mínimo. No hay manipulación de la variable independiente. Tampoco hay una referencia previa de cuál era, antes del estímulo, el nivel que tenía el grupo en la variable dependiente, ni grupo de comparación. El diseño adolece por tener varios grupos de comparación. No es posible establecer causalidad con certeza ni se controlan las fuentes de invalidación interna.
- b) **Experimental puro o verdadero:** Manipulan intencionalmente una o más variables independientes, mide el efecto que la variable independiente tiene en la variable dependiente y cumplen con los requisitos para lograr el control y validez interna. La asignación de la muestra es al azar
- c) **Cuasiexperimental:** Manipulan deliberadamente al menos una variable independiente. Difieren de los puros o verdaderos en que en los cuasiexperimentales los sujetos no se asignan al azar, sino que dichos grupos ya estaban formados antes del experimento. Pero por lo demás son iguales, la interpretación es similar, las comparaciones son las mismas y los análisis

estadísticos iguales (salvo que a veces se consideran las pruebas para datos no correlacionados).

**2) No experimental:** No se manipulan deliberadamente las variables, consiste en observar fenómenos tal y como se dan en su contexto natural, para después analizarlos (no se construye alguna situación). Se clasifican en los siguientes dos tipos:

- a) **Transeccional:** Se recolectan datos en un solo momento, en un lapso único. Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado.
- b) **Longitudinal:** Analiza cambios a través del tiempo, en puntos o períodos, para hacer inferencias respecto al cambio, sus determinantes y sus consecuencias. Tales puntos o períodos por lo común se especifican de antemano en el enfoque cuantitativo, y se van determinando conforme avanza el estudio en el enfoque cualitativo.

En la investigación de esta tesis, no se trata de un diseño preexperimental dado que sí hay un control de la variable independiente y sí se controlan las fuentes de invalidación interna. No es un experimento puro o verdadero porque los sujetos no fueron elegidos al azar. Tampoco es un diseño no experimental, ya que se manipulan deliberadamente las variables al hacer uso de herramientas nuevas para todos los sujetos. Entonces se trata de un diseño cuasiexperimental debido a que los grupos ya estaban formados antes del experimento.

Las fuentes de invalidación interna en la presente investigación, se consideraron de la manera siguiente (de acuerdo a la definición y ejemplos en [35]):

- a) **Historia:** Fueron eliminados de la muestra aquellos programas que durante el experimento sus desarrolladores mostraron problemas personales (sin abandonar el curso) y que les impidieron tener una consistencia en su desempeño.
- b) **Maduración:** Fueron excluidos de la muestra tres de los siete programas que cada uno de los desarrolladores generó, dado que las prácticas que el proceso involucraba del primero al tercero fueron diferentes (a partir del cuarto, se llevó el mismo número de prácticas en cada programa). Se siguieron las siguientes fases del proceso de desarrollo en los siete programas (indicadas por el PSP [40]): planeación, diseño del algoritmo (en pseudocódigo o diagrama de flujo), codificación, compilación, pruebas y postmortem. Las fases de revisión de código y revisión de diseño fueron integradas a partir del tercero y cuarto programa respectivamente. Los siguientes cuadernos de registro y estándares fueron usados por todos los desarrolladores de la población: estándar de codificación, estándar de cuantificación de código, estándar de tipos de defectos, sumario del plan del proyecto, cuaderno de registro de tiempos, cuaderno de registro de defectos y propuesta de mejoramiento de proceso (estos cinco últimos son mostrados en el anexo A). A partir del tercer programa se empleó la forma de registro de pruebas (se pidió documentar sólo tres de ellas para establecer consistencia en el registro de tiempo en la fase de pruebas). Otra razón para excluir del primero al tercer programa fue que por primera vez los desarrolladores generaban programas con base en el PSP y debido a que las prácticas fueron diferentes en cada programa, las dudas impedían constantemente un registro fidedigno del tiempo.

- 
- c) **Inestabilidad:** Al término de cada programa, las mediciones de cada desarrollador eran constantemente revisadas para evitar errores similares en posteriores programas. Se observó la disciplina de cada desarrollador y fueron excluidos los programas de aquellos indisciplinados.
  - d) **Administración de las pruebas:** La retroalimentación constante por cada programa revisado permitió la confiabilidad de los registros de los datos en programas siguientes.
  - e) **Instrumentación:** Cada uno de los desarrolladores siguió el mismo proceso de desarrollo y empleó el mismo conjunto de formas de registro (mencionadas en el inciso *b* de esta sección).
  - f) **Regresión estadística:** Los valores atípicos y de influencia fueron identificados mediante una gráfica de dispersión líneas de código-esfuerzo. Se encontró la causa especial y sólo con esta condición fueron eliminados de la muestra.
  - g) **Selección:** Los desarrolladores habían tomado un curso del lenguaje de programación empleado. No fue relevante un tiempo de experiencia mayor dado que los programas no exigían complejidad alta.
  - h) **Mortalidad experimental:** No fueron tomados en cuenta para la muestra los programas de aquellos desarrolladores que por aspectos personales abandonaron el curso,.
  - i) **Interacción entre selección y maduración:** La preparación académica de los desarrolladores rebasó como mínimo el 50% de estudios de licenciatura y el término de la licenciatura como máximo. Los programas no exigían más conocimiento que aquél obtenido en este lapso académico.

Asimismo, para evitar algún sesgo, se cuidó que los sujetos que participaran en el experimento desconocieran las hipótesis de este estudio (como se recomienda en [35]); incluso, con frecuencia se distrajo de los verdaderos propósitos del experimento.

### 1.5.5 Selección de la población y muestra

Una población es un conjunto de todos los casos que concuerdan con determinadas especificaciones. Una muestra es un subconjunto de esa población.

Existen dos tipos de muestras [35]:

- a) **No probabilísticas o dirigidas:** La elección de sus elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación. La muestra consiste de sujetos con la vaga esperanza de que serán representativos a la población determinada. Por ello, para fines deductivos-cuantitativos, donde la generalización o extrapolación de resultados hacia la población es una finalidad en sí misma, las muestras dirigidas en este sentido implican ciertas desventajas. Existen los siguientes tipos:
  - **Sujetos voluntarios:** El investigador elabora conclusiones a partir de de sujetos que llegan a sus manos en forma casual;

- **Expertos:** Cuando la opinión de un conjunto de gente con determinada pericia es necesaria;
- **Sujetos-tipo:** Grupos con determinadas características en común;
- **Por cuotas:** Conforme se obtienen sujetos, se van cumpliendo cuotas de porcentajes de ciertas variables predefinidas;
- **Estudios de caso:** Constan de una unidad (caso) de análisis (persona, pareja, familia, sistema, organización, comunidad, municipio, estado o nación, por ejemplo).

b) **Probabilísticas:** Todos los elementos de la población tienen la misma probabilidad de ser escogidos (mediante aleatoriedad).

La muestra de esta investigación no es probabilística debido a que los programas fueron desarrollados por programadores no elegidos aleatoriamente y por tanto no tuvieron la misma probabilidad de ser elegidos.

Entonces es una muestra no probabilística o dirigida. Dentro de esta clasificación no es del tipo sujetos voluntarios puesto que no fueron desarrollando sus programas, si bien no forzadamente, tampoco con una voluntad propia; no es del tipo expertos dado que no se identificó a expertos en el área para solicitarles alguna opinión; tampoco es por cuotas pues los resultados del experimento no eran condición de término de la investigación; y no se trató de un estudio de caso, debido a que los desarrolladores pertenecían a diferentes dependencias fueran de la academia, industria del software o gobierno. Por exclusión puede inferirse que se trata de una muestra no probabilística sujetos-tipo debido a que los desarrolladores deberían tener como característica común haber tenido al menos un curso del lenguaje de programación imperativa (en oposición a declarativa [5] ) con el que codificarían sus programas. El nivel de estudios no fue mayor debido a que los programas no lo exigían.

## 1.6 Organización del documento de tesis

En su primera parte este trabajo ha comenzado con el contexto del tema central de esta tesis. Se ha mencionado que desde el origen de la ingeniería de software en 1968 se ha manifestado la necesidad de la planeación de proyectos de software, área clave que requiere una estimación del esfuerzo de desarrollo de software y actividad que es considerada tanto en modelos de proceso de software como curriculares internacionales y de México. Así mismo se ha mencionado que para estimaciones más precisas es mejor comparar los resultados de más de un modelo de estimación.

De igual forma, se ha resaltado la importancia que el software representa para la economía de diversos países.

México ha elaborado un plan para la industria del software al 2010 y el criterio de su progreso se basa en niveles del CMM. Se justificó que para fines de la investigación, el PSP (que aplica al menos 12 de las 18 KPA del CMM) resulta un recurso adecuado.

En este capítulo se ha definido el objetivo de este estudio: el generar un sistema difuso para la estimación del esfuerzo de desarrollo de software a nivel personal y se ha descrito por qué se trata de una investigación cuantitativa con un alcance explicativo. Se han formulado tres hipótesis de investigación (con sus correspondientes nulas), se ha explicado por qué se trata de un diseño de investigación cuasiexperimental con base en una muestra no

probabilística y se han descrito las diversas consideraciones en las fuentes de invalidación interna en la presente investigación.

El capítulo 2 de esta tesis está dedicado al marco teórico comenzando con el proceso de estimación de un proyecto de software, dentro del cual se identifica aquel dedicado a la estimación del esfuerzo de desarrollo. Posteriormente, se describe el PSP incluyendo sus fases del proceso, estándares y cuadernos de registro. Más adelante se definen las tres principales clasificaciones de técnicas para estimar el esfuerzo de desarrollo de software: juicio experto, modelos algorítmicos y aprendizaje de máquina (la tercera de ellas más reciente en su aparición en comparación con las otras dos) y posteriormente plantea la necesidad de comparar los resultados de más de una técnica; expone los criterios que son utilizados por la mayoría de los investigadores de esta área para comparar la precisión de técnicas y son descritas las diversas mediciones del software.

Una sección del capítulo 2 es dedicada a los conceptos estadísticos utilizados para generar la muestra, el modelo lineal y la comparación entre los resultados de los modelos. Una penúltima sección se destina para definiciones y notación de la lógica difusa.

Este segundo capítulo finaliza con un análisis de artículos encontrados involucrando a la estimación del esfuerzo de desarrollo, prácticas personales y lógica difusa.

Un tercer capítulo se destina a describir el proceso general para la generación, verificación y validación de modelos. Se procede a describir con detalle la recolección de datos a partir de los cuales se generan los modelos; entonces es llevada a cabo la comprobación de la adecuación (verificación) de los modelos lineal y difusos.

El cuarto capítulo está dedicado a los resultados de la aplicación de los modelos de estimación a datos nuevos, es decir, a la validación de los modelos.

Un quinto y último capítulo se dedica a las conclusiones, recomendaciones, trabajo futuro y al legado de esta investigación.

## Capítulo 2

### Marco teórico

#### 2.1 El Proceso de estimación

*When you lie about the future, that's called optimism, and it is considered a virtue. Technically speaking you can't lie about the future because no one knows what will happen. When you apply this unique brand of optimism (not lying!) at work, that's called forecasting*

*Scott Adams. 2002 [92]*

La estimación es necesaria por lo siguiente [121]:

- El objeto a estimar es inaccesible;
- El objeto a estimar aún no existe.

Se ha comentado que la estimación es uno de los desafíos de la ciencia computacional. El software es especialmente difícil de estimar por lo siguiente [121]:

- Los requerimientos son difíciles de establecer con precisión;
- El producto es esencialmente invisible hasta que éste haya sido finalizado;
- El producto es difícil de medir (por su naturaleza intangible);
- La aceptación del producto depende del gusto del cliente.

El proceso de estimación del software consiste de las fases que se muestran en la Figura 2 [1]. La fase en la que el presente estudio está enfocada es en aquella denominada “Estimación del esfuerzo“ definida como la cantidad de tiempo requerido por un conjunto de personas para diseñar, codificar, probar y ejecutar otras actividades de apoyo relacionadas directamente con la obtención de un producto de software. Es importante diferenciar a esta etapa de aquellas que le suceden y preceden. Estimar el tamaño es un tema igualmente amplio que correspondería a una investigación con un alcance significativo; incluso existen investigaciones relacionadas con el tamaño del software con base en lógica difusa [77]. Los modelos que se generarán y validarán en este estudio emplearán como datos de entrada los tamaños reales. Es decir, el empleo de los modelos generados asume una estimación altamente precisa del tamaño (al igual que el modelo COCOMO 81 [9] asume que se ha tenido una exitosa obtención de requerimientos cuando se usan sus ecuaciones no lineales de estimación del esfuerzo). Como se observa en la Figura 2, diversos rubros pueden ser estimados dentro del ámbito de la estimación de proyectos [60] [121]:

- Tamaño (líneas de código o puntos de función);
- Esfuerzo (meses-persona);
- Costos (en pesos o dólares);
- Duración (tiempo calendario);
- Recursos de computadora;
- Personal (número);
- Calidad (defectos).

Específicamente, el proceso de estimación del esfuerzo incluye los siguientes pasos [83] (mismos que se siguen en esta investigación):

1. Obtención de datos a partir de proyectos anteriores
2. Generación de modelos de estimación
3. Evaluación de qué tan efectivos con los modelos

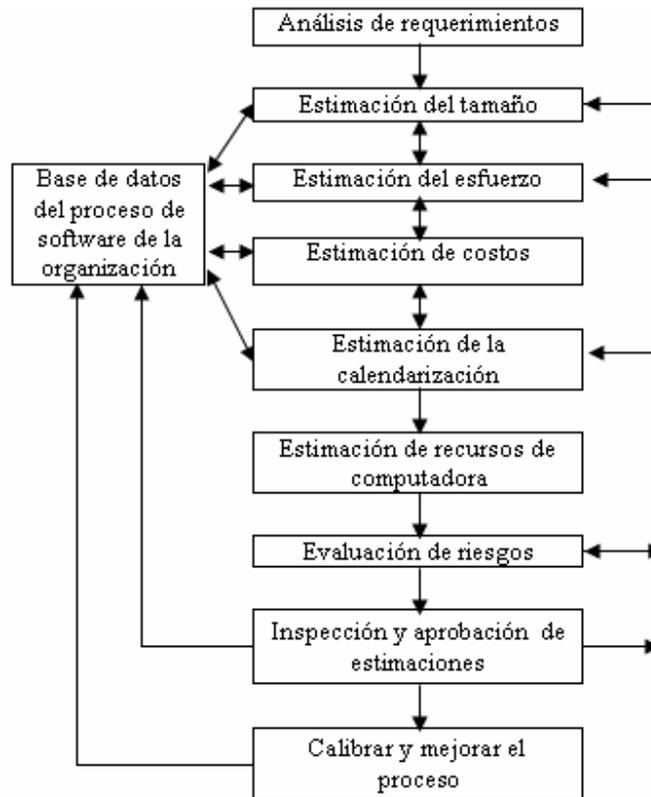


Figura 2. Proceso para la estimación del software

## 2.2 Proceso personal de software

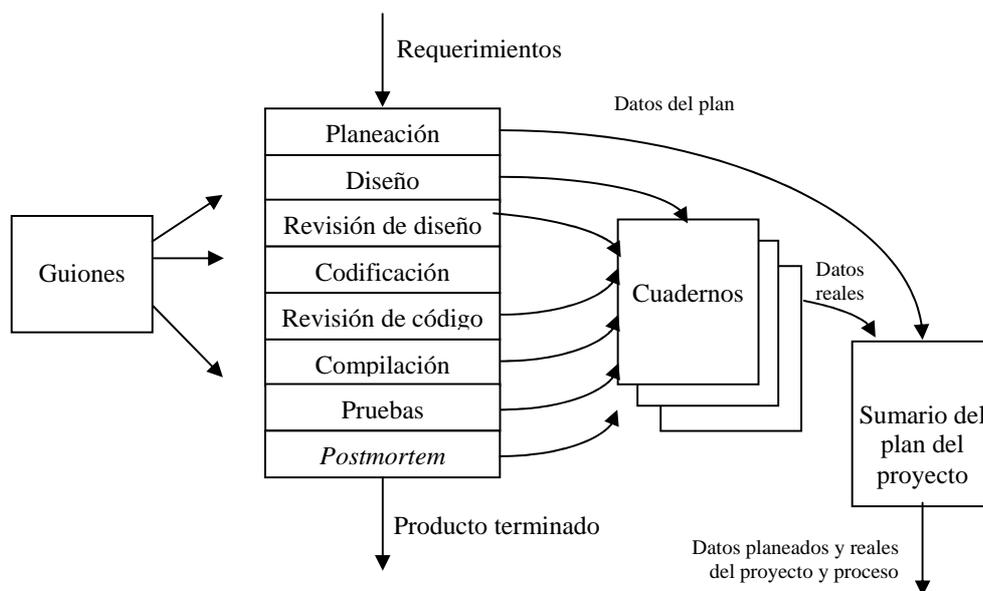
*“The PSP’s role is to help you a better software engineer. It is a powerful tool you can use in many ways, for example, to manage your work, assess your talents, and/or build your skills. It can help you plan better, track your performance precisely, and measure the quality of your products”.*

Watts Humphrey, 1995 [40]

El PSP fue diseñado para ayudar a los ingenieros de software a hacer bien su trabajo y muestra cómo aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona

métodos detallados de planificación y estimación, muestra a los ingenieros cómo controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo [41].

La estructura del PSP es mostrada en la Figura 3 [41], misma que incluye las ocho fases desde la planeación hasta el *postmortem*. Asimismo, involucra los guiones, cuadernos de registro (como aquellos de tiempos y defectos), formas (de revisiones de código y diseño, sumario del plan del proyecto), así como estándares (de codificación, defectos y cuantificación de líneas de código).



**Figura 3.** Proceso del PSP

La explicación de cada una de las ocho fases del proceso es la siguiente:

- 1) **Planeación:** Entendimiento de los requerimientos del programa y tiempo de estimación de fases siguientes (diseño a postmortem).
- 2) **Diseño:** Algoritmo representado, ya sea en diagrama de flujo o pseudocódigo
- 3) **Revisión de diseño:** Con base en los defectos registrados en programas previos a los que se les atribuye su introducción a la fase de diseño, se revisa que en el programa recientemente diseñado no se hayan cometido.
- 4) **Codificación:** Traducción del diseño a código (respetando estándar de codificación).
- 5) **Revisión de código:** Con base en los defectos registrados en programas previos a los que se les atribuye su introducción a la fase de codificación, se revisa que en el programa recientemente diseñado no se hayan cometido.
- 6) **Compilación:** Corrección de defectos detectados por el compilador.
- 7) **Pruebas:** Corrección de defectos no detectados por el compilador. Seguridad de que el programa cumple con los requerimientos.
- 8) **Postmortem:** Registro de datos reales en el Sumario del Plan del Proyecto.

El PSP permite reunir tres unidades de medida del software: (1) tiempo por fases (de la planeación al postmortem), (2) los tamaños de los programas que se desarrollan y (3) la calidad de los productos [41]:

**Tiempos por fases:** Son registrados en el cuaderno de registro de tiempos, donde se registra la fecha en que se lleva a cabo cada fase, su hora de inicio y su hora de fin, así como algún tiempo de interrupción (si la hubo). El desarrollo, de acuerdo con su definición dentro del ámbito del software [106], involucra aquellas actividades destinadas al diseño, codificación y prueba (para el caso del PSP involucra entonces el diseño, revisión de diseño, codificación, revisión de código, compilación y pruebas). Por ello, en este estudio se excluyen los tiempos de planeación y postmortem.

**Tamaño del programa:** Dado que el tiempo que toma desarrollar un programa es ampliamente determinado por su tamaño, primero se estima el tamaño del programa que el desarrollador desea planear. Las líneas de código (LDC) es la principal medida de tamaño del PSP; una medida puede ser considerada cuando cuando provea una aceptable correlación entre el tiempo de desarrollo y el tamaño del programa. Existe la siguiente clasificación de LDC:

- 2.1) **Base:** Cuando un programa (o programas) es mejorado, la LDC base es el tamaño de la versión del programa original antes que cualquier modificación le sea hecha.
- 2.2) **Adicionado (Nuevo):** Es aquel código escrito para un nuevo programa o adicionado a un programa base.
- 2.3) **Modificado (Cambiado):** Es aquel código base que es cambiado.
- 2.4) **Eliminado:** Es aquel código base que es borrado.
- 2.5) **Nuevo y Cambiado (NyC = Adicionado + Modificado):** Cuando el software es desarrollado, toma mucho más tiempo el adicionar o modificar una LDC que eliminarla o reutilizarla. Por ello, el PSP considera a las NyC para estimar el tamaño.
- 2.6) **Reusado:** Es el código tomado de alguna biblioteca y es usado sin modificación alguna.

**Medición de la calidad:** Son medidas en defectos introducidos. Para administrar los defectos se registran en qué fases son introducidos y corregidos, así como el tiempo que se tomó la corrección. Cada defecto es clasificado dentro de un estándar de defectos.

## 2.3 Técnicas de estimación

*Estimation models, procedures, and techniques are essential components of the software engineering discipline. As the field changes, the techniques of estimation must, of necessity, change.*

*Boehm and Fairley. 2000 [10]*

La ingeniería de software se preocupa por construir sistemas de software y productos dentro de las limitaciones de tiempo, recursos, tecnología, calidad y consideraciones del negocio. Los modelos, procedimientos y técnicas de estimación son componentes esenciales de la disciplina de la ingeniería de software.

Diversas técnicas de estimación han sido propuestas e investigadas durante los pasados 30 años [16] [60] [87]. Las investigaciones están dirigidas a (1) determinar cual técnica tiene la mayor precisión para predecir el esfuerzo y (2) proponer nuevas o combinadas técnicas que pudieran proveer mejores estimaciones.

Las técnicas de estimación han sido clasificadas de diversas formas. Entre ellos Boehm (uno de los investigadores más representativos de la estimación del software y creador del COCOMO 81 y II [9]) y Chulani [12] las clasifican en (a) basadas en el modelo, (b) basadas en la pericia, (c) orientadas al aprendizaje y (d) compuestas (incluyendo al enfoque bayesiano [20] [21] [22]). Shepperd [114] las clasifica en algorítmicas y no algorítmicas. El autor de esta tesis fue coautor de un artículo proponiendo un proceso involucrando diversas técnicas de estimación [68].

Las técnicas pueden encontrarse dentro de las siguientes tres categorías [87]:

### 1) Juicio experto:

Técnica ampliamente utilizada para derivar estimaciones basadas en la pericia de expertos en proyectos similares. El significado de derivar una estimación de este tipo no es explícita y no es repetible. Sin embargo, aunque siempre es difícil de cuantificar, el juicio experto puede ser una herramienta de estimación efectiva por si misma o bien ajustando un factor dentro de un modelo algorítmico. Aunque la estimación experta es comúnmente usada, es probable que no sea por su precisión; de hecho, parece ser tan imprecisa como el uso de modelos formales. Sin embargo, el juicio experto puede ser especialmente útil y frecuentemente la única opción para compañías con ya sea deficiente experiencia documentada de proyectos desarrollados o bien por tener recursos limitados para el proceso de estimación [92]. El término *estimación experta* no está claro y cubre un amplio rango de enfoques de estimación. Una característica común es, sin embargo, que los procesos *intuitivos* constituyen la mayor parte de la estimación [50]. Trabajos encontrados con la estimación del esfuerzo con base en esta técnica son los siguientes: [89] [90] [91] [92] [93] [100] [101] [109].

### 2) Modelos algorítmicos:

A la fecha son los más populares en la literatura [31]; intentan representar la relación entre el esfuerzo y una o más características del proyecto. El principal conductor de costo en tales modelos es usualmente tomado del tamaño del software (líneas de código, por ejemplo). Su forma general es una ecuación de regresión lineal como la usada por Kok [59], o no lineal como aquellas del COCOMO 81 [9] y COCOMO II [8]. El autor de la presente tesis publicó un artículo aplicando regresión lineal simple y múltiple para la estimación del esfuerzo de desarrollo de módulos de programas; los datos fueron obtenidos de un curso oficial recibido de PSP [76]; asimismo, es coautor de un trabajo más en el que se calibra la ecuación (no lineal) del esfuerzo del COCOMO 81 a un entorno local; los datos fueron obtenidos de proyectos de un centro de desarrollo de software [67].

### 3) Aprendizaje de máquina o no algorítmicos:

Estas técnicas han sido usadas en años recientes como complemento o alternativa al juicio experto y modelos algorítmicos [79] [118] [128]. Entre estos aparecen las redes neuronales [13] [14] [45] [52], razonamiento basado en casos [26] [27] [53] [56] [83] [86] [87] (la estimación por analogía es un tipo de éste [84] [114]), programación genética [18]

[25] [113] y la lógica difusa (referencias a esta última se mencionarán con más detalle en la sección 2.7). El autor de esta investigación es coautor de un trabajo que emplea una red neuronal para la estimación del esfuerzo de software [69].

Existen diversos trabajos que tratan específicamente sobre la comparación de modelos existentes: [15] [16] [19] [36] [55] [81] [82] [85] [123].

La estimación se puede ver desde tres perspectivas diferentes [2]: (1) el problema de estimación (¿qué etapa del proceso se estimará: desarrollo, mantenimiento, etc.?), (2) el problema particular (¿qué producto se estimará?) y (3) la técnica de estimación a usar dentro de las tres mencionadas categorías generales.

### 2.3.1 Necesidad de comparar técnicas

*“Experience has shown that there does not exist a best prediction technique outperforming all the others in every situation”*

*Ali Idri, et al. [47]*

Algunos investigadores han encontrado que la estimación por analogía genera mejores resultados que la regresión, mientras que otros han reportado resultados opuestos [47]. Por lo tanto, ningún método o modelo debería preferirse sobre los otros. La clave está entonces en usar una variedad de métodos y herramientas y luego investigar las razones del por qué las estimaciones provistas por una, podrían diferir significativamente de aquellas provistas por otra [12].

### 2.3.2 Criterios de evaluación para la comparación de técnicas

*“The MMRE, is probably the most widely used evaluation criterion for assessing the performance of competing software prediction models. It seems obvious that the purpose of MMRE is to assist us to select the best model. We have performed a simulation study demonstrating that MMRE does not select the best model. The consequences are dramatic for a vast body of knowledge in software engineering. The implications of this finding are that the results and conclusions on prediction models over the past 15-25 years are unreliable and may have misled the entire software engineering discipline.”*

*Tron Foss, 2003 [32]*

Un criterio común para la evaluación de modelos de estimación es la magnitud relativa del error (*Magnitud of Relative Error, MRE*) [15].

Otra medida semejante a la MRE ha sido propuesta, siendo esta la magnitud del error relativo a la estimación (*Magnitude of Error Relative to the Estimate, MER*). Intuitivamente es

preferible que la MRE dado que la MER mide el error relativo a la estimación. La MER usa el *Esfuerzo Estimado<sub>i</sub>* como denominador [58].

Los valores de la MRE y MER se calculan para cada observación (programa) *i* cuyo esfuerzo es estimado. Una agregación de la MRE y MER sobre múltiples observaciones (*N*), se puede obtener a través de la media de las MRE (MMRE) y de las MER (MMER).

Las ecuaciones para los cuatro criterios mencionados son los siguientes:

$$MRE_i = \frac{| \text{Esfuerzo Real}_i - \text{Esfuerzo Estimado}_i |}{\text{Esfuerzo Real}_i}$$

$$MER_i = \frac{| \text{Esfuerzo Real}_i - \text{Esfuerzo Estimado}_i |}{\text{Esfuerzo Estimado}_i}$$

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i$$

$$MMER = \frac{1}{N} \sum_{i=1}^N MER_i$$

Sin embargo, la MMRE y MMER son sensibles a predicciones individuales de las MRE o MER con valores grandes. Por ello, una medida menos sensible a valores extremos se considera en este estudio, siendo la mediana de los valores de las MRE y de las MER de las *N* observaciones. Su notación es MdMRE y MdMER respectivamente [16].

Un criterio complementario es el nivel de predicción *l*.  $\text{Pred}(l) = k/N$ , donde *k* es el número de observaciones donde la MRE o MER es menor o igual a *l*, y *N* es el número total de observaciones (programas desarrollados). De esta forma,  $\text{Pred}(0.25)$  da el porcentaje de programas que fueron estimados con una MRE o MER menor o igual que 0.25.

En general, la precisión de una técnica de estimación es proporcional a  $\text{Pred}(l)$  e inversamente proporcional a la MMRE y a la MMER.

Como referencia, una  $MMRE \leq 0.25$  es considerada como aceptable para modelos de predicción del esfuerzo [24]. Asimismo, un estudio reciente ha demostrado que la MER y MMER son medidas más adecuadas que la MRE y MMRE que han sido empleadas probablemente en forma errónea por varios lustros dentro de la investigación [32]. En este estudio se calculan ambos, dado que las MRE y MMRE han sido las más empleadas y por lo tanto facilitaría la comparación de los resultados de este estudio con otros previos. Se tomará con sus debidas proporciones este mismo valor como referencia para el grado de aceptación de este estudio ( $MMER \leq 0.25$ ), dado que no se encontró un criterio similar de aceptación para la MMER.

## 2.4 Medición del software

*“To measure software for scientific purposes, we will be concerned with two fundamental measurement principles: (1) The measurements must be reproducible, and (2) The attributes being measured must be valid”.*

*Munson, 2003 [96]*

Las razones para medir al software son tres [121], la última de ellas es el tema de central de esta tesis:

- Controlar los proyectos;
- Comprender los procesos;
- Estimar proyectos futuros con mayor precisión.

La aplicación más común de las métricas de software es desarrollar modelos que predigan el esfuerzo que será requerido para completar ciertas etapas de desarrollo de un sistema de software [30] [34] [88].

El tamaño del código fuente fue de interés como unidad de medida desde el principio de la historia del desarrollo de software. De esta manera Halstead, Chrysler y Boehm emplearon tanto análisis teórico como empírico para construir modelos iniciales para el tamaño de código fuente; a pesar de la disponibilidad de un amplio rango de medidas para el tamaño del producto software, las líneas de código fuente permanecen a favor en muchos modelos [77] [87]. El experimento dentro de la investigación de esta tesis considera a las líneas de código como variable independiente.

Las características de una buena medida de tamaño son las siguientes [121]:

- Está correlacionada con el esfuerzo de desarrollo del software consumido por los desarrolladores;
- Es independiente de la tecnología usada y del proceso de desarrollo;
- Puede ser estimada al inicio del proyecto;
- Puede ser estimada fácilmente;
- Puede ser cuantificada con precisión después de su uso.

Con todas estas características cuenta la medida empleada en este estudio.

## 2.5 Estadística

*“One of the most powerful tools available to a software engineer is that of statistics”*

*Munson, 2003 [96]*

### 2.5.1 Correlación y coeficiente de determinación

La correlación ( $r$ ) es el grado en que dos conjuntos de datos están relacionados. El valor de la correlación  $r$  varía de -1.0 a +1.0. El coeficiente de correlación entre líneas de código y esfuerzo, por ejemplo, puede ser calculado utilizando la siguiente ecuación [125]:

$$r = \frac{n \left[ \sum_{i=1}^n (LDC_i \cdot E_i) \right] - \left[ \left( \sum_{i=1}^n LDC_i \right) \left( \sum_{i=1}^n E_i \right) \right]}{\sqrt{\left[ n \left( \sum_{i=1}^n LDC_i^2 \right) - \left( \sum_{i=1}^n LDC_i \right)^2 \right] \left[ n \left( \sum_{i=1}^n E_i^2 \right) - \left( \sum_{i=1}^n E_i \right)^2 \right]}}$$

Donde  $n$  es el número de pares de observaciones,  $LDC$  son las líneas de código y  $E$  es el esfuerzo de desarrollo.

El coeficiente de determinación ( $r^2$ ) es la proporción de la variación en los valores observados en la variable de respuesta que es explicada por la regresión. El valor de  $r^2$  es obtenido usando la siguiente ecuación [125]:

$$r^2 = \frac{SCR}{SCT}$$

Donde:

SCR: Suma de Cuadrados de la Regresión. Es la variación en los valores observados de la variable de respuesta que es explicada por la regresión:

$$SCR = \sum_{i=1}^n (y'_i - \bar{y})^2$$

SCT: Suma de Cuadrados Totales. Es la variación en los valores observados de la variable de respuesta:

$$SCT = \sum_{i=1}^n (y_i - \bar{y})^2$$

## 2.5.2 Regresión lineal simple y múltiple

*“Linear statistical begins with a linear hypothesis and we will next need collect some data. Note the order: first we will postulate the model and then we will get the data”*

*Munson, 2003 [96]*

La mayoría de los métodos usados para desarrollar modelos de predicción son derivados a partir de la estadística inferencial. Entre sus ventajas está la relativa simplicidad en su formulación y su teoría de la probabilidad. Una línea recta bajo el modelo de mínimos cuadrados intenta encontrar la línea que minimice el error en la relación entre variable dependiente e independiente. La estructura de esta línea está normalmente expresada en forma de una ecuación.

La regresión lineal simple considera la relación que existe entre una variable dependiente y una independiente. Cualquier forma de regresión lineal es generalmente precedida por el uso de gráficas de dispersión y análisis de correlación para primero intuitivamente, y luego cuantitativamente, determinar el potencial de las relaciones que pudieran existir en los datos.

Cuando dos conjuntos de datos están fuertemente relacionados, es posible usar un procedimiento de regresión lineal para modelar esta relación. La ecuación de regresión lineal usando mínimos cuadrados y teniendo al esfuerzo como variable dependiente y a las líneas de código como independiente, es la siguiente:

$$E' = a + b(LDC)$$

Donde:

$$b = \frac{n \left[ \sum_{i=1}^n (LDC_i \cdot E_i) \right] - \left( \sum_{i=1}^n LDC_i \right) \left( \sum_{i=1}^n E_i \right)}{n \left( \sum_{i=1}^n LDC_i^2 \right) - \left( \sum_{i=1}^n LDC_i \right)^2} \quad a = \frac{\sum_{i=1}^n E_i}{n} - b \frac{\sum_{i=1}^n LDC_i}{n}$$

Por otra parte, una ecuación lineal con más de una variable independiente se le denomina múltiple. Por ejemplo, una ecuación con dos variables independientes podría ser expresada como:

$$y = b_0 + b_1x_1 + b_2x_2$$

Donde  $b_0$ ,  $b_1$  y  $b_2$  son constantes;  $x_1$  y  $x_2$  son variables independientes, y  $y$  es la variable dependiente.

Los valores de  $b_0$ ,  $b_1$  y  $b_2$  de la ecuación de regresión múltiple pueden ser obtenidos resolviendo el siguiente sistema de ecuaciones lineales:

$$\begin{aligned} \sum y &= nb_0 + b_1(\sum x_1) + b_2(\sum x_2) \\ \sum x_1y &= b_0(\sum x_1) + b_1(\sum x_1^2) + b_2(\sum x_1x_2) \\ \sum x_2y &= b_0(\sum x_2) + b_1(\sum x_1x_2) + b_2(\sum x_2^2) \end{aligned}$$

Para probar lo significativo de los coeficientes de las ecuaciones de regresión se lleva a cabo un análisis de varianza o ANOVA [3]:

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F
Modelo (media)	SCR	1	CMR	CMR / CME
Residual (error)	SCE	n - 2	CME	
Total	SCT	n - 1		

Donde:

CMR: Cuadrado medio debido a la regresión

$$CMR = \frac{SCR}{\text{grados de libertad de la regresión}}$$

CME: Cuadrado medio del error

$$CME = s^2 = \frac{SCE}{(n-2)}$$

SCR: Suma de cuadrados de la regresión

$$SCR = \sum_{i=1}^n (y_i' - \bar{y})^2$$

SCE: Suma de cuadrados debida al error

$$SCE = \sum_{i=1}^n (y_i - y')^2$$

SCT: Suma de cuadrados total

$$SCT = SCR + SCE$$

### 2.5.3 Valores atípicos y de influencia

*“Para el tratamiento de las observaciones inusuales debe tenerse en mente que probablemente una de ellas pueda ser la joya entre la piedras y no la piedra entre las joyas, debido a que alguna de ellas podría proporcionar información valiosa”.*

Weimer, 2002 [124]

Es importante identificar los valores inusuales dentro de un diagrama de dispersión que correlaciona la(s) variable(s) independiente(s) y la dependiente. Estos valores pueden afectar los resultados de la regresión. Debe entonces identificarse cada uno de ellos y encontrar la causa especial que justifique el retirarlo de la muestra, es decir, debe comprobarse que no se haya cometido un error al recopilar o registrar los datos.

Un valor atípico es una observación que no se ajusta a la tendencia que muestran los datos restantes, mientras que un valor de influencia (que también puede ser un valor atípico) es el alejamiento extremo del valor de la observación de la variable independiente [3]:

### 2.5.4 Pruebas para comparación de medias

Hay dos pasos principales cuando se usa un modelo de estimación [94]:

**1) Comprobación de la adecuación del modelo (verificación):** Debe determinarse si el modelo es adecuado para describir los valores observados. En este estudio los cuatro modelos (uno de regresión lineal y tres difusos) son aplicados a los datos originales de los programas: la entrada es el valor real de NyC y la salida el esfuerzo real en minutos. Entonces para cada programa son calculadas las MRE, MMRE, MER y MMER.

Posteriormente el conjunto de las MRE y MER de un modelo son comparados con los conjuntos del otro modelo. Ulteriormente, se hace un análisis de varianza (ANOVA) con prueba del estadístico  $F$  para la comparación de los cuatro modelos [3]:

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F
Tratamientos	SCT	$k - 1$	CMT	CMT / CME
Error	SCE	$n_T - k$	CME	
Total	STC	$n_T - 1$		

Donde:

$n_T$ : número de tratamientos

$k$ : número de poblaciones

CMT: Cuadrado medio debido a los tratamientos:

$$CMT = SCT / (k-1)$$

CME: Cuadrado medio debido al error

$$CME = SCE / (n_T - k)$$

SCT: Suma de cuadrados debido a los tratamientos

$$SCT = \sum_{j=1}^k n_j (\bar{x}_j - (\sum_{j=1}^k \bar{x}_j) / k)^2$$

SCE: Suma de cuadrados debidos al error

$$SCE = \sum_{j=1}^k (n_j - 1) s_j^2$$

$n$ : tamaño de muestra

$s^2$  = varianza de la muestra

STC: Suma total de cuadrados

$$STC = \sum_{j=1}^k \sum_{i=1}^{n_j} (x_{ij} - (\sum_{j=1}^k \bar{x}_j) / k)^2$$

Puesto que se asume que la muestra proviene de una población normal, es importante el análisis de los tres supuestos de residuales siguientes (el residual de una observación es la diferencia entre la observación y la media de la muestra que la contiene) [125]:

- 1.1) **Independencia:** En este estudio los desarrolladores generaron sus propios programas, de tal forma que los datos son independientes: cada programa tiene su propio tamaño y esfuerzo.
- 1.2) **Varianza constante:** En una gráfica de este tipo los residuales deberían mostrarse aproximadamente en una banda horizontal centrada y simétrica.
- 1.3) **Normalidad:** Una gráfica normal de residuales debería ser aproximadamente lineal.

2) **Validación del modelo:** Se concentra en la determinación de si el modelo funcionará bien en su ambiente pretendido de operación. Existen al menos dos formas de validar un modelo una vez que la adecuación de los modelos (o verificación) es comprobada: (1) consiste en dividir el conjunto de datos; entonces una parte del conjunto sirve para generar el modelo y la otra para probarlo, y la (2) que es usada en esta tesis: un nuevo conjunto de datos es estimado aplicando los modelos verificados. El criterio principal de comparación será la MMER. El criterio de la MMRE es calculado, pero no es relevante por recientes estudios publicados en [32]. Asimismo, este trabajo se basa en una reciente investigación [120], donde se demuestra que si la MRE o MER resultaran ser dependientes de la duración de los programas, el conjunto de datos debe dividirse (y entonces la MMRE y MMER deberían ser calculadas por cada submuestra).

En caso de ser divididas en submuestras, estas deben ser equivalentes [120], para ello el estadístico  $t$  de prueba es usado para la comparación de medias de submuestras, el cual se calcula de la siguiente forma:

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{s^2 (1/n_1 + 1/n_2)}}$$

Donde:

$$S^2 = \frac{(n_1 - 1) S_1^2 + (n_2 - 1) S_2^2}{n_1 + n_2 - 2}$$

$\bar{X}_1, \bar{X}_2$  = Medias de la muestras 1 y 2

$\mu_1, \mu_2$  = Medias poblacionales 1 y 2 supuestas (cero, en caso de desconocerlas)

$S_1^2, S_2^2$  = Varianzas de la muestras 1 y 2

$n_1, n_2$  = Número de elementos de las muestras 1 y 2

## 2.6 Lógica difusa

*El pensamiento humano, superposición de la intuición y de rigor, es decir, de una consideración global o en paralelo (necesariamente difusa) y de una consideración lógica o secuencial (necesariamente formal), es un pensamiento difuso.*

*Kaufman. 2000 [54]*

Recientes técnicas computacionales no algorítmicas para estimar esfuerzo fueron vistas en los noventas. Los investigadores pusieron su atención en un conjunto de enfoques basados en computación *soft* (o *soft-computing*: expresión utilizada para distinguirlo del enfoque algorítmico tradicional). Dentro de estos enfoques, la lógica difusa posee características tanto de representación lingüística que puede representar imprecisión en entradas y salidas, como proveyendo un enfoque basado en el conocimiento experto; con base en estas características entonces se construye el modelo de estimación [2].

Una desventaja en los modelos estadísticos es la forma por la que ellos disminuyen la comprensión de cómo son añadidas las variables, interacciones y transformaciones. Este problema puede ser al menos parcialmente superado con el uso de la lógica difusa, la cual fue desarrollada por la insatisfacción de la lógica clásica de *todo* o *nada*. La afirmación central subyacente de este enfoque difuso es que las entidades en el mundo real simplemente no pertenecen a una sola categoría. Por ejemplo, un proyecto no es solo *pequeño*, *mediano* o *grande*, sino podría de hecho ser un *gran* proyecto en su mayoría, pero también *mediano* en su minoría. Esto puede ser representado con un grado de pertenencia en una categoría lingüística particular [78].

Todas las técnicas de estimación tienen una limitación importante, la cual se muestra cuando proyectos de software son descritos usando datos categóricos (en escala nominal u ordinal) tales como *pequeño*, *mediano*, *promedio* o *alto*. Un enfoque más comprensivo para tratar con valores lingüísticos es precisamente usando la teoría de la lógica difusa [46].

Desde su origen por Zadeh en 1965, la lógica difusa ha sido objeto de importantes investigaciones. A principio de los noventas, la lógica difusa se mantuvo firme en términos de sus fundamentos teóricos y aplicaciones en diversos campos en los cuales estuvo usándose, tales como la robótica, medicina y procesamiento de imágenes. Un conjunto difuso es un conjunto con una función de membresía  $m$  en el intervalo real  $[0, 1]$ . Esta definición extiende aquella del conjunto clásico donde la función de membresía está en el par  $\{0, 1\}$ . Los

conjuntos difusos pueden ser efectivamente usados para representar valores lingüísticos tales como *bajo*, *mediano* o *complejo*. La representación por un conjunto difuso tiene las siguientes ventajas: (a) es más general, (2) imita la forma en la que la mente de los humanos interpretan los valores lingüísticos y (3) la transición de un valor lingüístico a un valor lingüístico continuo es más gradual que abrupto [47].

Hay un número de formas para la defuzificación de datos que podrían ser aplicadas al problema de la estimación del esfuerzo. Una de ellas y usada en esta tesis, es la de construir un sistema de inducción de reglas reemplazando los hechos abruptos (*crispy*) con entradas difusas; luego una máquina de inferencia usa las reglas para corresponder las entradas a la salida difusa, la cual puede ser traducida a partir de un valor abrupto (*crispy*) [111].

### 2.6.1 Conjuntos difusos

*“The term fuzzy logic is currently used in two different senses. In a narrow sense, fuzzy logic is a logical system (linguistic variable, if-then rules) that aims at a formalization of approximate reasoning. In a broad sense, fuzzy logic is almost synonymous with fuzzy set theory”*

*Lotfi A. Zadeh [127]*

En lógica difusa todo, incluyendo la verdad, es un asunto de grado [104].

La lógica difusa se deriva directamente de la teoría de los conjuntos difusos y los valores de verdad de la lógica difusa son representados por conjuntos difusos [104].

En esta sección se revisan las principales definiciones y conceptos de la teoría de los conjuntos ordinarios (aquellos que son el fundamento de las matemáticas modernas), pero estos conceptos se reexaminan y se extienden a las nociones que pertenecen a los conjuntos difusos. La notación y definiciones fueron extraídas de un libro que combina estos aspectos [54].

Sea  $E$  un conjunto y  $A$  un subconjunto de  $E$ :

$$A \subset E$$

Se acostumbra indicar con el símbolo  $\in$  que un elemento  $x$  del conjunto  $E$  pertenece al conjunto  $A$ :

$$x \in A$$

Para indicar esta membresía, también se puede utilizar otro concepto: el de función característica  $\mu_A(x)$ , cuyo valor indica si  $x$  pertenece o no a  $A$ .

$$\mu_A(x) = 1 \text{ si } x \in A$$

$$\mu_A(x) = 0 \text{ si } x \notin A$$

Entonces si se tiene un conjunto finito con 5 elementos:

$$E = \{x_1, x_2, x_3, x_4, x_5\}$$

y

$$A = \{x_2, x_3, x_5\}$$

Se escribe

$$\mu_A(x_1) = 0, \mu_A(x_2) = 1, \mu_A(x_3) = 1, \mu_A(x_4) = 0, \mu_A(x_5) = 1$$

Esto nos permite representar a  $A$  haciendo acompañar los elementos de  $E$  con valor de la función característica:

$$A = \{(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 0), (x_5, 1)\}$$

Para establecer el concepto de conjunto difuso, considérese el subconjunto  $A$  de  $E$  ( $A \subset E$ ). Los cinco elementos de  $E$  pertenecen o no pertenecen a  $A$ , lo uno o lo otro. La función característica sólo puede tomar al valor de 0 ó 1. Si por lo contrario, esta función pudiera tomar un valor cualquiera en el segmento  $[0,1]$ , un elemento  $x_i$  de  $E$  podría no pertenecer a  $A$  ( $\mu_A = 0$ ), pertenecer un poco a  $A$  ( $\mu_A$  cercano a 0), pertenecer bastante a  $A$  ( $\mu_A$  no muy cercano a 0 ni muy cercano a 1), pertenecer fuertemente a  $A$  ( $\mu_A$  cercano a 1) o, por último, pertenecer a  $A$  ( $\mu_A = 1$ ).

Entonces el concepto matemático de la expresión siguiente:

$$A = \{(x_1 | 0.2), (x_2 | 0), (x_3 | 0.3), (x_4 | 1), (x_5 | 0.8)\}$$

Donde  $x_i$  es un elemento del conjunto referencia  $E$  y donde el número colocado después de la barra (|) es el valor de la función característica de este elemento, este concepto es el de conjunto difuso de  $E$  y se representa de la siguiente forma:

$$\underset{\sim}{A} \subset \underset{\sim}{E} \quad \text{o también} \quad \underset{\sim}{A} \subset \underset{\sim}{E}$$

La membresía a un conjunto difuso se puede representar como sigue:

$$\underset{0.2}{x} \in \underset{\sim}{A} \quad \underset{1}{y} \in \underset{\sim}{A} \quad \underset{0}{z} \in \underset{\sim}{A}$$

El símbolo  $\underset{1}{\in}$  puede ser equivalente a  $\underset{0}{\in}$  y  $\underset{0}{\in}$  a  $\notin$ .

Una definición rigurosa de este concepto según Zadeh es la siguiente:

Sea  $E$  un conjunto, enumerable o no, y  $x$  un elemento de  $E$ ; entonces, un conjunto difuso  $A$  de  $E$  es el conjunto de pares ordenados siguiente:

$$\{(x | \mu_A(x)), \forall x \in E\}$$

Donde  $\mu_A(x)$  es el grado de membresía de  $x$  en  $A$ . Así, si  $\mu_A(x)$  toma sus valores en un conjunto  $M$  llamado conjunto de membresía, se puede decir que  $x$  toma sus valores en  $M$  mediante la función siguiente:

$$\underset{\sim}{\mu_A(x)}$$

Existen las siguientes operaciones simples entre conjuntos difusos:

**a) Inclusión.** Sean  $E$  un conjunto y  $M$  su conjunto de membresía asociado,  $A$  y  $B$  dos conjuntos difusos de  $E$ ; se dirá que  $A$  está incluido en  $B$  si:

$$x \in E: \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$$

Representado por

$$\tilde{A} \subset \tilde{B} \quad \text{o} \quad \tilde{A} \subset \tilde{B}$$

**b) Igualdad.** Sea  $E$  un conjunto y  $M$  su conjunto de membresía asociado,  $A$  y  $B$  dos conjuntos difusos de  $E$ ; se dirá que  $A$  y  $B$  son iguales si, y sólo si:

$$\forall x \in E: \mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x)$$

**c) Complemento.** Sean  $E$  un conjunto y  $M = [0,1]$  su conjunto de membresía asociado,  $A$  y  $B$  dos conjuntos difusos de  $E$ ; se dice que  $A$  y  $B$  son complementarios si:

$$\forall x \in E: \mu_{\tilde{B}}(x) = 1 - \mu_{\tilde{A}}(x)$$

**d) Intersección.** Sean  $E$  un conjunto y  $M = [0,1]$  su conjunto de membresía asociado,  $A$  y  $B$  dos conjuntos difusos de  $E$ ; se define la intersección

$$\tilde{A} \cap \tilde{B}$$

como el conjunto difuso más grande contenido a la vez en  $A$  y en  $B$ . Es decir:

$$\forall x \in E: \mu_{\tilde{A} \cap \tilde{B}}(x) = \text{MIN}(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

Lo que permite introducir un *and* difuso (“si  $A$  es el conjunto difuso de los números reales muy cercanos a 5 y  $B$  el conjunto difuso de los números reales muy cercanos a 10, entonces  $A \cap B$  es el conjunto difuso de los números reales muy cercanos a 5 y 10”, por ejemplo).

**e) Unión.** Sean  $E$  un conjunto y  $M = [0,1]$  su conjunto de membresía asociado,  $A$  y  $B$  dos conjuntos difusos de  $E$ ; se define la unión (o reunión) como sigue:

$$\tilde{A} \cup \tilde{B}$$

por el conjunto difuso más pequeño que contiene tanto a  $A$  como a  $B$ , esto es:

$$\forall x \in E: \mu_{\tilde{A} \cup \tilde{B}}(x) = \text{MAX}(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

Lo que permite introducir un *and/or* difuso (“si  $A$  es el conjunto difuso de los números reales muy cercanos a 5 y  $B$  el conjunto difuso de los números reales muy cercanos a 10, entonces  $A \cup B$  es el conjunto difuso de los números reales muy cercanos a 5 y/o a 10”, por ejemplo).

**f) Suma disyuntiva (o diferencia simétrica).** La suma disyuntiva de dos conjuntos difusos se define en términos de la unión e intersección:

$$A \oplus B = (\underbrace{A \cap B}_{\sim}) \cup (\underbrace{A \cap B}_{\sim})$$

En este caso corresponde al “o disyuntivo difuso”

**g) Diferencia.** Es definida por la siguiente relación

$$A - B = \underbrace{A}_{\sim} \cap \underbrace{B}_{\sim}$$

La línea (  $\sim$  ) sobre el conjunto B indica su complemento.

## 2.7 Estado del Arte

Fueron revisados diversos trabajos involucrando aspectos relacionados con la investigación de modelos basados en lógica difusa para la estimación del esfuerzo de desarrollo de software a nivel personal. Se describen enseguida aquellos encontrados:

Ahmed M. A. et al. [2] presentan un modelo difuso basado en funciones de membresía del tipo triangular. Su conjunto de datos para la validación fueron generados aleatoriamente, así como aquellos del COCOMO 81. Los resultados favorecieron ligeramente al modelo difuso sobre las ecuaciones del COCOMO 81.

Gray A. R. [33] compara diversos modelos de estimación. Menciona únicamente a los modelos difusos, pero no compara sus resultados con otros modelos.

Gray A. et al. [34] comparan los resultados de un modelo difuso para la estimación con otras técnicas como regresión lineal y redes neuronales. El modelo difuso hace uso de funciones de membresía triangulares y solo fue mejorado por una red neuronal perceptron multicapa. Los datos de prueba son extraídos de una tesis canadiense.

Host M. y Wohlin C. [38] reportan un experimento replicado a nivel personal (PSP), pero con base en la técnica de juicio experto (no hace uso de lógica difusa).

Huang X. et al. [39], proponen un modelo que combina la lógica difusa con las redes neuronales. Su fuente de datos es el conjunto de proyectos del COCOMO original (de 1981). Sus resultados favorecen al uso de la lógica difusa sobre el bien conocido COCOMO 81. Emplea funciones de membresía del tipo triangular.

Idri A. et al. [44] aplica la lógica difusa a los quince factores del COCOMO 81 intermedio. Hacen uso de funciones de membresía trapezoidales. Para validar el modelo generan (aleatoriamente) tres conjuntos artificiales de valores y se comparan con los valores reales del COCOMO 81. Sus resultados son muy similares con los originales (del COCOMO 81) y en algunos casos fue peor para los modelos difusos. En sus conclusiones sugieren el uso de otras funciones de membresía incluyendo la triangular.

Idri A. et al. [46] propone un enfoque basado en lógica difusa denominado *Fuzzy Analogy*, mas la técnica de estimación es por analogía. Su fuente de datos es la del COCOMO 81.

Musflek P. et al. [97] plantean un modelo difuso para el COCOMO 81 denominado f-COCOMO que hace énfasis en la relación de conjuntos difusos del tamaño con conjuntos difusos el esfuerzo. Hacen uso de funciones de membresía triangulares.

Pedrycz W. [103] justifica el uso de la lógica difusa para la estimación del esfuerzo de desarrollo. Como ejemplo incluye el COCOMO 81 y funciones de membresía triangulares.

Reformat M. et al. [108] proponen un modelo de estimación para el esfuerzo de desarrollo basado en redes neuronales difusas. Lo aplican a un sistema de información médico. El conjunto de datos es dividido en tres partes, una de ellas sirve como datos de prueba.

Ryder J. [110] propone el uso de la lógica difusa (con funciones de membresía triangular) en los factores de ajuste del modelo de estimación COCOMO 81 y en aquellos basados en Puntos de Función. No hace comparación de sus resultados.

Xu Z. et al. [126] proponen una técnica que genera funciones de membresía a partir de valores lingüísticos. Sus datos de prueba provienen del COCOMO 81. Sus resultados favorecen a la lógica difusa sobre el modelo COCOMO 81.

En esta muestra de trabajos encontrados puede identificarse que raramente las técnicas son investigadas con conjuntos de datos más allá de aquellos que fueron hace tiempo ya propuestos (como los datos de 63 programas a partir de los cuales Boehm en 1981 generó las ecuaciones del COCOMO 81). La explicación podría ser que dada la popularidad del COCOMO 81, entonces un hecho relevante sería comparar un modelo difuso con los resultados de las ecuaciones de Boehm.

En general, las evaluaciones de la precisión de la predicción de esfuerzo en los mencionados trabajos no solo revelan las fortalezas y debilidades de los métodos de estimación, sino también las ventajas y limitaciones de los conjuntos de datos usados. En esta investigación se reunieron datos propios de 364 programas desarrollados por 52 personas, de las cuales 37 fueron útiles para generar, verificar y validar los tres modelos difusos. Asimismo, son empleadas en esta investigación las funciones de membresía triangular, gaussiana y trapezoidal.

El autor de esta tesis es autor de artículos relacionados con el estudio de la ingeniería del software en nuestro país [63] [66] y con la aplicación propia del PSP [75]. Asimismo, es coautor de los siguientes cinco trabajos relacionados con la verificación y validación de modelos estadísticos y difusos aplicados a la estimación del esfuerzo de desarrollo de software a nivel personal y organizacional:

El primero [70] está basado en prácticas personales, pero no a nivel de programas, sino a nivel de 41 módulos de programas; la fuente de datos es de diez programas desarrollados por el autor; se hizo uso de funciones de membresía triangulares y los resultados (con alcance de verificación de modelos) favorecieron al modelo difuso sobre la regresión lineal múltiple.

El segundo [74] compara modelos de regresión simple con modelos difusos. Los modelos se generaron considerando noventa y nueve programas agrupados por prácticas de dieciocho desarrolladores. Se emplearon funciones de membresía triangulares y los resultados fueron mejores en los modelos difusos.

Un tercer artículo [73] compara un modelo difuso basado en líneas de código nuevas y cambiadas y en funciones de membresía triangular con uno de regresión. Ambos modelos se generaron de ciento cinco programas de treinta desarrolladores y ambos se validaron con 20 programas de otras siete personas. Los resultados favorecieron al modelo difuso.

Un cuarto trabajo [72] compara un modelo difuso conteniendo funciones de membresía triangular con uno de regresión. Ambos modelos no solo se basan en líneas de código nuevas y cambiadas, sino en reutilizadas. La fuente de datos fue de noventa y cuatro programas de dieciséis programadores. Los resultados favorecieron al modelo difuso.

El quinto trabajo [71] es un estudio replicado de diez programadores que desarrollaron siete programas cada uno. Por cada uno de ellos se genera un modelo difuso y uno de

regresión. Se hace un análisis para la comprobación de la adecuación de los modelos (verificación).

Con base en los criterios de comparación en los anteriores trabajos, la precisión de la estimación de un modelo difuso fue mejor en la mayoría de los casos. De igual forma resalta el uso exitoso de funciones de membresía triangulares.

En el capítulo tres se describe el proceso de la generación de los modelos difusos y bajo qué criterios fue elegido el mejor de ellos, incluso comparándolo con la regresión estadística.

## Capítulo 3

### Modelo propuesto

#### 3.1 Proceso para la generación del modelo difuso

En la Tabla 2 se describe el proceso para verificar y validar modelos difusos que hagan uso de funciones de membresía triangulares, gaussianas y trapezoidales.

Dentro del ámbito de la estimación del esfuerzo, una de las recomendaciones más representativas es la calibración de modelos a entornos locales [1] [29] [123]; el proceso descrito en la Tabla 2, además de ser el que siguió en esta investigación, sirve para que cualquier empresa pueda ser guiada por el mencionado proceso y generar sus propios modelos para después compararlos e identificar aquél con mayor precisión en la estimación del esfuerzo de desarrollo (precisión medida por su MMER).

El proceso de la Tabla 2 tiene cuatro secciones; la primera de ellas se refiere a aquellos requisitos que deben cumplirse antes de las demás actividades de secciones posteriores; la segunda describe los pasos a seguir para generar y verificar los modelos difusos. Una tercera parte se refiere a la validación de los modelos aplicados a programas nuevos, es decir, a la medición del comportamiento de los modelos generados en la sección tres. La cuarta parte establece el criterio de salida de este proceso: un modelo difuso que cumpla con la siguiente condición: haber logrado una  $MMER \leq 0.25$ .

**Tabla 2.** Proceso para la generación, verificación y validación de un modelo de estimación difuso

<b>Condiciones de entrada</b>	<ul style="list-style-type: none"><li>a) Seleccionar programas a desarrollar.</li><li>b) Establecer paradigma del lenguaje de programación con el que se codificarán los programas (imperativo: procedimental, orientado a objetos o concurrente; declarativo: lógico, funcional o manejadores de bases de datos).</li><li>c) Obtener tamaños y esfuerzos por cada programa con base en un diseño cuasiexperimental considerando las fuentes de invalidación interna siguientes: historia, maduración, inestabilidad, administración de las pruebas, instrumentación, regresión estadística (incluyendo identificación de valores atípicos), selección de participantes, mortalidad experimental e interacción entre selección y maduración.</li><li>d) Integrar una muestra de programas con tamaño y complejidad similar para la generación y verificación del modelo.</li><li>e) Comprobar hipótesis de correlación (<math>r &gt; 0.71</math>) y de coeficiente de determinación (<math>r^2 \geq 0.5</math>) entre variable(s) independiente(s) y esfuerzo en programas seleccionados dentro de la muestra.</li><li>f) Responder el cuestionario de Stutzke (<i>descrito en la sección 3.2</i>)</li></ul>
-------------------------------	--

Tabla 2. (continuación)

<b>Generación y verificación de modelos difusos</b>	<p>a) Definir reglas difusas con base en el resultado de la correlación de los datos: variable(s) independiente(s) – esfuerzo.</p> <p>b) Generar una primera versión del modelo difuso calculando los valores iniciales tanto del parámetro central superior como de los parámetros de la base de cada una de las funciones de membresía (sea esta del tipo triangular, gaussiana o trapezoidal). El cálculo de estos parámetros se hace tanto para la(s) entrada(s) como para la salida del modelo difuso.</p> <p>Para el modelo generado en (b) implementar (c), (d), (e) y (f):</p> <p>c) Aplicar el modelo a los datos originales de la muestra que sirvió como base para generar el modelo.</p> <p>d) Obtener un primer valor de la MMER</p> <p>e) Ajustar los parámetros superiores e inferiores de cada función de membresía del modelo</p> <p>f) Obtener la MMER</p> <p>g) Repetir (e) y (f) hasta obtener la menor MMER posible</p>
<b>Validación de modelos</b>	<p>a) Estimar el esfuerzo de desarrollo de nuevos programas aplicando el modelo difuso verificado</p> <p>b) Calcular la MER por programa.</p> <p>c) Comprobar hipótesis de precisión de estimación del modelo (<math>MMER \leq 0.25</math>) considerando análisis de heterocedasticidad (MER - Esfuerzo) previa.</p>
<b>Condición de salida</b>	Modelo difuso con una $MMER \leq 0.25$

### 3.2 Recolección de datos

Es importante diferenciar entre estimación del esfuerzo (ejemplo: en meses-persona) y estimación del tiempo calendario (ejemplo: en meses). El primero se refiere a la totalidad del tiempo que se debe invertir en un proyecto sea grande o pequeño, mientras que el segundo es cómo (en qué tipo: semanas o meses, y en cuántos lapsos de este tipo) ese esfuerzo es distribuido [121]. El esfuerzo recolectado en este trabajo se refiere a un esfuerzo individual (no en equipo) invertido en programas cuyo desarrollo no rebasó un día de duración, por lo que esfuerzo y tiempo calendario son equivalentes.

Stutzke [121] propone una lista de revisión (*checklist*) compuesta de 10 preguntas para la revisión de las medidas de productividad a usarse dentro del proceso de estimación del esfuerzo. Para este estudio, se mencionan y se responden a continuación:

1. ¿Qué tipos de software son involucrados con el tamaño?  
*Respuesta: Programas pequeños, más adelante en esta sección se describen los programas involucrados en este experimento.*
2. ¿Cuál es la definición de tamaño?  
*Respuesta: Líneas de código. En la sección 2.2 (Proceso personal de software) fueron definidos sus tipos. Asimismo, en la Tabla 3 se indica el estándar de cuantificación de código que siguieron todos los programas de este experimento.*

3. ¿Qué lenguaje de programación es usado?  
*Respuesta: Pascal, C, Basic y COBOL (lenguajes imperativos).*
4. ¿Qué ciclo de vida del proyecto fue usado?  
*Respuesta: Lineal secuencial, descrito por las fases propuestas por el PSP.*
5. ¿Qué fases del modelo de ciclo de vida del proyecto son cubiertas?  
*Respuesta: De acuerdo con la definición de desarrollo (sección 2.2) se consideraron las fases de diseño a pruebas, excluyendo planeación y postmortem.*
6. ¿Qué actividades del modelo de ciclo de vida del proyecto son cubiertas?  
*Respuesta: Registro de tamaño, tiempos y defectos dentro de las fases siguientes: diseño, revisión de diseño, codificación, revisión de código, compilación y pruebas (definidas en la sección 2.2)*
7. ¿Cuántas horas-persona son un mes persona?  
*Respuesta: Por tratarse de programas pequeños, la unidad de medida es en minutos.*
8. ¿El informe del esfuerzo incluye tiempo extra?  
*Respuesta: No, ni lapsos de interrupción durante el experimento, esto se controló haciendo uso del cuaderno de registro de tiempos (Anexo A).*
9. ¿Qué tan formal fue el proceso de desarrollo?  
*Respuesta: Consistente y siempre se respetó, no fueron considerados dentro de este estudio aquellos programas en los que sucedió lo contrario.*
10. Para todas las preguntas anteriores, ¿cada proyecto usó la misma definición, valor, modelo de ciclo de vida, formalidad, etc.?  
*Respuesta: Si.*

Los cuatro siguientes grupos conformaron la fuente de datos para la investigación:

- A) Integrantes del área de desarrollo de la Comisión Federal de Electricidad División Jalisco.
- B) Desarrolladores de empresas e integrantes de la generación Alfa del programa PAFTI organizado por el gobierno de Jalisco (Consejo Estatal de Ciencia y Tecnología de Jalisco, Coecytjal), industria (Grupo Aportia, *cluster* de empresas de Jalisco) y academia (Cinvestav, unidad Guadalajara).
- C) Desarrolladores de empresas e integrantes de la generación Beta programa PAFTI organizado por el gobierno (Coecytjal), industria (Grupo Aportia, *cluster* de empresas de Jalisco) y academia (Cinvestav, Guadalajara).
- D) Estudiantes de la Universidad del Valle de Atemajac (UNIVA), Guadalajara.

El número de participantes por grupo es el siguiente (sus nombres se muestran en el anexo B):

Grupo	Número de integrantes
A	13
B	10
C	15
D	14
Total	52

Cada uno de los cincuenta y dos programadores desarrolló siete programas, en total se obtuvieron datos de treientos sesenta y cuatro programas. Por cada uno se obtuvieron (a) el

número de líneas de código tanto NyC (nuevas y cambiadas) como reusadas y (b) los tiempos de cada una de las fases. Tal como se explicó en la sección 2.2, el tiempo total de desarrollo considerado por programa fue la suma de tiempos desde la fase de diseño hasta la de pruebas.

Siete programas fue un número debido a la disponibilidad de los participantes. Diez sesiones fueron llevadas a cabo, la primera sesión fue destinada a la elaboración de sus estándares de codificación y cuantificación. Siete sesiones fueron dedicadas a la elaboración de programas y las dos restantes fueron destinadas a la interpretación de sus datos con base en tópicos estadísticos y a la elaboración de un informe final.

Cada desarrollador había tomado al menos un curso sobre el lenguaje de programación imperativo que él/ella misma eligió. Todos ellos siguieron el mismo proceso de desarrollo y fueron constantemente supervisados y asesorados en cualquier duda que surgiera.

Treinta y siete de los cincuenta y dos desarrolladores integraron la muestra final y sus programas fueron la fuente de datos para este estudio. el modelo de regresión y los tres difusos fueron generados a partir de los programas de treinta de ellos. Los programas de los otros siete sirvieron para la validación de los modelos.

Cada programador estableció al inicio del curso dos estándares, uno de ellos de codificación y otro de cuantificación de código. Todos los desarrolladores coincidieron en lo siguiente con relación al estándar de codificación (susceptible de ser considerado como línea de código por el estandar de cuantificación): fueron escritas en una sola línea cada directiva del compilador, declaración de variable, definición de constante, delimitador, sentencia de asignación, instrucciones de estructuras de control de flujo selectiva y repetitiva.

Con base en [40], en la Tabla 3 se muestra el estándar de cuantificación de código al que todos los programas se sujetaron.

**Tabla 3.** Estándar de cuantificación de código

<b>Tipo de conteo</b>	<b>Tipo</b>
Físico/Lógico	Físico
<b>Tipo de instrucción</b>	<b>¿Es incluido?</b>
Ejecutable	Sí
No Ejecutables	
Declaraciones	Sí, una por línea de texto
Directivas del Compilador	Sí, una por línea de texto
Comentarios	No
Líneas en blanco	No
<b>Delimitadores</b>	
{ y }, begin y end	Si

Los lapsos de la impartición del curso de PSP por grupo fue la siguiente:

<b>Grupo</b>	<b>Lapso</b>
A	Julio – Agosto 2005
B	Enero – Marzo 2006
C	Marzo – Mayo 2006
D	Enero – Diciembre 2005

La muestra de los treinta y siete desarrolladores es la siguiente (se les asocia un identificador):

A) Comisión Federal de Electricidad

- CFE1 Barraza Arellano Irma Judith
- CFE2 De la Cruz Preciado Omar
- CFE3 Flores Gómez Carmen
- CFE4 Galindo Gauna Rogelio
- CFE5 García Ramos Manuel
- CFE6 Guerra Martínez Azor
- CFE7 Guzmán Martínez Alberto
- CFE8 Hernández Hernández Paola
- CFE9 Hernández Ramos Adriana
- CFE10 Partida Menchuca Laura

B) Grupo PAFTI-Alfa

- A1 Alegría Bobadilla Juan Manuel
- A2 Escamilla Rodríguez Jorge Alej.
- A3 BaltasarRamírez Francisco José
- A4 Montesinos S. José Carlos
- A5 Morales López Diego Jesús
- A6 Plascencia Sánchez Juan Manuel
- A7 Reynoso Rojas Ricardo
- A8 Rivera Vega Bolivar Ulises
- A9 Vega Baray Fermín Martín
- A10 Viramontes Cortés Armando

C) Grupo PAFTI-Beta

- B1 Cordero Baltasar Diego
- B2 Davis Alcaraz Robert
- B3 Díaz Infante Montes Jorge
- B4 Domínguez Zárata Sahara
- B5 Duarte Lobo Miguel Angel
- B6 Jiménez Galicia Nayeli
- B7 Montero Silva Alberto
- B8 Martínez Sotelo Nicandro
- B9 Rocha Hernández José
- B10 Vega Ávalos César

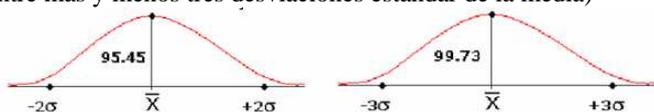
D) UNIVA

- U1 Becerril Ramírez Jorge Luis
- U2 Caro Guerra Rafael
- U3 Gutiérrez Hernández Ana Christina
- U4 Herrera Rábago Francisco Javier
- U5 Medina Estrada Christian
- U6 Sánchez Sánchez Francisco Javier
- U7 Tamayo Emmanuel

Fue desarrollado por cada grupo un conjunto diferente de programas que sirvió para generar los modelos de regresión y difusos. En la Tabla 4 se muestran los programas desarrollados en los grupos.

**Tabla 4.** Programas desarrollados dentro de la investigación

Grupo	Descripción del programa
A	<p>1. Cálculo de la media de una muestra de <math>n</math> números reales:</p> $\bar{X} = \frac{\sum_{i=1}^n (x_i)}{n}$ <p>2. Cálculo de la desviación estándar de una muestra de <math>n</math> números reales</p> $S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{(n-1)}}$ <p>3. Cálculo de la suma de dos matrices compuestas por números reales</p> <p>4. Cálculo de la suma de la diagonal de una matriz compuesta por números reales</p> <p>5. Transformación de una cantidad en número a letras</p> <p>6. Cálculo de la correlación entre dos series de números</p> $r = \frac{n \left[ \sum_{i=1}^n (LDC_i \cdot E_i) \right] - \left[ \left( \sum_{i=1}^n LDC_i \right) \left( \sum_{i=1}^n E_i \right) \right]}{\sqrt{\left[ n \left( \sum_{i=1}^n LDC_i^2 \right) - \left( \sum_{i=1}^n LDC_i \right)^2 \right] \left[ n \left( \sum_{i=1}^n E_i^2 \right) - \left( \sum_{i=1}^n E_i \right)^2 \right]}}$ <p>7. Cálculo de los parámetros <math>a</math> y <math>b</math> de la ecuación de regresión</p> $b = \frac{n \left[ \sum_{i=1}^n (LDC_i \cdot E_i) \right] - \left( \sum_{i=1}^n LDC_i \right) \left( \sum_{i=1}^n E_i \right)}{n \left( \sum_{i=1}^n LDC_i^2 \right) - \left( \sum_{i=1}^n LDC_i \right)^2}$ $a = \frac{\sum_{i=1}^n E_i}{n} - b \frac{\sum_{i=1}^n LDC_i}{n}$

B	<p>1. Cálculo de la media de una muestra de <math>n</math> números reales</p> $\bar{X} = \frac{\sum_{i=1}^n (x_i)}{n}$ <p>2. Cálculo de la media de números negativos y de positivos a partir de una lista de <math>n</math> números reales positivos y negativos</p> <p>3. Cálculo de los valores <math>z</math> de una muestra de <math>n</math> números reales</p> $z_i = \frac{x_i - \bar{X}}{s}$ <p>4. Cálculo del tamaño de una muestra</p> $n = \left( \frac{z * s}{E} \right)^2$ <p>5. Cálculo de los valores <math>y</math> con la ecuación de la distribución normal a partir de una muestra aleatoria de <math>n</math> números reales</p> $y = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$ <p>6. Cálculo del error estándar de estimación</p> $S_{y,x} = \sqrt{\frac{\sum (y - y')^2}{n-2}}$ <p>7. Cálculo del coeficiente de determinación (<math>r^2</math>) de una muestra de pares de datos:</p> $r^2 = \frac{SCR}{SCT}$ <p>Donde: <math>SCR = \sum_{i=1}^n (y'_i - \bar{y})^2</math>    <math>SCT = \sum_{i=1}^n (y_i - \bar{y})^2</math></p>
C	<p>1. Cálculo de la desviación estándar de una muestra de <math>n</math> números reales</p> $s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{(n-1)}}$ <p>2. Cálculo de los límites superior e inferior de una muestra de <math>n</math> números reales con base en la varianza y media.</p> $LS = \bar{X} + 3 \frac{s}{\sqrt{n}} \quad LI = \bar{X} - 3 \frac{s}{\sqrt{n}}$ <p>3. Comparación de la dispersión relativa en dos distribuciones mediante el coeficiente de variación.</p> $CV = \frac{s}{\bar{X}}$ <p>4. Estimación de valores esperados dentro de una muestra de números reales mediante la regla empírica (aproximadamente el 68% de las observaciones estarán entre más y menos una desviación estándar de la media; cerca del 95.45% de las observaciones entre más y menos dos desviaciones estándar de la media y 99.73% estarán entre más y menos tres desviaciones estándar de la media)</p>  <p>5. Cuantificación de líneas de código de un programa fuente</p> <p>6. Almacenamiento y recuperación de registros en un archivo</p> <p>7. Eliminación y modificación de un registro almacenado en un archivo</p>

El grupo D, que sirvió para validar los modelos, desarrolló los programas del grupo A. Los datos de los 30 desarrolladores y sus 210 programas se muestran en el Anexo C.

Los criterios bajo los que se basa la muestra extraída del Anexo C, son los siguientes:

- Incluirá solo aquellos programas del cuarto al séptimo (por tener prácticas iguales dentro del proceso de desarrollo, lo que del primero al tercero no sucede).
- Se excluirán programas muy pequeños (cuyo número de NyC se encuentre alejado, en desviaciones estándar, de la media del resto de los demás datos)
- Se excluirán observaciones atípicas o de influencia cuando se haya encontrado la justificación para ello.

La Tabla 5 muestra los programas cumpliendo el criterio A.

**Tabla 5.** Datos de desarrolladores para la generación de modelos aplicando el criterio A (D: Desarrollador, P: Programa, RD: Revisión de Diseño; RC: Revisión de Código)

	D	P	Código		Esfuerzo ( <i>en minutos</i> )					Esfuerzo Total	
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación		Prueba
1	CFE1	4	22	58	35	0	47	22	44	38	186
2		5	153	0	25	0	70	15	20	15	145
3		6	89	0	25	0	34	20	28	18	125
4		7	22	84	21	0	17	15	19	10	82
5	CFE2	4	17	77	15	1	14	5	10	11	56
6		5	77	0	47	1	44	6	14	12	124
7		6	96	0	35	1	37	7	5	49	134
8		7	17	89	11	2	18	3	1	28	63
9	CFE3	4	33	64	14	0	18	5	2	31	70
10		5	125	0	15	2	64	2	4	37	124
11		6	58	21	23	5	36	9	8	18	99
12		7	35	63	12	4	30	3	16	35	100
13	CFE4	4	5	25	5	1	14	2	2	23	47
14		5	75	23	15	5	89	2	5	8	124
15		6	32	17	17	1	26	2	3	43	92
16		7	20	40	14	5	28	3	0	9	59
17	CFE5	4	15	50	8	5	13	7	2	23	58
18		5	84	0	31	14	50	10	40	40	185
19		6	51	0	10	3	28	5	3	4	53
20		7	3	47	5	2	10	3	2	6	28
21	CFE6	4	13	32	7	3	9	3	3	8	33
22		5	129	0	22	4	36	4	17	5	88
23		6	30	25	18	1	19	4	1	49	92
24		7	8	48	7	5	8	2	1	26	49
25	CFE7	4	10	73	15	0	10	3	10	10	48
26		5	102	0	3	2	12	3	3	5	28
27		6	81	0	20	2	12	5	22	15	76
28		7	12	69	5	2	10	3	15	10	45
29	CFE8	4	20	75	5	1	27	3	1	35	72
30		5	117	0	4	3	39	8	4	13	71
31		6	136	0	3	3	68	3	10	33	120
32		7	21	115	5	1	21	4	1	40	72

Tabla 5. (continuación)

	D	P	Código		Esfuerzo (en minutos)					Esfuerzo Total	
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación		Prueba
33	CFE9	4	34	10	10	0	40	0	15	30	95
34		5	137	12	20	0	70	0	30	35	155
35		6	34	5	10	0	70	0	15	45	140
36		7	20	0	10	0	25	0	20	30	85
37	CFE10	4	22	68	15	5	25	5	10	25	85
38		5	111	20	30	5	50	5	25	20	135
39		6	87	12	45	5	40	5	10	20	125
40		7	28	68	25	5	20	5	15	25	95
41	A1	4	17	28	8	2	26	6	2	21	65
42		5	22	27	13	5	33	4	2	27	84
43		6	30	17	16	4	28	4	1	18	71
44		7	13	37	13	4	16	4	1	15	53
45	A2	4	47	42	15	6	54	4	4	44	127
46		5	30	33	18	3	100	1	0	55	177
47		6	47	59	10	2	54	4	2	18	90
48		7	17	96	2	1	25	0	1	3	32
49	A3	4	34	78	5	0	19	4	1	14	43
50		5	35	38	3	0	25	5	1	18	52
51		6	54	52	5	0	27	7	1	16	56
52		7	50	80	5	0	25	5	1	16	52
53	A4	4	11	33	5	3	19	5	3	9	44
54		5	13	28	18	3	22	2	10	65	120
55		6	25	8	10	3	10	2	1	24	50
56		7	14	27	7	3	15	2	1	4	32
57	A5	4	31	33	22	1	19	4	1	14	61
58		5	12	25	13	1	14	1	1	8	38
59		6	23	22	20	1	19	2	1	11	54
60		7	10	42	12	1	8	1	1	12	35
61	A6	4	19	56	13	3	13	11	4	15	59
62		5	26	40	13	13	22	9	4	41	102
63		6	23	11	14	4	15	3	2	33	71
64		7	11	33	12	2	15	3	3	39	74
65	A7	4	11	42	9	0	4	5	3	2	23
66		5	15	42	10	4	34	3	34	37	122
67		6	15	8	11	2	12	2	3	12	42
68		7	23	33	17	1	15	1	10	7	51
69	A8	4	26	34	14	0	40	0	1	36	91
70		5	17	41	10	0	44	0	4	35	93
71		6	19	23	8	0	23	0	2	33	66
72		7	13	34	8	0	32	0	1	21	62
73	A9	4	11	57	4	0	12	1	1	16	34
74		5	12	52	7	0	30	1	2	22	62
75		6	12	40	6	0	20	1	1	11	39
76		7	10	46	4	0	15	1	1	9	30
77	A10	4	60	51	12	2	17	2	1	24	58
78		5	45	18	57	4	16	3	35	30	145
79		6	49	0	14	2	27	3	3	31	80
80		7	81	65	7	2	30	2	4	14	59

Tabla 5. (continuación)

	D	P	Código		Esfuerzo (en minutos)					Esfuerzo Total	
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación		Prueba
81	B1	4	50	24	16	2	28	3	6	28	83
82		5	93	0	12	6	18	2	2	28	68
83		6	143	0	12	8	52	5	30	15	122
84		7	91	143	15	6	25	4	19	24	93
85	B2	4	36	24	20	0	50	10	10	10	100
86		5	17	21	5	0	20	5	5	30	65
87		6	117	0	15	0	55	10	40	30	150
88		7	42	111	20	0	35	5	10	30	100
89	B3	4	75	30	7	0	34	6	5	52	104
90		5	71	0	18	0	45	7	5	17	92
91		6	79	0	23	0	54	10	11	30	128
92		7	157	32	10	0	80	5	8	20	123
93	B4	4	48	41	42	15	55	14	10	55	191
94		5	45	17	38	17	58	23	6	44	186
95		6	81	32	12	4	68	18	7	46	155
96		7	119	100	14	13	67	14	6	54	168
97	B5	4	54	0	22	5	50	4	23	30	134
98		5	28	0	11	4	15	5	20	30	85
99		6	103	8	22	7	75	10	17	40	171
100		7	97	30	17	10	50	8	16	30	131
101	B6	4	37	21	32	1	39	8	8	35	123
102		5	16	12	12	1	12	2	1	36	64
103		6	49	18	51	3	30	6	54	15	159
104		7	123	46	109	2	40	7	7	30	195
105	B7	4	41	35	37	0	32	9	3	40	121
106		5	20	15	16	0	55	4	9	30	114
107		6	81	69	30	0	30	14	25	30	129
108		7	84	150	7	0	100	4	3	10	34
109	B8	4	29	0	10	0	22	3	4	10	49
110		5	29	63	8	38	0	10	3	39	98
111		6	182	50	8	0	48	30	35	20	141
112		7	63	150	3	0	50	20	20	20	113
113	B9	4	75	46	29	0	33	6	0	47	115
114		5	44	5	7	1	52	4	15	18	97
115		6	100	17	29	3	72	8	2	39	153
116		7	121	111	26	2	69	8	7	32	144
117	B10	4	79	43	4	0	62	10	1	56	133
118		5	51	0	10	0	25	10	1	24	70
119		6	112	8	3	0	57	11	6	15	92
120		7	84	119	6	0	51	11	1	9	78

El siguiente paso es cumplir el criterio B identificando aquellos programas muy pequeños. Es posible identificar los programas más pequeños empleando el puntaje  $z$  [124].

$$z_i = \frac{X_i - \overline{X}}{s}$$

Para lo que se requiere calcular la media y la desviación estándar de las NyC de los 120 programas. Se identificaron tres (Tabla 6) como más pequeños (menores que 10 NyC).

**Tabla 6.** Programas con su valor  $z$  más pequeño de la muestra, excluidos para generación de modelos

	D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
13	CFE4	4	5	25	5	1	14	2	2	23	47
20	CFE5	7	3	47	5	2	10	3	2	6	28
24	CFE6	7	8	48	7	5	8	2	1	26	49

Estos tres programas tuvieron el puntaje  $z$  más bajo obteniendo los valores que se calculan enseguida:

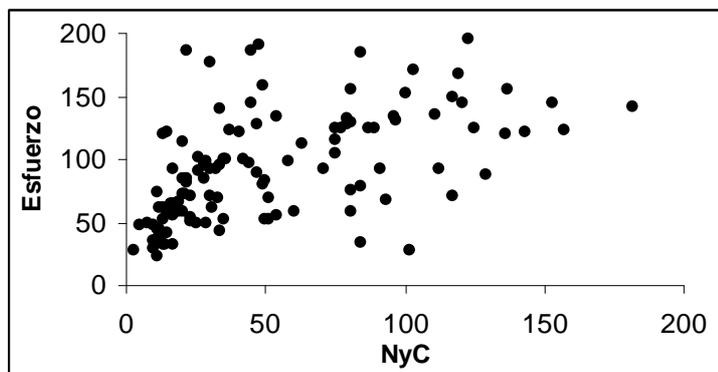
$$\text{Con } \begin{aligned} \bar{X} &= 51.50 \\ S &= 40.75 \end{aligned}$$

$$Z_{20} = \frac{x_i - \bar{X}}{S} = \frac{3 - 51.50}{40.75} = \frac{-48.5}{40.75} = -1.190$$

$$Z_{13} = \frac{x_i - \bar{X}}{S} = \frac{5 - 51.50}{40.75} = \frac{-46.5}{40.75} = -1.141$$

$$Z_{24} = \frac{x_i - \bar{X}}{S} = \frac{8 - 51.50}{40.75} = \frac{-43.5}{40.75} = -1.067$$

El criterio C trata sobre los valores atípicos. Con base en la gráfica de dispersión de la Figura 4 (generada a partir de la Tabla 5) se identificaron visualmente aquellos puntos alejados de la tendencia del resto de los demás, mismos que de manera personalizada fueron analizados con su correspondiente desarrollador.

**Figura 4.** Diagrama de dispersión de la Tabla 5

Durante el experimento se pudieron corregir los errores relacionados con el tamaño en NyC dado que se contaba con el listado del código fuente, mas en el caso de errores por el registro de tiempo por fase no existía evidencia para cambiar los datos reportados. Se identificaron doce casos (Tabla 7) en los que el desarrollador admitió algún error en su registro tiempos (pero ya no había forma de obtener el dato real del tiempo).

Tabla 7. Programas atípicos excluidos para generación de modelos

	D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
1	CFE1	4	22	58	35	0	47	22	44	38	186
26	CFE7	5	102	0	3	2	12	3	3	5	28
30	CFE8	5	117	0	4	3	39	8	4	13	71
35	CFE9	6	34	5	10	0	70	0	15	45	140
36	CFE9	7	20	0	10	0	25	0	20	30	85
46	A2	5	30	33	18	3	100	1	0	55	177
54	A4	5	13	28	18	3	22	2	10	65	120
66	A7	5	15	42	10	4	34	3	34	37	122
93	B4	4	48	41	42	15	55	14	10	55	191
94	B4	5	45	17	38	17	58	23	6	44	186
106	B7	5	20	15	16	0	55	4	9	30	114
108	B7	7	84	150	7	0	100	4	3	10	34

Con base en los criterios A, B y C establecidos para la muestra, de los 210 programas originales quedaron 105, los otros 105 se resumen en la Tabla 8.

Tabla 8. Resumen de programas excluidos para generación de modelos

Criterio		Número de programas
A	Programas 1,2 y 3	90
B	NyC < 10	3
C	Atípicos	12
Total		105

En la Figura 5 se observa el diagrama de dispersión de los 105 programas de la muestra final. El valor de su correlación y el del coeficiente de determinación de esta muestra son los siguientes:

$$r = 0.72, r^2 = 0.52$$

Con base en estos valores, la primera hipótesis nula formulada en este trabajo (sección 1.5.3, inciso a) se rechaza, aceptándose la siguiente: *A partir de una población, la muestra extraída posee una correlación (r) mayor o igual a 0.71 entre las líneas de código y el esfuerzo, es decir, con un  $r^2 \geq 0.50$ . Entonces la muestra de la población corresponde con un valor aceptable para un modelo de predicción lineal (aquel con el que serán comparados los modelos difusos).*

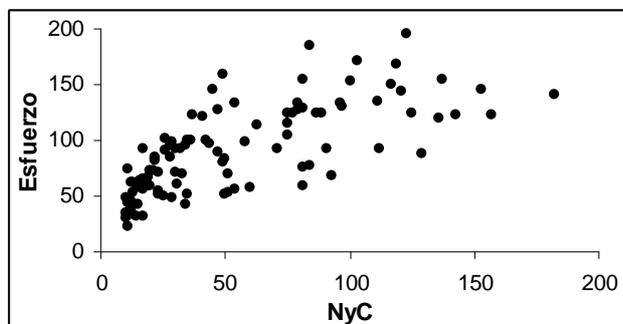


Figura 5. Diagrama dispersión de la muestra de programas para esta investigación

Por otra parte, en el Anexo D se muestran los datos de los 7 programadores que desarrollaron 49 programas de entre los cuales se validan los modelos.

Los mismos criterios A, B y C usados para la generación de los modelos, son aplicados para seleccionar la muestra de programas que servirá para validar los modelos (a extraer del Anexo D). La Tabla 9 muestra los programas cumpliendo el criterio A.

**Tabla 9.** Datos de desarrolladores para la validación de modelos aplicando criterio A (D: Desarrollador, P: Programa, RD: Revisión de Diseño; RC: Revisión de Código)

	D	P	Código		Esfuerzo ( <i>en minutos</i> )						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
1	U1	4	86	0	29	0	23	0	5	48	105
2	U1	5	75	0	37	0	67	0	1	25	130
3	U1	6	58	0	7	0	9	0	1	15	32
4	U1	7	56	0	48	0	67	0	5	40	160
5	U2	4	69	0	21	0	20	2	1	23	67
6	U2	5	107	0	47	0	56	1	10	27	141
7	U2	6	91	0	19	0	20	1	1	18	59
8	U2	7	53	0	30	0	55	1	1	35	122
9	U3	4	19	41	19	0	20	0	1	40	80
10	U3	5	92	7	26	0	50	0	1	100	177
11	U3	6	27	8	14	0	30	0	1	27	72
12	U3	7	7	31	4	0	9	0	1	20	34
13	U4	4	78	0	10	0	50	0	10	30	100
14	U4	5	155	0	20	0	120	0	10	30	180
15	U4	6	29	0	15	0	30	0	10	20	75
16	U4	7	40	0	15	0	45	0	10	20	90
17	U5	4	2	20	15	0	40	10	5	20	90
18	U5	5	57	0	10	0	30	10	5	20	75
19	U5	6	50	0	10	0	20	5	5	25	65
20	U5	7	55	0	10	0	15	5	5	20	55
21	U6	4	55	0	18	0	19	0	7	10	54
22	U6	5	53	0	16	0	22	0	16	8	62
23	U6	6	60	0	17	0	26	0	8	9	60
24	U6	7	6	60	6	0	13	0	6	5	30
25	U7	4	13	20	7	0	32	0	3	7	49
26	U7	5	109	0	25	0	69	0	3	9	106
27	U7	6	86	4	8	0	104	0	2	7	121
28	U7	7	6	88	8	0	38	0	1	9	56

Con base en el criterio B, se excluyen para su validación aquellos programas con menos de 10 NyC (puesto que los modelos no incorporaron programas con este tamaño). Cuatro programas tuvieron esta característica (Tabla 10).

**Tabla 10.** Programas menores que 10 NyC excluidos para validación de modelos

	D	P	Código		Esfuerzo ( <i>en minutos</i> )						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
12	U3	7	7	31	4	0	9	0	1	20	34
17	U5	4	2	20	15	0	40	10	5	20	90
24	U6	7	6	60	6	0	13	0	6	5	30
28	U7	7	6	88	8	0	38	0	1	9	56

Asimismo, con relación al criterio C, cuatro programas tuvieron errores en su registro de tiempos (Tabla 11).

**Tabla 11.** Programas atípicos excluidos para validación de modelos

	D	P	Código		Esfuerzo ( <i>en minutos</i> )						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
3	U1	6	58	0	7	0	9	0	1	15	32
7	U2	6	91	0	19	0	20	1	1	18	59
9	U3	4	19	41	19	0	20	0	1	40	80
26	U7	5	109	0	25	0	69	0	3	9	106

Con base en los criterios establecidos para la muestra, de los 49 programas originales quedaron 20, los otros 29 se resumen en la Tabla 12.

**Tabla 12.** Resumen de programas excluidos para validación de modelos

Criterio		Número de programas
A	Programas 1,2 y 3	21
B	NyC < 10	4
C	Atípicos	4
Total		29

### 3.3 Generación de modelos

#### 3.3.1 Modelos de regresión

##### 3.3.1.1 Regresión lineal múltiple

En la sección 2.2 (proceso personal de software) se mencionó que las líneas nuevas y cambiadas (NyC) toman mucho más tiempo (al adicionar o modificar una LDC) que eliminar o reutilizar código. Por otra parte, se ha determinado que el código reutilizado podría influir en la productividad de un desarrollador [121].

En este estudio se considera en la ecuación de regresión una variable independiente más, misma que es destinada a las líneas de código reutilizadas para conocer si es significativa

considerarla para la muestra de los 105 programas. En caso de ello, los modelos difusos asimismo deberían considerarse para efectos de comparación.

En la sección 2.5 (Estadística) se expuso el sistema de ecuaciones lineales que se deben resolver para generar una ecuación de regresión lineal múltiple. La variable dependiente corresponde al esfuerzo, mientras que las dos independientes corresponden a líneas de código nuevas y cambiadas (NyC), así como a las reutilizadas.

Es importante mencionar que para la generación de la ecuación se consideraron solo aquellos programas en los que se reutilizó código (85 de los 105 programas). El análisis de la ecuación generada se muestra en la Tabla 13 (validada con una herramienta estadística [119]).

**Tabla 13.** Análisis de parámetros del modelo de regresión lineal múltiple

Parámetro	Valor estimado	Estadístico <i>t</i>	Valor-p
Constante	56.7117	9.61586	0.0000
NyC	0.77068	10.4318	0.0000
Reutilizado	-0.12989	-1.4482	0.1514

Es decir, la ecuación es la siguiente:

$$\text{Esfuerzo} = 56.71 + (0.77 * \text{NyC}) - (0.129 * \text{Reutilizado})$$

El análisis de varianza (ANOVA) de este modelo se muestra en la Tabla 14.

**Tabla 14.** ANOVA del modelo de regresión lineal múltiple

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F	Valor-p
Modelo	73162.3	2	36581.2	54.84	0.000
Residual	54698.1	82	667.05		
Total	127860.0	84			

Dado que el valor-p en el ANOVA es menor que 0.01, entonces hay una relación estadística significativa con un 99% de nivel de confianza entre las variables NyC, reusado y el esfuerzo.

El coeficiente de determinación ( $r^2$ ) de esta ecuación es de 0.5722, es decir, que el modelo explica el 57.22 % de la variabilidad en el esfuerzo.

Con relación a si el modelo puede ser simplificado o no, debido a que el valor-p mayor a 0.10 en las variables independientes lo tiene el código reutilizado (con 0.1514), entonces esta variable no es estadísticamente significativa a un 90% de nivel de confianza. Por lo tanto, *reutilizado* puede ser extraído de la ecuación de regresión múltiple.

Puede comprobarse esto de otra forma comparando los coeficientes de determinación ( $r^2$ ) de la ecuación lineal simple con aquella lineal múltiple para el mismo conjunto de programas que reutilizan código.

La ecuación de regresión lineal simple con la variable independiente NyC de aquellos 85 programas, es la siguiente:

$$\text{Esfuerzo} = 50.992 + 0.765 * \text{NyC}$$

Su análisis de varianza se muestra en la Tabla 15, observándose que hay una relación estadística significativa con un 99% de nivel de confianza entre la variable NyC y el esfuerzo. Los datos presentan una correlación ( $r$ ) de 0.7491, es decir, su coeficiente de determinación ( $r^2$ ) es de 0.5612.

**Tabla 15.** ANOVA del modelo de regresión lineal simple considerando sólo programas con reutilización de código

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F	Valor-p
Modelo	71763.3	1	71763.3	106.18	0.0000
Residual	56097.1	83	675.869		
Total	127860.0	84			

Comparando los coeficientes de determinación de la ecuación múltiple  $r^2 = 57.22\%$  con la de la ecuación simple  $r^2 = 56.12\%$  se observa que la diferencia es mínima (1.1%). Es decir, que para la muestra de este estudio no es significativo considerar al código reutilizado.

### 3.3.1.2 Regresión lineal simple

Una vez que se ha descartado el uso del código reutilizado dentro de la ecuación de regresión, se procede a generar la ecuación de regresión simple considerando ahora sólo NyC en los 105 programas de la muestra. El análisis de la ecuación generada se muestra en la Tabla 16 (validada con una herramienta estadística [119]):

**Tabla 16.** Análisis de parámetros del modelo de regresión lineal simple

Parámetro	Valor estimado	Estadístico $t$	Valor-p
Intercepto	53.2915	12.1856	0.0000
Pendiente	0.687458	10.5906	0.0000

Es decir, tanto el intercepto como la pendiente son significativas (valor-p = 0.0) y la ecuación es la siguiente:

$$\text{Esfuerzo} = 53.291 + 0.687 * \text{NyC}$$

El análisis de varianza (ANOVA) de este modelo se muestra en la Tabla 17.

Tabla 17. ANOVA del modelo de regresión lineal simple

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F	Valor-p
Modelo	83389.1	1	83389.1	112.16	0.000
Residual	76577.7	103	743.472		
Total	159967.0	104			

Debido a que el valor-p en el ANOVA es menor que 0.01, entonces hay una relación estadística significativa con un 99% de nivel de confianza entre las variables NyC y el esfuerzo.

El coeficiente de determinación ( $r^2$ ) de esta ecuación es de 0.5212 (con correlación  $r = 72.2$ ), es decir, que el modelo explica el 52.12 % de la variabilidad en el esfuerzo. Obsérvese que estos valores de  $r$  y  $r^2$  coinciden con aquellos de la sección 3.2.

### 3.3.2 Modelos difusos

Un modelo difuso puede operar a nivel de términos lingüísticos (conjuntos difusos representados por funciones de membresía) [104].

El término difuso usualmente se refiere a las técnicas y algoritmos para construir modelos difusos a partir de datos.

Hay dos enfoques principales para obtener un modelo difuso a partir de datos [126]:

- 1) El conocimiento experto en modo verbal que es traducido a un conjunto de reglas si-entonces. Un cierto modelo puede ser creado y los parámetros de esta estructura, tales como funciones de membresía y pesos de las reglas, pueden establecerse por un experto usando datos de entrada y salida.
- 2) Sin conocimiento previo acerca de un sistema bajo estudio se formulan las reglas y luego se construye un modelo difuso a partir de datos basados en un cierto algoritmo. Se espera que las reglas y las funciones de membresía puedan explicar el comportamiento del sistema. Un experto puede modificar las reglas o suministrar nuevas basadas en la propia experiencia y su presencia es opcional.

Este estudio se basa en el primer enfoque. Las reglas difusas son formuladas por la correlación ( $r$ ) de pares de datos, a partir de la cual se interpreta que para el conjunto de datos reunido en esta investigación el esfuerzo es directamente proporcional al tamaño (esto es, que a menor tamaño menor esfuerzo y que a mayor tamaño mayor esfuerzo, con un  $r=0.72$ ). A partir de este valor de  $r$ , se derivan las siguientes tres reglas:

1. Si (*Nuevo y Cambiado es Pequeño*) entonces (*Esfuerzo es Bajo*)
2. Si (*Nuevo y Cambiado es Mediano*) entonces (*Esfuerzo es Promedio*)
3. Si (*Nuevo y Cambiado es Grande*) entonces (*Esfuerzo es Alto*)

Implementar un sistema difuso involucra que las categorías de las diferentes entradas y salidas sean representadas por conjuntos difusos a través de funciones de membresía (FM). En

este experimento se generan tres modelos difusos, cada uno con una función de membresía diferente: (1) triangular, debido a los casos de éxito que ha reportado su uso dentro de la estimación del esfuerzo de desarrollo de software, (2) gaussiana y (3) trapezoidal [2] [54] [104]. Para los tres modelos: (a) los valores de los parámetros son definidos de acuerdo con los valores cercanos o iguales a los valores mínimo y máximo de los tamaños y esfuerzos, (b) estos intervalos son divididos por los siguientes segmentos: *pequeño*, *mediano* y *grande* para el código NyC y en *bajo*, *promedio* y *alto* para el esfuerzo.

Los valores iniciales de los puntos centrales superiores (cuando son iguales a 1) de cada valor lingüístico (*pequeño*, *mediano* y *grande*, así como *bajo*, *promedio* y *alto*) se establecen como sigue:

1) Para cada entrada y salida (NyC y esfuerzo) se calcula el rango de valores identificando los valores mínimo y máximo de los 105 programas (en este estudio sus valores coincidieron):

	NyC	Esfuerzo
<b>Valor mínimo</b>	10	23
<b>Valor máximo</b>	182	195
<b>Rango</b>	172	172

2) El rango se divide por tres (por ser tres sus valores lingüísticos y por tanto se obtienen las tres funciones de membresía)

	NyC	Esfuerzo
<b>Rango</b>	172	172
<b>División por tres</b>	57.3	57.3

3) Se obtienen tres conjuntos de datos. Los límites del rango del primer conjunto son determinados por la suma del valor menor de los 105 programas más el valor resultante del punto (2). Para los límites del segundo conjunto se le suma el mismo valor del punto (2) al límite superior del primer conjunto. Por último, los límites del tercer conjunto son determinados por la suma del valor obtenido del punto (2) al límite superior del segundo conjunto.

NyC			Esfuerzo		
Pequeño	Mediano	Grande	Bajo	Promedio	Alto
$10+57=67$	$67+57=124$	$124+57=181$	$23+57=80$	$80+57=137$	$137+57=194$

4) Para cada uno de los tres conjuntos de entrada y salida obtenidos en el punto (3), se calcula el valor central superior inicial dividiendo por dos el rango obtenido de cada conjunto difuso (función de membresía).

Conjunto	NyC			Esfuerzo		
	Pequeño	Mediano	Grande	Bajo	Promedio	Alto
<b>Intervalo</b>	$10 \leq m \leq 67$	$68 \leq m \leq 124$	$125 \leq m \leq 182$	$23 \leq m \leq 80$	$81 \leq m \leq 137$	$132 \leq m \leq 195$
<b>m</b>	28	93	148	50	96	148

Para el caso de las funciones trapezoidales (que se explicará en la sección 3.2.2.3), el valor central superior ( $m$ ) se encuentra en el centro del intervalo  $b \leq x \leq c$ .

Asimismo, los dos parámetros iniciales correspondientes a la base de cada función de membresía se establecen de una manera tal, que equidistan del punto central superior de la función a la que pertenecen.

Los parámetros superiores e inferiores de cada función de membresía se ajustan con base en iteraciones hasta obtener el mejor resultado posible (MMER).

Durante las iteraciones, se evitó que el parámetro central superior de la función de membresía central (mediano, promedio) fuera tanto menor o igual que el parámetro central de la función de membresía izquierdo (pequeño, bajo), como mayor o igual que el parámetro central de la función de membresía derecho (grande, alto).

De igual forma, durante las iteraciones se aseguró que el parámetro izquierdo de la función de membresía central (mediano, promedio) se intersecara siempre con el parámetro derecho de la función de membresía izquierda (pequeño, bajo), así como que el parámetro derecho de la función de membresía central (mediano, promedio) se intersecara siempre con el parámetro izquierdo de la función de membresía derecha (grande, alto).

### 3.3.2.1 Modelo difuso con funciones de membresía triangular

La función de membresía triangular es una función de tres puntos, definidos como valores mínimo ( $a$ ), máximo ( $b$ ) y un punto entre ambos ( $m$ ), esto es,  $FM(a,m,b)$  donde  $a \leq m \leq b$ . En la Tabla 18 se muestran los parámetros finales de las tres funciones de membresía tanto de entrada como de salida. Sus parámetros escalares ( $a, m, b$ ) se definen como sigue:

$$FM(x) = 0 \text{ si } x < a$$

$$FM(x) = 1 \text{ si } x = m$$

$$FM(x) = 0 \text{ si } x > b$$

**Tabla 18.** Características de las funciones de membresía triangulares

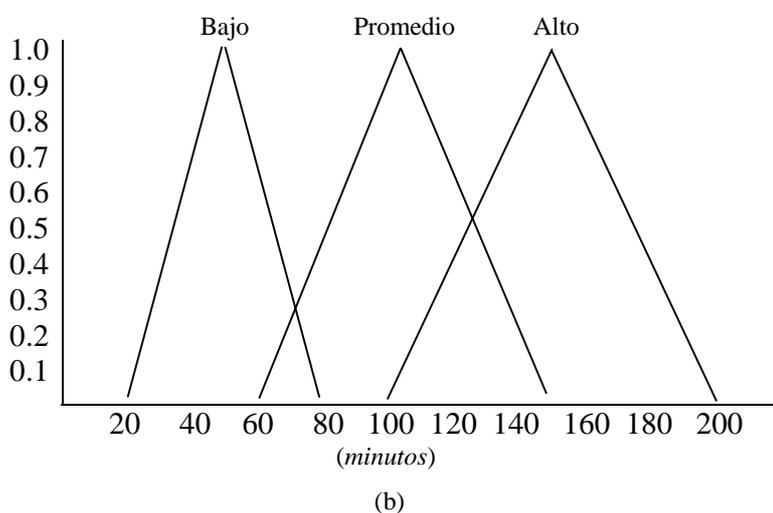
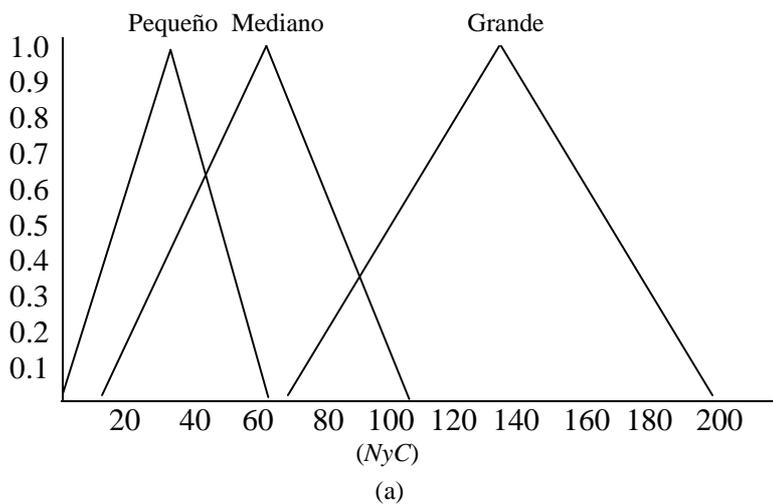
#### Entrada

Nombre de variable	Rango	Función de membresía	Parámetros		
			a	m	B
NyC	1 – 200	Pequeño	1	35	65
		Mediano	18	65	113
		Grande	72	136	200

#### Salida

Nombre de variable	Rango	Función de membresía	Parámetros		
			a	m	b
Esfuerzo	20 – 200	Bajo	20	51	80
		Promedio	60	104	153
		Alto	100	149	200

Las gráficas de las funciones de membresía de la Tabla 18 se muestran en las Figura 6.



**Figura 6.** Funciones de membresía triangulares del sistema difuso (a) NyC y (b) Esfuerzo

### 3.3.2.2 Modelo difuso con funciones de membresía gaussianas

Una función de membresía con curva gaussiana contiene dos parámetros. Uno de ellos ( $k$ ) determina la forma de la curva y el otro ( $m$ ) la posición central de la curva.

En la Tabla 19 se muestran los parámetros finales de las tres funciones de membresía tanto de entrada como de salida. Sus parámetros escalares ( $k$ ,  $m$ ) se definen como sigue:

$$FM(x) = e^{-k(x-m)^2}$$

Donde  $k > 0$ .

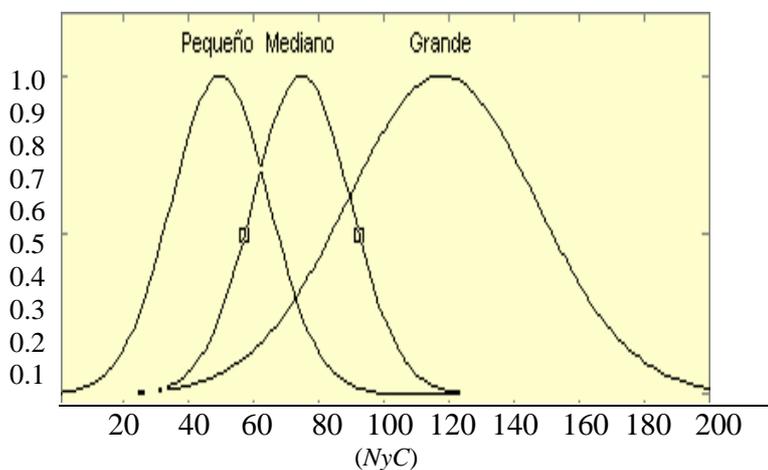
**Tabla 19.** Características de las funciones de membresía gaussianas**Entrada**

Nombre de variable	Rango	Función de Membresía	Parámetros	
			k	m
NyC	1 – 200	Pequeño	15	50
		Mediano	15	70
		Grande	29	118

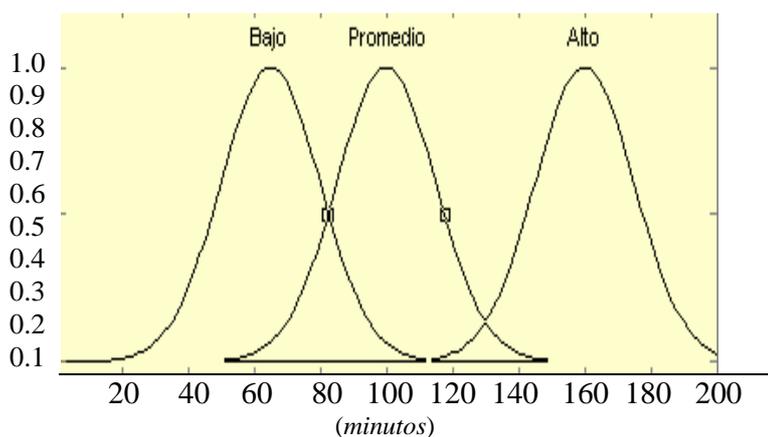
**Salida**

Nombre de variable	Rango	Función de Membresía	Parámetros	
			k	m
Esfuerzo	20 – 200	Bajo	15	65
		Promedio	15	100
		Alto	15	160

Las gráficas de las funciones de membresía de la Tabla 19 se muestran en la Figura 7.



(a)



(b)

**Figura 7.** Funciones de membresía gaussianas del sistema difuso (a) NyC y (b) Esfuerzo

### 3.3.2.3 Modelo difuso con funciones de membresía trapezoidales

Una función de membresía trapezoidal contiene cuatro parámetros  $a$ ,  $b$ ,  $c$  y  $d$ , cumpliéndose que  $a \leq b$  y  $c \leq d$ .

En la Tabla 20 se muestran los parámetros finales de las tres funciones de membresía tanto de entrada como de salida. Sus parámetros escalares ( $a$ ,  $b$ ,  $c$ ,  $d$ ) se definen como sigue:

$$FM(x) = 0 \text{ si } -\infty < x \leq a$$

$$FM(x) = 1 \text{ si } b \leq x \leq c$$

$$FM(x) = 0 \text{ si } d \leq x < \infty$$

**Tabla 20.** Características de las funciones de membresía trapezoidales

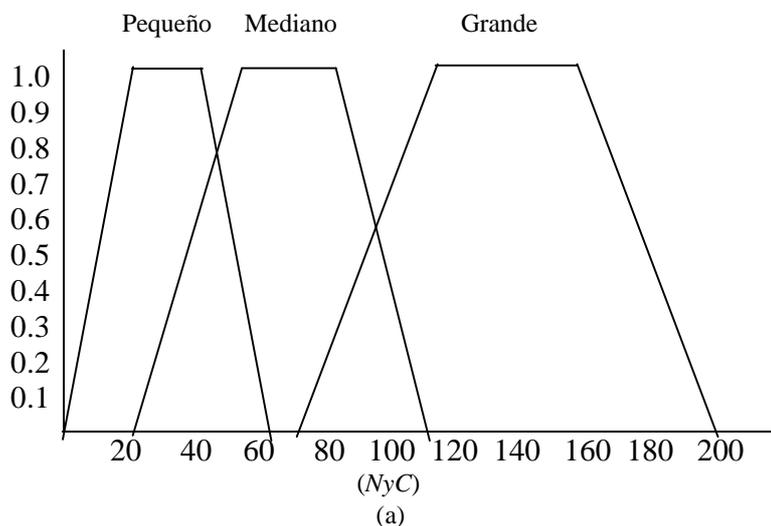
#### Entrada

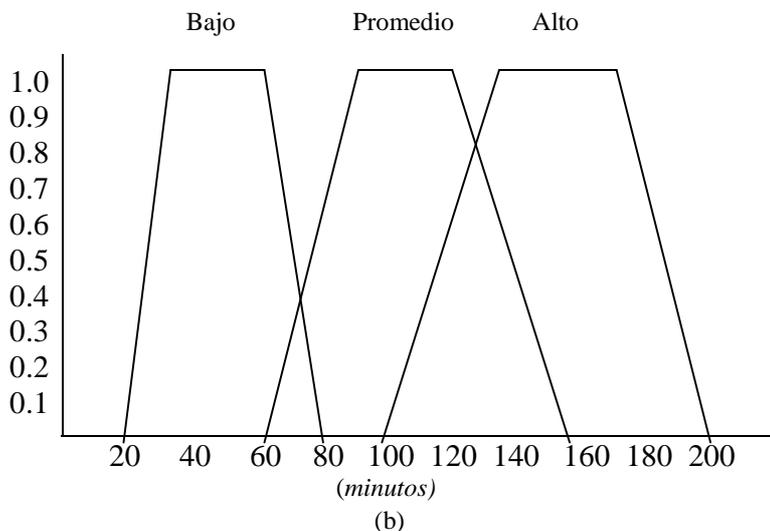
Nombre de variable	Rango	Función de Membresía	Parámetros			
			a	b	c	d
NyC	1 – 200	Pequeño	1	22	44	65
		Mediano	22	53	83	110
		Grande	72	115	157	200

#### Salida

Nombre de variable	Rango	Función de Membresía	Parámetros			
			a	b	c	d
Esfuerzo	20 – 200	Bajo	20	35	60	80
		Promedio	60	91	122	153
		Alto	100	133	166	200

Las gráficas de las funciones de membresía de la Tabla 20 se muestran en la Figura 8.





**Figura 8.** Funciones de membresía trapezoidales del sistema difuso (a) NyC y (b) Esfuerzo

### 3.4 Análisis de datos

#### 3.4.1 Comprobación de la adecuación de los modelos (verificación)

La ecuación de regresión simple y los modelos difusos se aplican a los datos originales (105 programas). El dato de entrada es el número de líneas de código (NyC) y el de salida el esfuerzo en minutos.

En la Tabla 21 se muestran los datos reales de las NyC y esfuerzo por programa, así como el valor que cada modelo predice; entonces son calculadas por modelo tanto la MRE como la MER de cada programa. Para generar los resultados de los modelos difusos se empleó una herramienta matemática computacional [80] con (a) tipo mamdani, (b) método *and: min*, (c) método *or: max*, (d) implicación: *min*, agregación: *max* y defuzzificación (*defuzzycation*): centroide.

En la sección 2.5.4 (pruebas para comparación de medias) se mencionó que para la validez de una comparación entre medias con base en pruebas  $F$  (ANOVA), deberían cumplirse los tres supuestos de residuales. El primero de ellos (independencia) se cumple dada la explicación en la sección 2.5.4, inciso 1.1.

En las Figuras 9a. y 9c. se muestran las gráficas de varianza constante, se observa que los residuales se encuentran aproximadamente en una banda horizontal centrada y simétrica.

En las Figuras 9b. y 9d. se muestran las gráficas de normalidad, en las que se observa que son aproximadamente lineales.

Las gráficas 9c y 9d relacionadas con la MER (principal criterio de comparación de este estudio) se apegan mejor a los criterios de los supuestos residuales.

De acuerdo a los criterios de la sección 2.5.4, se infiere que los tres supuestos de residuales se cumplen de manera satisfactoria (principalmente en el caso de la MER).

**Tabla 21.** MRE y MER de cada programa para la comprobación de la adecuación de modelos (D: Desarrollador, P: Programa, ER: Esfuerzo Real; ERL: Esfuerzo por Regresión Lineal; ELDT: Esfuerzo por Lógica Difusa Triangular; ELDG: Esfuerzo por Lógica Difusa Gaussiana; ELDP: Esfuerzo por Lógica Difusa Trapezoidal)

D	P	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
A1	4	17	65	64.98	0.00	0.00	50.2	0.23	0.29	67.80	0.04	0.04	49.20	0.24	0.32
A1	5	22	84	68.42	0.19	0.23	62.7	0.25	0.34	68.10	0.19	0.23	48.90	0.42	0.72
A1	6	30	71	73.92	0.04	0.04	73.5	0.04	0.03	69.90	0.02	0.02	68.40	0.04	0.04
A1	7	13	53	62.23	0.17	0.15	50.2	0.05	0.06	68.10	0.28	0.22	49.40	0.07	0.07
A2	4	47	127	85.60	0.33	0.48	84.8	0.33	0.50	81.10	0.36	0.57	83.30	0.34	0.52
A2	6	47	90	85.60	0.05	0.05	84.8	0.06	0.06	81.10	0.10	0.11	83.30	0.07	0.08
A2	7	17	32	64.98	1.03	0.51	50.2	0.57	0.36	67.80	1.12	0.53	49.20	0.54	0.35
A3	4	34	43	76.67	0.78	0.44	76.6	0.78	0.44	71.50	0.66	0.40	73.50	0.71	0.41
A3	5	35	52	77.35	0.49	0.33	77.3	0.49	0.33	72.00	0.38	0.28	74.50	0.43	0.30
A3	6	54	56	90.41	0.61	0.38	90.5	0.62	0.38	87.20	0.56	0.36	90.20	0.61	0.38
A3	7	50	52	87.66	0.69	0.41	87	0.67	0.40	83.90	0.61	0.38	86.20	0.66	0.40
A4	4	11	44	60.85	0.38	0.28	50.2	0.14	0.12	68.40	0.55	0.36	49.50	0.13	0.11
A4	6	25	50	70.48	0.41	0.29	68	0.36	0.26	68.60	0.37	0.27	58.30	0.17	0.14
A4	7	14	32	62.92	0.97	0.49	50.2	0.57	0.36	68.00	1.13	0.53	49.30	0.54	0.35
A5	4	31	61	74.60	0.22	0.18	74.4	0.22	0.18	70.20	0.15	0.13	69.90	0.15	0.13
A5	5	12	38	61.54	0.62	0.38	50.2	0.32	0.24	68.20	0.79	0.44	49.40	0.30	0.23
A5	6	23	54	69.10	0.28	0.22	64.7	0.20	0.17	68.20	0.26	0.21	52.40	0.03	0.03
A5	7	10	35	60.17	0.72	0.42	50.2	0.43	0.30	68.60	0.96	0.49	49.50	0.41	0.29
A6	4	19	59	66.35	0.12	0.11	54.3	0.08	0.09	67.80	0.15	0.13	49.10	0.17	0.20
A6	5	26	102	71.17	0.30	0.43	69.3	0.32	0.47	68.80	0.33	0.48	60.80	0.40	0.68
A6	6	23	71	69.10	0.03	0.03	64.7	0.09	0.10	68.20	0.04	0.04	52.40	0.26	0.35
A6	7	11	74	60.85	0.18	0.22	50.2	0.32	0.47	68.40	0.08	0.08	49.50	0.33	0.49
A7	4	11	23	60.85	1.65	0.62	50.2	1.18	0.54	68.40	1.97	0.66	49.50	1.15	0.54
A7	6	15	42	63.60	0.51	0.34	50.2	0.20	0.16	67.90	0.62	0.38	49.30	0.17	0.15
A7	7	23	51	69.10	0.35	0.26	64.7	0.27	0.21	68.20	0.34	0.25	52.40	0.03	0.03
A8	4	26	91	71.17	0.22	0.28	69.3	0.24	0.31	68.80	0.24	0.32	60.80	0.33	0.50
A8	5	17	93	64.98	0.30	0.43	50.2	0.46	0.85	67.80	0.27	0.37	49.20	0.47	0.89
A8	6	19	66	66.35	0.01	0.01	54.3	0.18	0.22	67.80	0.03	0.03	49.10	0.26	0.34
A8	7	13	62	62.23	0.00	0.00	50.2	0.19	0.24	68.10	0.10	0.09	49.40	0.20	0.26
A9	4	11	34	60.85	0.79	0.44	50.2	0.48	0.32	68.40	1.01	0.50	49.50	0.46	0.31
A9	5	12	62	61.54	0.01	0.01	50.2	0.19	0.24	68.20	0.10	0.09	49.40	0.20	0.26
A9	6	12	39	61.54	0.58	0.37	50.2	0.29	0.22	68.20	0.75	0.43	49.40	0.27	0.21
A9	7	10	30	60.17	1.01	0.50	50.2	0.67	0.40	68.60	1.29	0.56	49.50	0.65	0.39
A10	4	60	58	94.54	0.63	0.39	97.6	0.68	0.41	92.20	0.59	0.37	97.70	0.68	0.41
A10	5	45	145	84.23	0.42	0.72	83.5	0.42	0.74	79.30	0.45	0.83	81.50	0.44	0.78
A10	6	49	80	86.98	0.09	0.08	86.2	0.08	0.07	83.00	0.04	0.04	85.20	0.07	0.06
A10	7	81	59	108.98	0.85	0.46	115	0.95	0.49	119.00	1.02	0.50	116.00	0.97	0.49
B1	4	50	83	87.66	0.06	0.05	87	0.05	0.05	83.90	0.01	0.01	86.20	0.04	0.04
B1	5	93	68	117.23	0.72	0.42	127	0.87	0.46	137.00	1.01	0.50	127.00	0.87	0.46
B1	6	143	122	151.60	0.24	0.20	153	0.25	0.20	160.00	0.31	0.24	150.00	0.23	0.19
B1	7	91	93	115.85	0.25	0.20	125	0.34	0.26	134.00	0.44	0.31	125.00	0.34	0.26
B2	4	36	100	78.04	0.22	0.28	77.9	0.22	0.28	72.50	0.28	0.38	75.50	0.25	0.32
B2	5	17	65	64.98	0.00	0.00	50.2	0.23	0.29	67.80	0.04	0.04	49.20	0.24	0.32
B2	6	117	150	133.72	0.11	0.12	153	0.02	0.02	159.00	0.06	0.06	150.00	0.00	0.00
B2	7	42	100	82.16	0.18	0.22	81.6	0.18	0.23	76.60	0.23	0.31	79.60	0.20	0.26
B3	4	75	104	104.85	0.01	0.01	109	0.05	0.05	111.00	0.07	0.06	110.00	0.06	0.05

Tabla 21. (continuación)

D	P	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
B3	5	71	92	102.10	0.11	0.10	106	0.15	0.13	105.00	0.14	0.12	106.00	0.15	0.13
B3	6	79	128	107.60	0.16	0.19	113	0.12	0.13	117.00	0.09	0.09	114.00	0.11	0.12
B3	7	157	123	161.22	0.31	0.24	153	0.24	0.20	160.00	0.30	0.23	150.00	0.22	0.18
B4	6	81	155	108.98	0.30	0.42	115	0.26	0.35	119.00	0.23	0.30	116.00	0.25	0.34
B4	7	119	168	135.10	0.20	0.24	153	0.09	0.10	159.00	0.05	0.06	150.00	0.11	0.12
B5	4	54	134	90.41	0.33	0.48	90.5	0.32	0.48	87.20	0.35	0.54	90.20	0.33	0.49
B5	5	28	85	72.54	0.15	0.17	71.6	0.16	0.19	69.30	0.18	0.23	65.00	0.24	0.31
B5	6	103	171	124.10	0.27	0.38	138	0.19	0.24	101.00	0.41	0.69	139.00	0.19	0.23
B5	7	97	131	119.97	0.08	0.09	131	0.00	0.00	143.00	0.09	0.08	132.00	0.01	0.01
B6	4	37	123	78.73	0.36	0.56	78.5	0.36	0.57	73.10	0.41	0.68	76.30	0.38	0.61
B6	5	16	64	64.29	0.00	0.00	50.2	0.22	0.27	67.90	0.06	0.06	49.20	0.23	0.30
B6	6	49	159	86.98	0.45	0.83	86.2	0.46	0.84	83.00	0.48	0.92	85.20	0.46	0.87
B6	7	123	195	137.85	0.29	0.41	153	0.22	0.27	160.00	0.18	0.22	150.00	0.23	0.30
B7	4	41	121	81.48	0.33	0.49	81	0.33	0.49	75.90	0.37	0.59	79.00	0.35	0.53
B7	6	81	129	108.98	0.16	0.18	115	0.11	0.12	119.00	0.08	0.08	116.00	0.10	0.11
B8	4	29	49	73.23	0.49	0.33	72.6	0.48	0.33	69.60	0.42	0.30	66.80	0.36	0.27
B8	5	29	98	73.23	0.25	0.34	72.6	0.26	0.35	69.60	0.29	0.41	66.80	0.32	0.47
B8	6	182	141	178.41	0.27	0.21	154	0.09	0.08	159.00	0.13	0.11	150.00	0.06	0.06
B8	7	63	113	96.60	0.15	0.17	102	0.10	0.11	95.10	0.16	0.19	103.00	0.09	0.10
B9	4	75	115	104.85	0.09	0.10	109	0.05	0.06	111.00	0.03	0.04	110.00	0.04	0.05
B9	5	44	97	83.54	0.14	0.16	82.8	0.15	0.17	78.40	0.19	0.24	80.60	0.17	0.20
B9	6	100	153	122.04	0.20	0.25	135	0.12	0.13	147.00	0.04	0.04	135.00	0.12	0.13
B9	7	121	144	136.47	0.05	0.06	153	0.06	0.06	159.00	0.10	0.09	150.00	0.04	0.04
B10	4	79	133	107.60	0.19	0.24	113	0.15	0.18	117.00	0.12	0.14	114.00	0.14	0.17
B10	5	51	70	88.35	0.26	0.21	87.8	0.25	0.20	84.80	0.21	0.17	87.20	0.25	0.20
B10	6	112	92	130.29	0.42	0.29	152	0.65	0.39	157.00	0.71	0.41	150.00	0.63	0.39
B10	7	84	78	111.04	0.42	0.30	118	0.51	0.34	124.00	0.59	0.37	118.00	0.51	0.34
CFE1	5	153	145	158.47	0.09	0.09	153	0.06	0.05	160.00	0.10	0.09	150.00	0.03	0.03
CFE1	6	89	125	114.48	0.08	0.09	123	0.02	0.02	131.00	0.05	0.05	123.00	0.02	0.02
CFE1	7	22	82	68.42	0.17	0.20	62.7	0.24	0.31	68.10	0.17	0.20	48.90	0.40	0.68
CFE2	4	17	56	64.98	0.16	0.14	50.2	0.10	0.12	67.80	0.21	0.17	49.20	0.12	0.14
CFE2	5	77	124	106.23	0.14	0.17	111	0.10	0.12	114.00	0.08	0.09	112.00	0.10	0.11
CFE2	6	96	134	119.29	0.11	0.12	130	0.03	0.03	141.00	0.05	0.05	130.00	0.03	0.03
CFE2	7	17	63	64.98	0.03	0.03	50.2	0.20	0.25	67.80	0.08	0.07	49.20	0.22	0.28
CFE3	4	33	70	75.98	0.09	0.08	75.9	0.08	0.08	71.00	0.01	0.01	72.40	0.03	0.03
CFE3	5	125	124	139.22	0.12	0.11	153	0.23	0.19	160.00	0.29	0.23	150.00	0.21	0.17
CFE3	6	58	99	93.16	0.06	0.06	94.9	0.04	0.04	90.50	0.09	0.09	94.80	0.04	0.04
CFE3	7	35	100	77.35	0.23	0.29	77.3	0.23	0.29	72.00	0.28	0.39	74.50	0.26	0.34
CFE4	5	75	124	104.85	0.15	0.18	109	0.12	0.14	111.00	0.10	0.12	110.00	0.11	0.13
CFE4	6	32	92	75.29	0.18	0.22	75.2	0.18	0.22	70.60	0.23	0.30	71.20	0.23	0.29
CFE4	7	20	59	67.04	0.14	0.12	57.6	0.02	0.02	67.90	0.15	0.13	49.00	0.17	0.20
CFE5	4	15	58	63.60	0.10	0.09	50.2	0.13	0.16	67.90	0.17	0.15	49.30	0.15	0.18
CFE5	5	84	185	111.04	0.40	0.67	118	0.36	0.57	124.00	0.33	0.49	118.00	0.36	0.57
CFE5	6	51	53	88.35	0.67	0.40	87.8	0.66	0.40	84.80	0.60	0.38	87.20	0.65	0.39
CFE6	4	13	33	62.23	0.89	0.47	50.2	0.52	0.34	68.10	1.06	0.52	49.40	0.50	0.33
CFE6	5	129	88	141.97	0.61	0.38	153	0.74	0.42	160.00	0.82	0.45	150.00	0.70	0.41
CFE6	6	30	92	73.92	0.20	0.24	73.5	0.20	0.25	69.90	0.24	0.32	68.40	0.26	0.35



**Tabla 22.** ANOVA de la (a) MRE y (b) MER de modelos

(a)

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F	Valor-p
Entre grupos	0.243301	3	0.0811003	1.12	0.3398
Dentro de grupos	30.0658	416	0.072273		
Total	30.3091	419			

(b)

Fuente	Suma de cuadrados	Grados de Libertad	Cuadrados medios	F	Valor-p
Entre grupos	0.059242	3	0.01974	0.56	0.64
Dentro de grupos	14.6038	416	0.03510		
Total	14.663	419			

Esto es, que con base en la prueba  $F$ , ninguno de los modelos resultó significativamente mejor que el otro.

La prueba de rangos múltiples para la MRE y MER por cada modelo indica cuál de los modelos tiene el mejor resultado. En la Tabla 23 se aplica un procedimiento de comparación múltiple para determinar qué medias tienen diferencia significativa sobre las otras. El método usado para discriminación entre medias es el procedimiento de la diferencia mínima significativa de Fisher (*least significant difference, LSD*). En la Tabla 23, cada uno de los valores absolutos de la columna “diferencia” es menor que su valor de LSD. Esto asimismo refuerza el hecho de que los modelos no son significativamente diferentes.

**Tabla 23.** Prueba de rangos múltiples de la (a) MRE y (b) MER

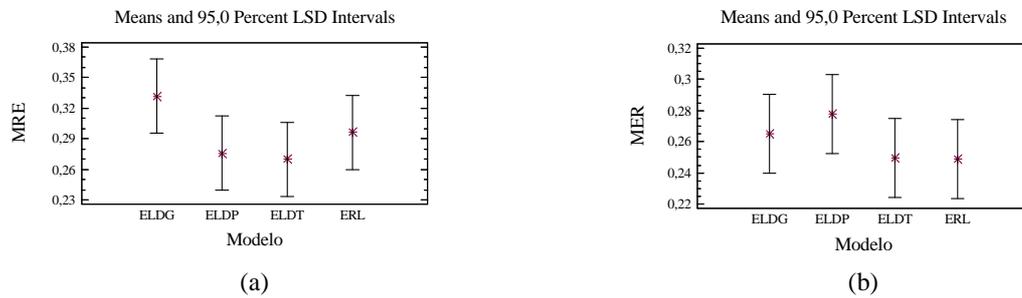
(a)

Modelo	MMRE	MMRE		
		Contraste	Diferencia	Valor de LSD
ERL	0,296095	ELD – ELDP	0.055714	0.072933
ELDT	0,270095	ELD – ELDT	0.061523	0.072933
ELDG	0,331619	ELD – ERL	0.035523	0.072933
ELDP	0,275905	ELDP – ELDT	0.005809	0.072933
		ELDP – ERL	-0.020190	0.072933
		ELDT – ERL	-0.026000	0.072933

(b)

Modelo	MMER	MMER		
		Contraste	Diferencia	Valor de LSD
ERL	0.248952	ELD – ELDP	-0.0124762	0.0508301
ELDT	0.249714	ELD – ELDT	0.0154286	0.0508301
ELDG	0.265143	ELD – ERL	0.0161905	0.0508301
ELDP	0.277619	ELDP – ELDT	0.0279048	0.0508301
		ELDP – ERL	0.0286667	0.0508301
		ELDT – ERL	0.0007615	0.0508301

Las gráficas de medias de la Tabla 23, se muestran en las Figura 10.



**Figura 10.** Gráficas de medias (a) MRE y (b) MER

Estos resultados, permiten concluir que la segunda hipótesis nula formulada en este trabajo (sección 1.5.3, inciso b) se acepta: *tras la verificación de los modelos lineal y difusos, no existe diferencia significativa entre las medias de las MER de los cuatro modelos.*

## Capítulo 4

### Resultados

#### 4.1 Procedimiento para la validación de los modelos

Probablemente durante los pasados tres decenios se hayan elegido en algunos casos los modelos incorrectos con base en el criterio de la MMRE (un reciente estudio así lo demuestra [32]). Además de ello, otro estudio relevante ha demostrado que las MRE pueden ser dependientes de la duración de los programas [120].

Estos dos importantes estudios están reflejados en la validación de los modelos de esta investigación:

- 1) La MMRE no será el criterio principal para la comparación. La MMRE se calcula con la finalidad de poder hacer comparaciones con otros estudios previos. De acuerdo con [32] se empleará la MMRER como criterio de comparación entre modelos.
- 2) La muestra destinada a la validación de los modelos deberá dividirse en submuestras en caso de que sus MRE y MER resultaran dependientes de la duración (esfuerzo) de los programas [120]. Esta dependencia puede identificarse mediante un análisis de heterocedasticidad de los datos dentro de un diagrama de dispersión.

#### 4.2 Obtención de las MRE y MER

La muestra de programas para validar los modelos (mostrada en la Tabla 24) se generó considerando los criterios A, B y C descritos en la sección 3.2.

En la Tabla 25 los modelos de regresión lineal y difusos son aplicados a los datos de la Tabla 24. Después son calculados sus MRE y MER por programa. En la Tabla 25 la duración de los programas ha sido deliberadamente colocada en orden ascendente.

**Tabla 24.** Muestra de programas para validación de los modelos

	D	P	Código		Esfuerzo ( <i>en minutos</i> )						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
1	U1	4	86	0	29	0	23	0	5	48	105
2	U1	5	75	0	37	0	67	0	1	25	130
3	U1	7	56	0	48	0	67	0	5	40	160
4	U2	4	69	0	21	0	20	2	1	23	67
5	U2	5	107	0	47	0	56	1	10	27	141
6	U2	7	53	0	30	0	55	1	1	35	122
7	U3	5	92	7	26	0	50	0	1	100	177
8	U3	6	27	8	14	0	30	0	1	27	72
9	U4	4	78	0	10	0	50	0	10	30	100
10	U4	5	155	0	20	0	120	0	10	30	180

Tabla 24. (continuación)

	D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total
			NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba	
11	U4	6	29	0	15	0	30	0	10	20	75
12	U4	7	40	0	15	0	45	0	10	20	90
13	U5	5	57	0	10	0	30	10	5	20	75
14	U5	6	50	0	10	0	20	5	5	25	65
15	U5	7	55	0	10	0	15	5	5	20	55
16	U6	4	55	0	18	0	19	0	7	10	54
17	U6	5	53	0	16	0	22	0	16	8	62
18	U6	6	60	0	17	0	26	0	8	9	60
19	U7	4	13	20	7	0	32	0	3	7	49
20	U7	6	86	4	8	0	104	0	2	7	121

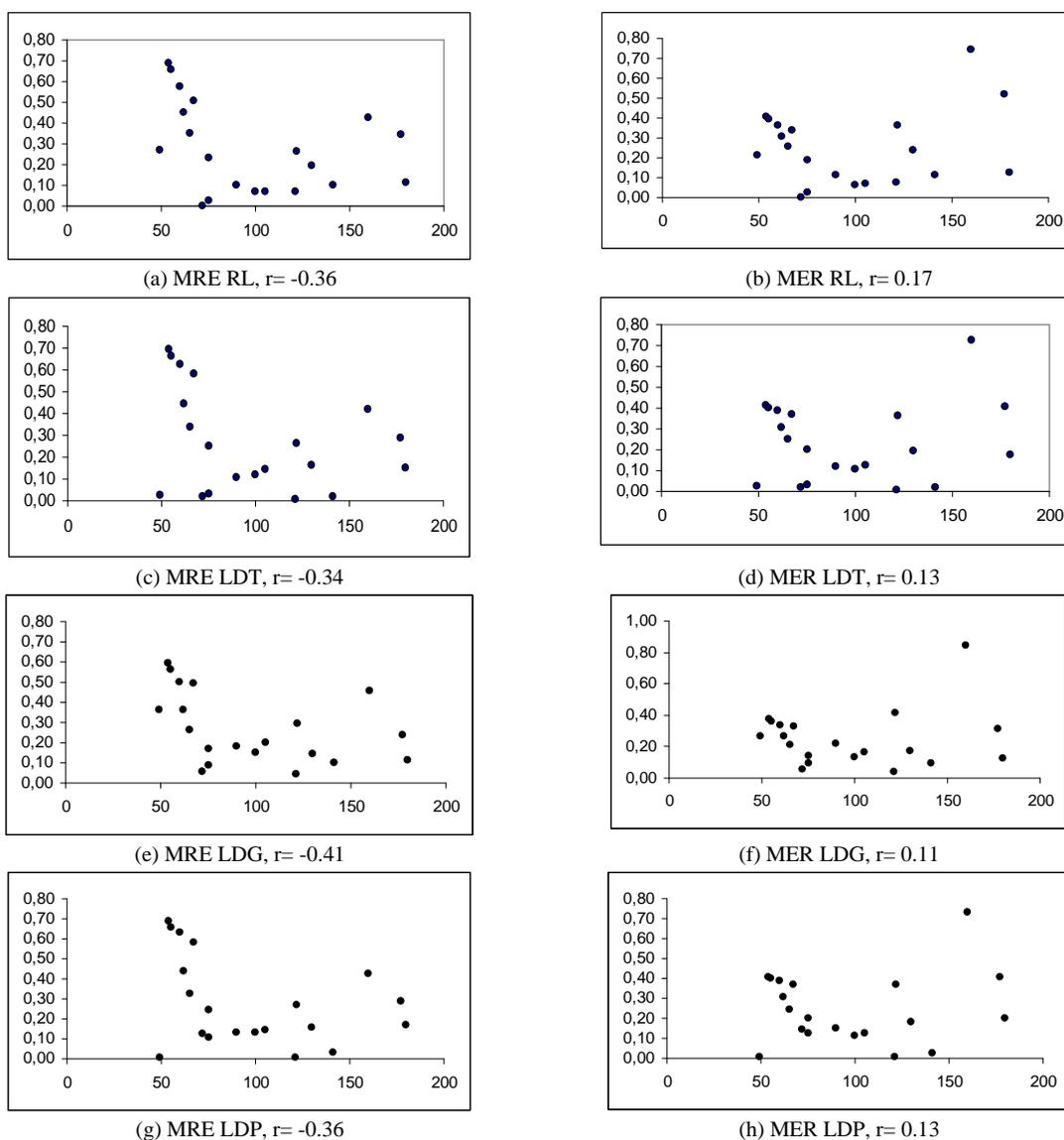
Tabla 25. MRE y MER por programa para validación de modelos (D: Desarrollador, P: Programa, ER: Esfuerzo Real; ERL: Esfuerzo por Regresión Lineal; ELDT: Esfuerzo por Lógica Difusa Triangular; ELDG: Esfuerzo por Lógica Difusa Gaussiana; ELDP: Esfuerzo por Lógica Difusa Trapezoidal)

	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
1	13	49	62.23	0.27	0.21	50.20	0.02	0.02	66.80	0.36	0.27	49.40	0.01	0.01
2	55	54	91.10	0.69	0.41	91.50	0.69	0.41	86.00	0.59	0.37	91.20	0.69	0.41
3	55	55	91.10	0.66	0.40	91.50	0.66	0.40	86.00	0.56	0.36	91.20	0.66	0.40
4	60	60	94.54	0.58	0.37	97.60	0.63	0.39	90.10	0.50	0.33	97.70	0.63	0.39
5	53	62	89.73	0.45	0.31	89.60	0.45	0.31	84.50	0.36	0.27	89.30	0.44	0.31
6	50	65	87.66	0.35	0.26	87.00	0.34	0.25	82.10	0.26	0.21	86.20	0.33	0.25
7	69	67	100.73	0.50	0.33	106.00	0.58	0.37	100.00	0.49	0.33	106.00	0.58	0.37
8	27	72	71.85	0.00	0.00	70.60	0.02	0.02	68.10	0.05	0.06	63.00	0.13	0.14
9	29	75	73.23	0.02	0.02	72.70	0.03	0.03	68.60	0.09	0.09	66.80	0.11	0.12
10	57	75	92.48	0.23	0.19	93.70	0.25	0.20	87.60	0.17	0.14	93.50	0.25	0.20
11	40	90	80.79	0.10	0.11	80.40	0.11	0.12	73.80	0.18	0.22	78.40	0.13	0.15
12	78	100	106.91	0.07	0.06	112.00	0.12	0.11	115.00	0.15	0.13	113.00	0.13	0.12
13	86	105	112.41	0.07	0.07	120.00	0.14	0.13	126.00	0.20	0.17	120.00	0.14	0.13
14	86	121	112.41	0.07	0.08	120.00	0.01	0.01	126.00	0.04	0.04	120.00	0.01	0.01
15	53	122	89.73	0.26	0.36	89.60	0.27	0.36	86.40	0.29	0.41	89.30	0.27	0.37
16	75	130	104.85	0.19	0.24	109.00	0.16	0.19	111.00	0.15	0.17	110.00	0.15	0.18
17	107	141	126.85	0.10	0.11	144.00	0.02	0.02	155.00	0.10	0.09	145.00	0.03	0.03
18	56	160	91.79	0.43	0.74	92.60	0.42	0.73	86.80	0.46	0.84	92.30	0.42	0.73
19	92	177	116.54	0.34	0.52	126.00	0.29	0.40	135.00	0.24	0.31	126.00	0.29	0.40
20	155	180	159.85	0.11	0.13	153.00	0.15	0.18	160.00	0.11	0.13	150.00	0.17	0.20

MMRE | 0,27 | | 0,27 | | 0,27 | | 0,28 |  
MMER | 0,25 | | 0,23 | | 0,25 | | 0,25 |

En la Tabla 25 se calculan asimismo los valores de la MMRE (éste sólo como referencia para otros estudios) y la MMER; en ellos se observa un resultado que favorece al modelo difuso con funciones de membresía triangular (debido a que muestra la menor MMER con 0.23). Para tener una comparación más justa se requiere un análisis de dependencia entre la MER y el esfuerzo (homocedasticidad/heterocedasticidad).

Una vez que los cuatro modelos han sido aplicados, en la Figura 11 se muestran ocho gráficas de dispersión que muestran la relación entre la MRE y MER con el esfuerzo. Las Figuras 11a, 11c, 11e y 11g sugieren que la MRE decrece con el esfuerzo, es decir presentan una notoria heterocedasticidad (en oposición a homocedasticidad); mientras que las Figuras 11b, 11d, 11f y 11h correspondientes a las MER presentan heterocedasticidad con menor grado que las MRE. De hecho, una de las justificaciones para hacer uso de la MER como criterio de comparación, es precisamente que es menos sensible a la duración pequeña del esfuerzo [32] y en este estudio se ha comprobado (con los valores de correlación  $r$  de la Figura 11).



**Figura 11.** Gráficas de dispersión (MRE y MER versus esfuerzo) y sus valores de correlación ( $r$ )

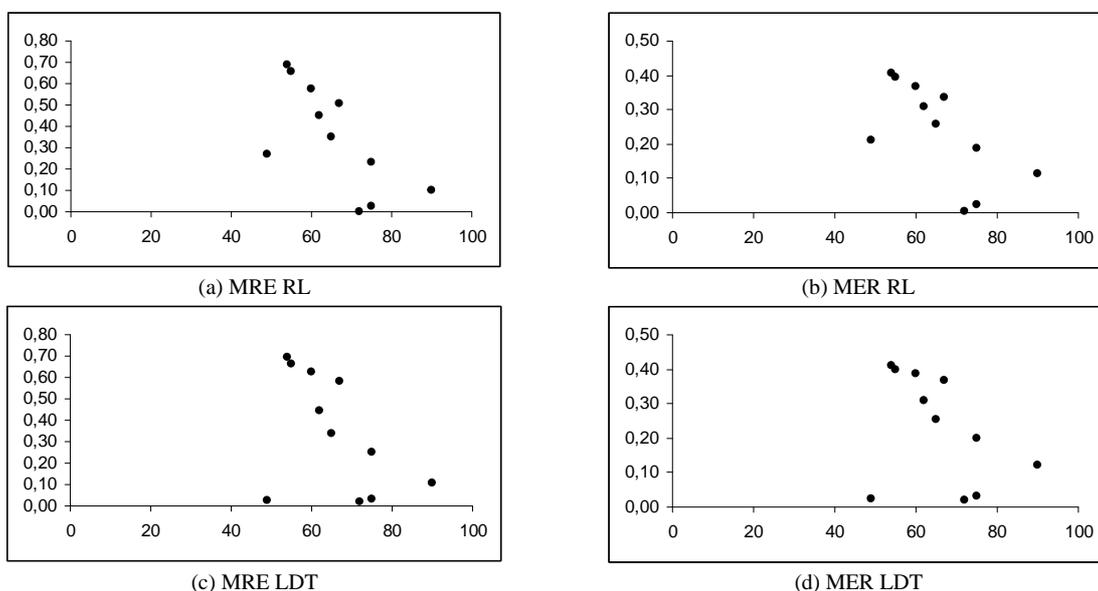
Con base en inspecciones visuales y valores de correlación  $r$  de las gráficas de la Figura 11, el conjunto de datos debe entonces ser dividido dado que presenta heterocedastidad. Se observa que 100 LDC es un punto medio para los casos. Entonces se forman dos submuestras A y B por modelo. La submuestra A contendrá aquellos programas con esfuerzo menor a 100 minutos (once programas) y la submuestra B corresponderá a los programas mayores o iguales que 100 minutos (nueve programas).

Para que la comparación entre submuestras sea más justa, en la Tabla 26 las medias de cada una de las submuestras se compararon para asegurar que no haya diferencia significativa entre ellas [120].

**Tabla 26.** Comparación de medias entre submuestras A y B.

Técnica	Criterio	t	Valor-p
RL	MRE	1.823	0.0848
	MER	-0.224	0.8251
LDT	MRE	1.692	0.1078
	MER	-0.087	0.9313
LDG	MRE	1.8158	0.0860
	MER	-0.1644	0.8712
LDP	MRE	1.9549	0.0662
	MER	0.1086	0.9146

La Tabla 26 muestra (por los valores-p) que no existen diferencias significativas entre las medias de las submuestras. Se puede inferir que la división permitirá una comparación más justa entre los modelos con base en este análisis estadístico aunada a una inspección visual de las Figuras 12 y 13, donde se muestran los diagramas de dispersión tras la división de la muestra en submuestras A y B.



**Figura 12.** Gráficas de dispersión de submuestras A

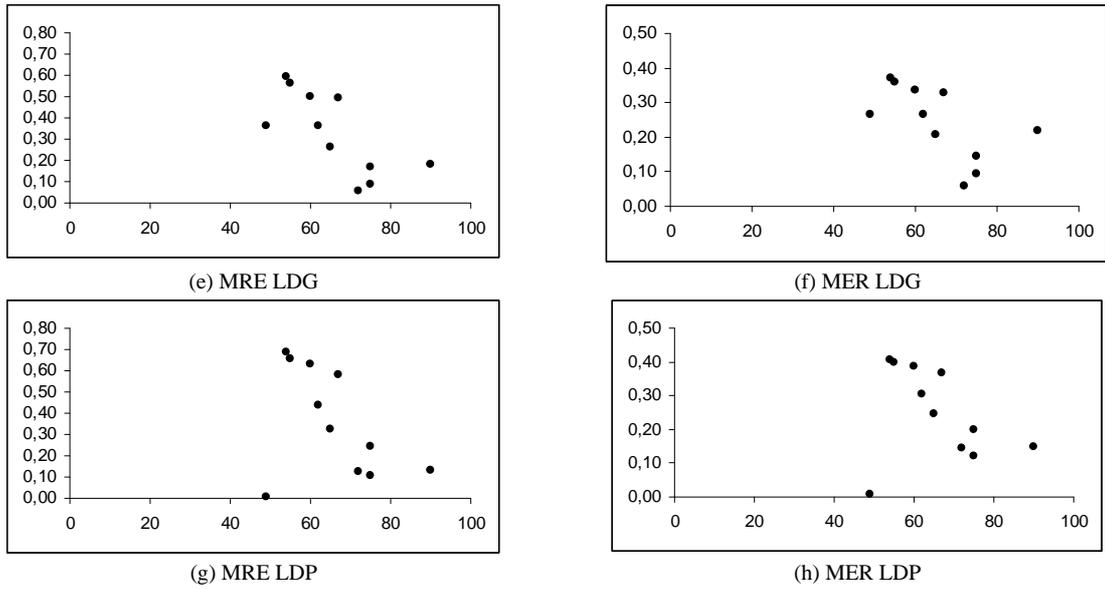


Figura 12. (continuación)

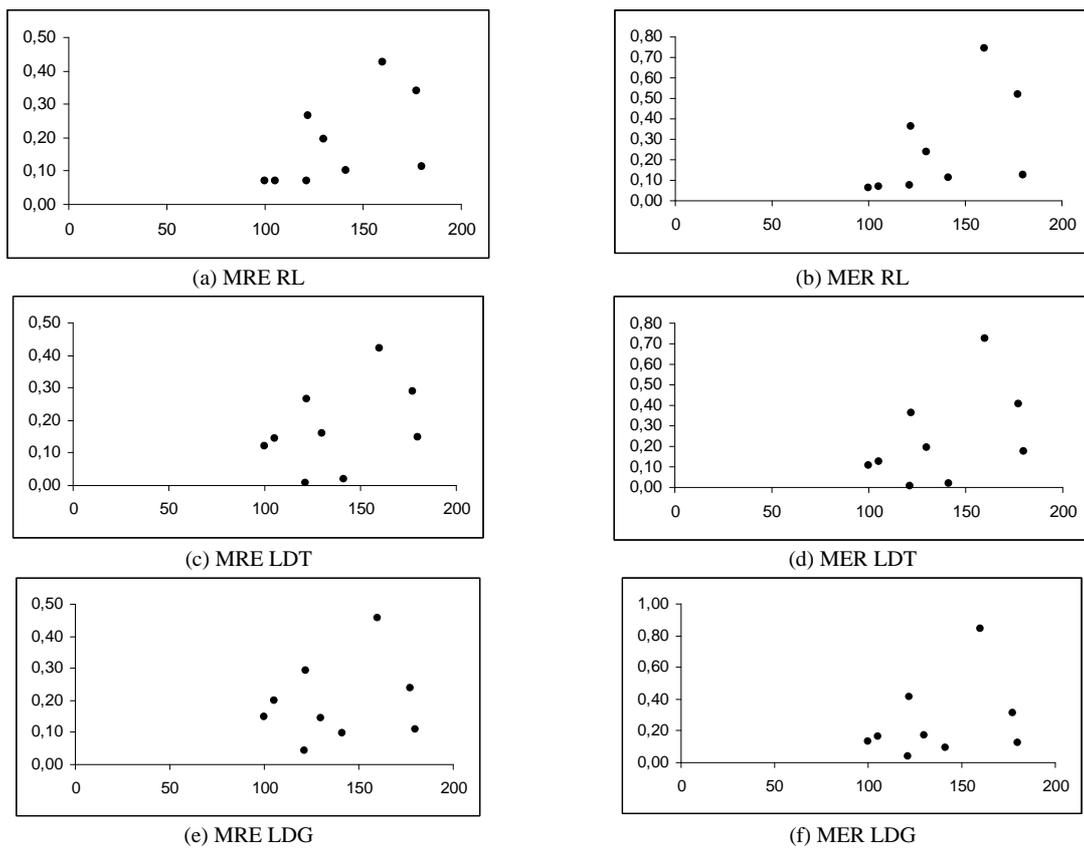


Figura 13. Gráficas de dispersión de submuestras B

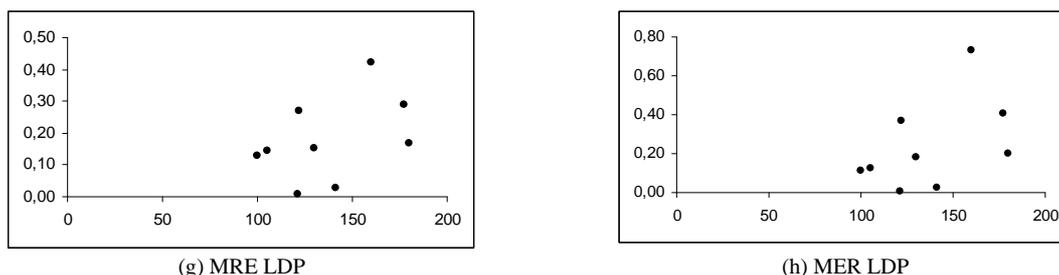


Figura 13. (continuación)

La Tabla 27 muestra los cálculos de los criterios de evaluación MMRE y MMR por submuestra. Se observa que el sistema difuso con funciones de membresía triangular tiene el mismo valor (MMER = 0.23) para ambas submuestras A y B. En la submuestra A representa el menor valor, mientras que en la submuestra B es sólo igualado por el modelo difuso con funciones trapezoidales.

Tabla 27. MMRE y MMR por submuestras A (a) y B (b)

(a)

	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
1	13	49	62.23	0.27	0.21	50.20	0.02	0.02	66.80	0.36	0.27	49.40	0.01	0.01
2	55	54	91.10	0.69	0.41	91.50	0.69	0.41	86.00	0.59	0.37	91.20	0.69	0.41
3	55	55	91.10	0.66	0.40	91.50	0.66	0.40	86.00	0.56	0.36	91.20	0.66	0.40
4	60	60	94.54	0.58	0.37	97.60	0.63	0.39	90.10	0.50	0.33	97.70	0.63	0.39
5	53	62	89.73	0.45	0.31	89.60	0.45	0.31	84.50	0.36	0.27	89.30	0.44	0.31
6	50	65	87.66	0.35	0.26	87.00	0.34	0.25	82.10	0.26	0.21	86.20	0.33	0.25
7	69	67	100.73	0.50	0.33	106.00	0.58	0.37	100.00	0.49	0.33	106.00	0.58	0.37
8	27	72	71.85	0.00	0.00	70.60	0.02	0.02	68.10	0.05	0.06	63.00	0.13	0.14
9	29	75	73.23	0.02	0.02	72.70	0.03	0.03	68.60	0.09	0.09	66.80	0.11	0.12
10	57	75	92.48	0.23	0.19	93.70	0.25	0.20	87.60	0.17	0.14	93.50	0.25	0.20
11	40	90	80.79	0.10	0.11	80.40	0.11	0.12	73.80	0.18	0.22	78.40	0.13	0.15
	<b>MMRE</b>			0.35			0.34			0.33			0.36	
	<b>MMER</b>				0.24			<b>0.23</b>			0.24			0.25

(b)

	NyC	ER	EERL	MRE	MER	EELT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
1	78	100	106.91	0.07	0.06	112.00	0.12	0.11	115.00	0.15	0.13	113.00	0.13	0.12
2	86	105	112.41	0.07	0.07	120.00	0.14	0.13	126.00	0.20	0.17	120.00	0.14	0.13
3	86	121	112.41	0.07	0.08	120.00	0.01	0.01	126.00	0.04	0.04	120.00	0.01	0.01
4	53	122	89.73	0.26	0.36	89.60	0.27	0.36	86.40	0.29	0.41	89.30	0.27	0.37
5	75	130	104.85	0.19	0.24	109.00	0.16	0.19	111.00	0.15	0.17	110.00	0.15	0.18
6	107	141	126.85	0.10	0.11	144.00	0.02	0.02	155.00	0.10	0.09	145.00	0.03	0.03
7	56	160	91.79	0.43	0.74	92.60	0.42	0.73	86.80	0.46	0.84	92.30	0.42	0.73
8	92	177	116.54	0.34	0.52	126.00	0.29	0.40	135.00	0.24	0.31	126.00	0.29	0.40
9	155	180	159.85	0.11	0.13	153.00	0.15	0.18	115.00	0.15	0.13	113.00	0.13	0.12
	<b>MMRE</b>			0.18			0.18			0.20			0.17	
	<b>MMER</b>				0.26			<b>0.23</b>			0.25			<b>0.23</b>

La Tabla 28 muestra el criterio principal de comparación de este estudio: MMER (en negrita), además de aquellos otros criterios complementarios. Con relación a las medianas de la MER, las sumuestras A de los modelos difusos con funciones de membresía triangular y trapezoidal presentaron mejores valores y en la submuestra B el modelo de regresión muestra mejor valor (*en cursiva*). Basándose en el criterio de pred(25), los modelos difusos resultaron para todos los casos mejores o iguales que la regresión lineal (*en cursiva*).

**Tabla 28.** Resumen de criterios de evaluación y comparación de modelos (a) MMRE y MMER, (b) Mediana y (c) Pred(25)

(a)

Submuestra	MMRE				MMER			
	LR	LDT	LDG	LDP	LR	LDT	LDG	LDP
A	0.35	0.34	0.33	0.36	0.24	<b>0.23</b>	0.24	0.25
B	0.18	0.18	0.20	0.17	0.26	<b>0.23</b>	0.25	<b>0.23</b>

(b)

Submuestra	Mediana							
	MdMRE				MdMER			
	LR	LDT	LDG	LDP	LR	LDT	LDG	LDP
A	0.35	0.34	0.36	0.33	0.26	0.25	0.27	0.25
B	0.11	0.15	0.15	0.15	0.13	0.18	0.17	0.18

(c)

Submuestra	Pred(25)							
	MRE				MER			
	LR	LDT	LDG	LDP	LR	LDT	LDG	LDP
A	0.36	0.45	0.36	0.45	0.45	0.55	0.45	0.55
B	0.67	0.67	0.78	0.67	0.67	0.67	0.67	0.67

Estos resultados finales sobre la validación de los modelos, permiten concluir que la tercera y última hipótesis nula formulada en este trabajo (sección 1.5.3, inciso c) se rechaza, aceptándose la siguiente: *al término de la validación de los modelos, en la predicción de programas a nivel personal haciendo uso de un modelo difuso, se obtiene una MMER  $\leq 0.25$ .*

El modelo que mostró mejores resultados en su predicción (con base en su MMER) es el modelo difuso con funciones de membresía triangular. Esto por las siguientes tres razones:

- Sin considerar heterocedasticidad en los datos, presentó la menor MMER (con 0.23) que el resto de los tres modelos (dos difusos y uno de regresión).
- Al dividirse el conjunto de datos en dos submuestras (A y B), tras la demostración de la existencia de heterocedasticidad, en la submuestra A presentó mejor MMER (con 0.23), mientras que en la submuestra B sólo fue igualado por otro modelo difuso con funciones trapezoidales.
- Considerando o no la heterocedasticidad, la regresión lineal nunca fue mejor que el modelo difuso con base en su MMER.

## Capítulo 5

### Conclusiones, recomendaciones, trabajo futuro y legado de la investigación

#### 5.1 Conclusiones

Esta investigación estuvo fundamentada en los siguientes hechos:

- a) Desde el origen de la ingeniería del software en 1968, la estimación del esfuerzo de desarrollo de software ha sido una actividad relevante. Incluso, esta ha sido identificada como uno de los tres grandes desafíos de la ciencia computacional.
- b) La estimación del esfuerzo de desarrollo de software es considerada tanto en normas nacional (MoProSoft) e internacionales (CMM, SPICE) de procesos como en cuerpos de conocimiento para certificaciones y modelos curriculares (SWEBOK, ANIEI).
- c) El Plan de Desarrollo de la Industria del Software, derivado del Plan Nacional de Desarrollo de México, tiene como referencia de crecimiento al CMM.
- d) Más del 90% de las empresas desarrolladoras de software en México carece de prácticas para administrar proyectos.
- e) Antes de integrarse a equipos, un profesionista del desarrollo de software debería tener un entrenamiento personal.
- f) El Proceso de Software Personal (PSP) permite aplicar a nivel individual, al menos 12 de las 18 áreas clave de proceso del CMM.
- g) Con el antecedente de que ninguna técnica de estimación del esfuerzo de desarrollo de software es mejor para todas las situaciones, es importante proponer nuevos modelos que permitan comparar sus resultados y entonces generar estimaciones más realistas.

Adicionalmente, uno de los objetivos de esta investigación fue el de reunir datos propios y no acudir a alguna base de datos conteniendo ingentes historiales de mediciones obtenidas de proyectos en los que no constaban los detalles de cómo ellos habían sido obtenidos.

Los modelos producto de esta investigación involucraron a cincuenta y dos programadores divididos en cuatro grupos y quienes en total desarrollaron trecientos cincuenta y cuatro programas dentro de un diseño de investigación cuasiexperimental que se llevó a cabo dentro de un entorno continuamente controlado. Similar a aquel proceso que se sigue para pulir un diamante, de igual forma fueron siendo excluidos de la muestra aquellos programas que por sus características especiales, redujeran la confiabilidad de esta investigación. Una vez conformada la muestra, los datos de los programas superaron diversas pruebas estadísticas tanto para la comprobación de la adecuación de los modelos (verificación) como para su validación.

Diversos criterios fueron considerados para concluir la utilidad de los datos y los modelos a partir de los cuales se generaron. Para medir la precisión de los modelos, se prefirió

---

el uso de la MMER en lugar del criterio que por decenios probablemente ha permitido elecciones erróneas: la MMRE. Adicionalmente, para la validación de los modelos fue considerada la heterocedasticidad de los datos.

Tres hipótesis de investigación (con sus correspondientes nulas) fueron formuladas al principio de esta investigación, siendo aceptadas las siguientes:

- A partir de una población, la muestra extraída posee una correlación ( $r$ ) mayor o igual a 0.71 entre las líneas de código y el esfuerzo, es decir, un  $r^2 \geq 0.50$ .
- En la verificación de los modelos lineal y difusos, no existe diferencia significativa entre las medias de las MER de los cuatro modelos.
- En la validación de los modelos, en la predicción de programas a nivel personal haciendo uso de un modelo difuso, se obtiene una MMER  $\leq 0.25$ .

Siendo aún más específico en la tercera viñeta, aparte de que el modelo difuso triangular logró una MMER de 0.23 en ambas submuestras A y B, superó asimismo al modelo de regresión lineal que obtuvo una MMER de 0.24 y 0.26 para las muestras A y B, respectivamente.

Con relación a los objetivos planteados en esta tesis, se logró el general al generar un sistema difuso integrado por reglas difusas, conjuntos difusos y funciones de membresía, derivadas tanto de tamaño (en líneas de código) como de esfuerzo de desarrollo (en minutos/persona), de programas elaborados con base en un proceso común y dentro de un diseño cuasiexperimental, con la finalidad de aportar una alternativa para la estimación del esfuerzo de desarrollo de software a nivel personal.

Asimismo, las aportaciones de esta tesis se describen con base en los objetivos específicos al proponer un modelo para la estimación del esfuerzo de desarrollo de software a nivel personal basado en lógica difusa, extensible a otras instancias del ciclo de vida del software; proveer a la industria y a la academia de un modelo que permita iniciar la práctica de la estimación del esfuerzo a nivel personal y, en lo sucesivo, un mejor desempeño en equipos de trabajo dentro de una organización; al proponer un modelo que permita comparar sus resultados con otros modelos y al demostrar que la aplicación de la lógica difusa a la estimación del esfuerzo de desarrollo tiene resultados iguales o mejores que el modelo más empleado hasta ahora: la regresión estadística.

De igual forma se propone un proceso que permitirá generar a las empresas sus propios modelos de estimación con base en lógica difusa, de tal manera que permita su verificación y validación dentro de otro entorno de desarrollo de software. Estos modelos les permitirán calendarizar sus actividades y elaborar sus presupuestos, administrar sus riesgos y analizar su mejoramiento de inversión, al mismo tiempo que evitaría la deficiencia en la calidad de los productos, insatisfacción en clientes y frustración en los desarrolladores. Todo ello basándose en otra alternativa adicional al juicio experto o algún modelo algorítmico que estuvieran empleando.

El aporte científico del proceso propuesto para la generación de modelos de estimación basados en lógica difusa está relacionado en lo general con la ingeniería del software y específicamente con la estimación del esfuerzo de desarrollo.

## 5.2 Recomendaciones

El modelo difuso triangular debería ser aplicado a nuevos entornos, de tal forma que permita su continua validación y ajuste (calibración) a otros entornos locales. Al aplicarlo, es importante considerar las características bajo las que este modelo fue generado, tales como los tipos de programas, el proceso seguido, así como el nivel de estudios y la consistencia en su disciplina. Sin estas características, sería difícilmente confiable el modelo que fuera generado y desde luego injusta su comparación con otros modelos.

Por otra parte, también debería considerarse que el modelo difuso producto de esta investigación fue generado de 105 programas codificados en lenguajes de programación correspondientes al paradigma imperativo (excluyendo a todo aquel declarativo) y respetando un estándar establecido al principio de cada curso, así como cuantificado de manera homogénea.

Las estructuras de las versiones de los lenguajes imperativos empleados fueron similares (declaración de variables, definición de constantes directivas de compilador, estructuras de control de flujo secuenciales, selectivas y repetitivas). La reunión futura de más datos podría generar modelos de estimación por paradigmas declarativos e incluso por tipo particular de lenguaje (tanto para lenguajes imperativos como para declarativos), como en [105].

En los anexos E, F, G y H se proporcionan los valores que los modelos de regresión lineal y aquellos tres difusos generan respectivamente. El rango en líneas de código es de 10 a 182 debido a que los modelos en esta investigación, fueron generados dentro de este intervalo. Estas tablas son recomendables para ser consultadas directamente por programadores (previa estimación del tamaño en líneas de código nuevas y cambiadas) para predecir el esfuerzo de desarrollo de sus programas. De igual forma, estas tablas podrían ser consultadas por diseñadores de exámenes de programación para estimar su duración.

Con base en los datos del esfuerzo de los 105 programas de este estudio, podría estimarse la duración de un método contenido en alguna clase (considerando la equivalencia de cada programa de esta muestra con un método de clase), con ello, podría igualmente estimarse la duración del esfuerzo de desarrollo de determinado caso de uso identificado en la etapa de especificación de requerimientos (mismo que pudiera incluir una o más clases). El esfuerzo de cada método ( $E_m$ ) podría obtenerse al calcular la media del esfuerzo de los  $n$  programas:

$$E_m = \frac{\sum_{i=1}^n (EP_i)}{n}$$

Donde  $EP_i$  es el esfuerzo real del  $i$ -ésimo programa

Para el caso de la muestra de este trabajo (105 programas), se tiene el siguiente resultado:

$$E_m = \frac{\sum_{i=1}^n (EP_i)}{n} = \frac{9455}{105} = 90 \text{ minutos}$$

Es decir, que si se tuviera un caso de uso que implicara una clase con tres métodos, se esperaría que el esfuerzo del desarrollo de ese caso de uso fuera de 270 minutos, considerando una media de 53 líneas de código por programa obtenida de esta muestra de 105 programas.

De igual forma, dependiendo del tamaño (en líneas de código) de cada método, podrían consultarse los esfuerzos estimados en los anexos E, F, G y H para calcular el esfuerzo necesario por método.

### 5.3 Trabajo Futuro

El trabajo futuro debería involucrar la generación de otros modelos basados en lógica difusa haciendo uso de otras muestras de datos de proyectos a nivel personal, a nivel equipo y luego organizacional (ajustando las reglas difusas y parámetros de las funciones de membresía propuestas en esta investigación). Otras funciones de membresía deberían también ser utilizadas.

Asimismo, cuando se desarrollen programas de mayor tamaño y complejidad, podría considerarse la incorporación de factores adicionales que influyen en el desarrollo de software.

En este trabajo, tras un análisis estadístico se descartó considerar al código reusado como variable independiente; si en el futuro pudieran ser recogidos más datos, probablemente este tipo de código pudiera llegar a ser significativo y entonces considerarlo. La complejidad ciclomática de McCabe e incluso el acoplamiento de Dhama podrían también ser considerados como variables independientes cuando los programas fueran más grandes. De igual forma, podrían ser empleadas otras unidades de medida como casos de uso, clases u operaciones de clases.

De igual forma, el diseño de algoritmos para la generación automática de reglas y conjuntos difusos, podría contribuir al único trabajo encontrado durante esta investigación [126].

### 5.4 El legado de la Investigación

Durante el Primer Foro de la Industria del Software en Jalisco en junio del 2003; evento que reunió empresas desarrolladoras de software, academia y gobierno de Jalisco, el director de una empresa comentaba que a ella solían acudir aspirantes con dominio notable en determinado lenguaje de programación; sin embargo, al momento de someterlos a procesos les costaba mucho la adaptación, es decir, debido a que su formación académica solía ser con ausencia de prácticas personales, esos hábitos los llevaban a las empresas y les era difícil seguir procesos de desarrollo de software. Este hecho se vio reflejado en esta investigación: muchos programas fueron excluidos debido que a sus creadores les fue en algunos casos desesperante llevar el control de su propio trabajo y las inconsistencias de sus anotaciones lo reflejaban; incluso había quien abandonaba el curso al ver que *lo suyo no era eso*, o que ellos *no requerían seguir procesos formales para desarrollar software libre de defectos*. Sin embargo, la mayoría de los participantes en verdad disfrutaba observar su desempeño de manera cuantitativa. Al momento de aplicar tópicos estadísticos como el análisis de correlación o el de regresión lineal, muchos de ellos expresaban su descontento por no haber aprendido la materia de estadística dentro del contexto de la computación. Este hecho apoyaba

una investigación que el autor de esta tesis realizó y presentó en un congreso nacional en la que enfatizaba la necesidad de enseñar estadística aplicándola a la ingeniería de software [65] y una más igualmente presentada en que se proponen algunos tópicos estadísticos para la equivalencia de líneas de código y puntos de función [64].

Fue notorio el cambio de actitud que los desarrolladores iban teniendo conforme se desarrollaban los programas. Durante el primer programa, el someterse al registro de tiempos, tamaño y defectos de sus programas no era muy convincente para los programadores. Sin embargo, en el segundo y tercer programas notaron por ellos mismos que se trataba de un hábito más que no les consumía demasiado tiempo. Más aún, convencidos quedaban al ver los beneficios en su mejora de estimación de tamaño y tiempos, así como la minimización de los defectos.

Relevante asimismo fue la confirmación de que la calidad de esta investigación iba siendo directamente proporcional a las publicaciones aceptadas en congresos nacionales e internacionales en los que el autor de esta tesis aparecía (trece aceptados hasta el momento y dos más en revisión). Fue valiosa la continua incorporación de cada vez más tópicos que fundamentaran esta tesis (incluyendo retroalimentación de revisores). Al término de esta tesis, se lograron dos publicaciones [73][74] reconocidas por la institución de indexación científica ISI-Thomson.

Un logro paralelo al desarrollo de esta obra, fue que a lo largo del desarrollo de la misma y tras el envío de un artículo, en marzo del 2006 el autor de esta tesis fue invitado como revisor (del artículo FSS-D-06-0038) por parte del Dr. Fernando Comide, miembro del consejo editorial de la revista *ISI Fuzzy Sets and Systems* editada por Elsevier<sup>1</sup>, en donde el Dr. Lotfi A. Zadeh es miembro honorario y tanto el Dr. E. H. Mamdani y el Dr. M. Sugeno fungen como asesores. El artículo a revisar combinaba la lógica difusa y la estimación del esfuerzo de desarrollo de software.

La continua asesoría y cuestionamiento por parte de experimentados investigadores, fue fundamental para el logro de este trabajo. El aprendizaje logrado por las continuas dudas expuestas a base de preguntas concretas, fue la base para formular la siguiente analogía: así como los aplausos son el alimento de un artista, las preguntas lo son para un investigador. El empeñarse por dar una respuesta bien fundamentada, más que corresponder a una preocupación y evadirla, se ha convertido en un placer personal.

En 1972 Zadeh mencionaba que la lógica difusa podría abrir muchas fronteras nuevas en psicología, sociología, ciencias políticas, filosofía, fisiología, economía, lingüística, investigación de operaciones, administración y otros dominios; con el cumplimiento de la meta de esta tesis podría adicionarse a su lista, que la lógica difusa es asimismo útil para predecir el esfuerzo de desarrollo de software.

---

<sup>1</sup> *Fuzzy Sets and Systems*, Elsevier: [http://www.elsevier.com/wps/find/journaleditorialboard.cws\\_home/505545/editorialboard](http://www.elsevier.com/wps/find/journaleditorialboard.cws_home/505545/editorialboard)

## Referencias

- [1] Agarwal R., Kumar M., Mallick Y. M., Bharadwaj R.M., Anantwar D. Estimating Software Projects. ACM SIGSOFT. Software Engineering Notes. vol. 26 no. 4. 2001
- [2] Ahmed M. A, Saliu M.O., AlGhamdi J. Adaptive fuzzy logic-based framework for software development effort prediction. Information and Software Technology, Vol. 47, No. 1. 2004
- [3] Anderson D. R., Sweeney D. J., Williams T. A. Estadística. Thomson. 1999
- [4] ANIEI, INEGI. Modelos Curriculares, Nivel Licenciatura, Informática y Computación. México 1997
- [5] Appleby D., Vandekopple J.J. Lenguajes de Programación, Paradigma y Práctica. McGrawHill. 1998
- [6] Bagert D. J., Hilburn T. B., Hislop G., Lutz M., McCracen M., Mengel S. Guidelines for Software Engineering Education. Version 1.0. CMU/SEI-99-TR-032. Carnegie Mellon University. Software Engineering Institute. 1999.
- [7] BANCOMEEXT y AMITI. Esquema de apoyo gubernamental a la industria de software. 2001
- [8] Boehm B. COCOMO II. Prentice Hall. 2000
- [9] Boehm B. Software Engineering Economics. Prentice Hall. 1981
- [10] Boehm B. W., Failey R. E. Software Estimation Perspectives. IEEE Software November/December 2000.
- [11] Boehm B. W., Sullivan K. J. Software Engineering Economics. Future of Software Engineering Limerick Ireland. ACM. 2000
- [12] Boehm B., Abts Ch., Chulani S. Software Development Cost Estimation Approaches – A Survey. Chulani Ph. D. Report. 1998
- [13] Boetticher G. D. An Assessment of Metric Contribution in the Construction of a Neural Network-Based Effort Estimator. Second International Workshop on Soft Computing. ACM. 2001
- [14] Boetticher G. D. Using Machine Learning to Predict Project Effort: Empirical Case Studies in Data-Starved Domains. Proceedings of the Model Based Requirements Workshop. ACM. 2001
- [15] Briand I.C., Emam K.E., Surmann D., Wiczorek I., Maxwell K. D. An Assesment and Comparison of Common Software Cost Estimation Modeling Techniques. ISERN98-27. 1998.
- [16] Briand L.C., Langley T., Wiczorek I. A replicated Assessment and Comparison of Common Software Cost Modeling Techniques, *IEEE International Conference on Software Engineering (ICSE)*, Limerick, Ireland. 2000.
- [17] Brooks, F. P. Jr. Three Great Challenges for Half-Century-Old Computer Science. Journal of the ACM, Vol. 50, No. 1 pp. 25-26. January 2003
- [18] Burgess C. J., Lefley M. Can genetic programming improve software effort estimation? A comparative evaluation. Information and Software Technology. Elsevier. 2001
- [19] Cerpa N., Otoy S. Comparing Estimated with Actual Efforts for a Dynamic Methodology used by a Small Software Company. Proceedings of the World Multiconference on Systems. ACM. 2001
- [20] Chulani S., Boehm B., Steece B. Bayesian Analysis of Empirical Software Engineering Cost Models. USC-CSE-1999.
- [21] Chulani S., Boehm B., Steece B. Calibrating Software Cost Models Using Bayesian Analysis. USC-CSE-1998.
- [22] Chulani S., Steece B. A Bayesian Software Estimating Model using a generalized g-Prior Approach. Technical Report. University of Southern California, Los Angeles 1998
- [23] CMMI for Software Engineering, Version 1.1. CMU/SEI-2002-TR-029. 2002
- [24] Conte S.D., Dunsmore H.E., Shen V.Y. Software Engineering Metrics and Models. Benjamin/Cummings. M. Park CA. 1986
- [25] Cordero R., Costramagna M., Paschetta. A Genetic Algorithm Approach for the Calibration of COCOMO-like Models. !2th COCOMO Forum. 1997
- [26] Delany S. J., Cunningham P. The application of Case-Based Reasoning to Early Software project Cost Estimation and Risk Assessment. Department of Computer Science, Trinity College Dublin, TDS-CS-2000-10, 2000
- [27] Delany S. J., Cunningham P., Wilke W. The Limits of CBR in Software Project Estimation. German Workshop on Case-Based Reasoning. 1998
- [28] Digital Planet: The Global Information Economy. WITSA. Noviembre de 2000

## Referencias (continuación)

- [29] Failey D. Making Accurate Estimates. IEEE Software. Special Report. November/December 2002. pp. 61-63
- [30] Fenton N. (1994): Software Measurement A Necessary Scientific Basis, IEEE Transactions on software engineering, Vol. 20, No. 3, March, pp. 199-206
- [31] Finnie G. R., Wittig G. E. AI Tools for Software Development Effort Estimation. IEEE 1996.
- [32] Foss T., Stensrud E., Kitchenham B, Myrtveit I. A Simulation Study of the Model Evaluation Criterion MMRE. IEEE Transactions on Software Engineering. Vol. 29, No. 11. 2003
- [33] Gray A. R. A Simulation-Based Comparison of Empirical Modeling Techniques for Software Metrics Models of Development Effort. 6th International Conference on Neural Information Processing. 1999.
- [34] Gray A. R., MacDonell S. G. Applications of Fuzzy Logic to Software Metric Models for Development Effort Estimation, Proceedings of NAFIPS. 1997
- [35] Hernández Sampieri R., Fernández Collado C., Baptista Lucio P. Metodología de la Investigación. Tercera Edición. Mc Graw Hill. 2003.
- [36] Hihn J. Cost Estimation of Software Intensive projects: A survey of Current Practices. IEEE 1991.
- [37] Hilburn T., Hirmanpour I., Khajenoori S., Turner R., Qasem A. A Software Engineering Body of Knowledge, Versión 1.0. Carnegie Mellon. CMU/SEI-99-TR-004.1999
- [38] Höst M., Wohlin C. A subjective effort estimation experiment. Information and Software Technology. Elsevier. 1997
- [39] Huang X., Capretz L.F., Ren J., Ho D. A Neuro-Fuzzy Model for Software Cost Estimation. Preceedings of the Third International Conference on Quality Software. 2003
- [40] Humphrey W. A Discipline for Software Engineering. Addison Wesley. 1995.
- [41] Humphrey W. Introducción al Proceso Software Personal. Addison Wesley. 2001
- [42] Humphrey, W. S. The Personal Software Process (PSP), Technical Report, CMU/SEI-2000-TR-022. 2000
- [43] Humphrey, W. S. The Team Software Process (PSP), Technical Report, CMU/SEI-2000-TR-023. 2000
- [44] Idri A., Abran A., Kjiri L. COCOMO Cost Model Using Fuzzy Logic. 7<sup>th</sup> International Conference on Fuzzy Theory & Technology. 2000
- [45] Idri A., Khoshgoftaar T. M., Abran A. Can Neural Networks be Easily Interpreted in Software Cost Estimation. World Congress on Computational Intelligence. Hawaii. 2002
- [46] Idri, A., Abran A., Khoshgoftaar T. Estimating Software Project Effort by Analogy Based on Linguistic Values. Eight IEEE Symposium on Software Metrics. 2002
- [47] Idri, A., Abran, A., Khoshgoftaar T. M. Computational Intelligence in Empirical Software Engineering, First USA-Morocco Workshop on Information Technology , ENSIAS, Rabat. 2003
- [48] IEEE Computer Society, ACM, Software Engineering Body of Knowledge (SWEBOK). 2001. Disponible: <http://www.swebok.org/>
- [49] IEEE Standard Glossary of Software Engineering Terminology. IEEE.1990
- [50] Jorgensen M., Kirkeboen G., Sjoberg D., Anda B., Brathall L. Human Judgement in Effort Estimation of Software Projects, *International Conference on Software Engineering*, Limerick, Ireland. 2000
- [51] Jorgensen M., Practical Guidelines for Better Software Effort Estimation, Simula Research Laboratory. Available (2005): [www.simula.no/departments/engineering/.artifacts/ieee\\_jorgensen\\_guidelines5.pdf](http://www.simula.no/departments/engineering/.artifacts/ieee_jorgensen_guidelines5.pdf)
- [52] Jun E. S., Lee J. K. Quasi-optimal case-selective neural network model for software effort estimation. Expert Systems and Applications. PERGAMON. 2001
- [53] Kadoda G., Cartwright M., Chen L., Shepperd M. Experiences Using Case-Based Reasoning to Predict Software Project Effort. Proceedings of the EASE Conference Keele, UK. 2000
- [54] Kaufmann A. Introducción a la teoría de los subconjuntos borrosos, Elementos teóricos de base. C.E.C.S.A. 1982
- [55] Kemerer C. F. An Empirical Validation of Software Cost Estimation Models. Communications of the ACM. Vol. 30 No. 5. 1997.
- [56] Kirsopp, Shepperd M., Hart J. Search Heuristics, Case-Based reasoning and Software project Effort Prediction. GECCO. 2002

## Referencias (continuación)

- [57] Kitchenham B. A., Pfleeger S. L., Pickard L.M., Jones P. W., Hoaglin D. C., Emam K.E., Rosenberg J. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on SE, Vol. 28, No. 8. August 2002
- [58] Kitchenham B.A., MacDonell S.G., Pickard L.M., Shepperd M.J. What Accuracy Statistics Really Measure. IEE Proceedings Software. 148(3). pp. 81-85. 2001
- [59] Kok P., Kitchenhan B.A, Kirakowski J. The MERMAID Approach to Software Cost Estimation. Proceedings ESPRIT Technical week. 1990
- [60] Leung H., Software Cost Estimation. <http://www.st.cs.uni-sb.de/edu/empirical-se/2006/PDFs/leung.pdf> Disponible junio 21, 2006
- [61] Lewis J.P. Large Limits to Software Estimation, *ACM Software Engineering Notes Vol 26, No. 4*, July, 2001.
- [62] Lind D. A., Mason R. D., Marchal W. G. Estadística. McGraw Hill. 2001
- [63] López Martín Cuauhtémoc. Knowledge Areas Priorities for Software Engineering Mexican Education. Congreso Interuniversitario de Electrónica, Computación y Eléctrica (CIECE-IEEE). Guadalajara, Jalisco. 2004
- [64] López Martín Cuauhtémoc. Lines of Code as a Source for Function Point Estimation Using Linear Regression and Correlation. XVI Congreso Nacional y II Internacional de Informática y Computación. Guadalupe, Zacatecas. 2003.
- [65] López Martín Cuauhtémoc. The Need of Applied Statistics Teaching for the Software Engineering Education in Mexico. Congreso Interuniversitario de Electrónica, Computación y Eléctrica (CIECE-IEEE). Guadalajara, Jalisco, 2004
- [66] López-Martín Cuauhtémoc, Ellerbracke-Román Sergio, Mijes Elba. Ingeniería de Software en México: Necesidad de un Currículum Basado en Competencias. XVI Congreso Nacional y II Internacional de Informática y Computación 2003. Guadalupe, Zacatecas, octubre del 2003
- [67] López-Martín Cuauhtémoc, Felipe-Riverón Edgardo. A Comparison of Calibrated Equations for Software Development Effort Estimation. XVIII Congreso Nacional y IV Internacional de Informática y Computación. Torreón, Coahuila, octubre del 2005
- [68] López-Martín Cuauhtémoc, Gutiérrez-Tornés Agustín. Software Effort Estimation: A Designed Process for Structured and Object Oriented Software Engineering Approaches. 11° Congreso Internacional de Investigación en Ciencias Computacionales, Tlalnepantla, Estado de México, México, septiembre- octubre del 2004
- [69] López-Martín Cuauhtémoc, Leboeuf-Pasquier Jerome, Gutiérrez-Tornés Agustín, Olivares Ceja Jesús. Software Development Effort Estimation Using A General Regression Neural Network: A Case Study. 12° Congreso Internacional de Investigación en Ciencias Computacionales. Monterrey, Nuevo León, noviembre, 2005
- [70] López-Martín Cuauhtémoc, Leboeuf-Pasquier Jerome, Yáñez-Márquez Cornelio, Gutiérrez-Tornés Agustín. Software Development Effort Estimation Using Fuzzy Logic: A Case Study. Encuentro Internacional en Ciencias de la Computación, ENC 2005. IEEE Computer Society Press. Puebla, Puebla. Septiembre del 2005
- [71] López-Martín Cuauhtémoc, Yáñez-Márquez Cornelio, Gutiérrez-Tornés Agustín, Edgardo Felipe-Riveron. Adequacy Checking of Personal Software Development Effort Estimation Models Based upon Fuzzy Logic: A Replicated Experiment. Enviado para revision a la Revista Computación y Sistemas. Centro de Investigación en Computación, Instituto Politécnico Nacional, México, Distrito Federal. 2006
- [72] López-Martín Cuauhtémoc, Yáñez-Márquez Cornelio, Gutiérrez-Tornés Agustín. A Fuzzy Logic Model Based Upon Reused and New&Changed Code for Software Development Effort Estimation at Personal Level. Congreso Internacional en Computación, Centro de Investigación en Computación, Instituto Politécnico Nacional, México, Distrito Federal, Noviembre del 2006

## Referencias (continuación)

- [73] López-Martín Cuauhtémoc, Yáñez-Márquez Cornelio, Gutiérrez-Tornés Agustín. A Fuzzy Logic Model for Software Development Effort Estimation at Personal Level. Mexican International Conference Artificial Intelligence. November 2006. LNAI 4293, pp. 122 – 133, Springer-Verlag Berlin Heidelberg 2006
- [74] López-Martín Cuauhtémoc, Yáñez-Márquez Cornelio, Gutiérrez-Tornés Agustín. Software Fuzzy Logic Systems for Software Development Effort Estimation Based Upon Clustering of Programs Segmented by Personal Practices. Electronics, Robotics and Automotive Mechanics Conference, CERMA IEEE Computer Society Press. Cuernavaca, México, 2006. Vol II. Pp. 367-372
- [75] López-Martín Cuauhtémoc. A Personal Empirical Study Encouraging the Use of Personal Software Process Inside Universities. XVII Congreso Nacional y III Congreso Internacional de Informática y Computación, Tepic, Nayarit, octubre del 2004.
- [76] López-Martín Cuauhtémoc. Software Module Effort Estimation using Simple and Multiple Regression Based on Defects, Dhama Coupling and McCabe Complexity. Congreso Internacional en Computación, Centro de Investigación en Computación, Instituto Politécnico Nacional, México, Distrito Federal, octubre del 2003
- [77] MacDonell S. G. Software source code sizing using fuzzy logic modelling. Information and Software Technology. Elsevier. 2003
- [78] MacDonell S. G., Gray A. R. (1996): Alternatives to Regression Models for Estimating Software Projects, Proceedings of the IFPUG Fall Conference, Dallas TX
- [79] Mair C., Kadoda G., Lefley M., Phalp K., Shofield C. An Investigation of Machine Learning Based Prediction Systems. Journal Systems and Software. 2000
- [80] MATLAB®, The Language of Technical Computing. Version 6.1.0.450. Release 12.1. 2001
- [81] Maxwell K., Wassenhove I. V., Dutta S. Performance Evaluation of General and company Specific Models in Software Development Effort Estimation. Management Science, Vol. 45, No. 6. 1999
- [82] Meli R. Human factors and analytical models in software estimation: an integration perspective. ESCOM-SCOPE. 2000
- [83] Mendes E., Counsell S., Mosley N. Towards the Prediction of Development Effort for Hypermedia Applications. ACM. ISBN 1-59113-420-7/01/0008. 2001
- [84] Mendes E., Hall W. Towards the Prediction of Development Effort for Web Applications. ACM. ISBN 1-58113-227-1/00/0005. 2000
- [85] Mendes E., Mosley N. Comparing Effort prediction Models for Web Design and Authoring using Boxplots. IEEE. 2001
- [86] Mendes E., Mosley N., Counsell S. Do Adaptation Rules Improve Web Cost Estimation. ACM 2003.
- [87] Mendes E., Mosley N., Watson I., A Comparison of Case-Based Reasoning Approaches to Web Hypermedia Project Cost Estimation. ACM. 2002
- [88] Meyer B. Software Engineering in the Academy. IEEE Computer. 2001. pp. 28-35
- [89] Mockus A. Analogy Based Prediction of Work Item Flow in Software Projects. Symposium on Empirical Software Engineering. Italy. 2003
- [90] Molokken K., Jorgensen M. A review of surveys on Software effort Estimation. Symposium on Empirical Software Engineering. Italy. 2003
- [91] Molokken K., Jorgensen M. A., Tanilkan S.S., Gallis H., Lien A. C., Hove S.E. A survey on Software Estimation in the Norwegian Industry. Proceedings of 10th International Software Metrics. 2004
- [92] Molokken K., Jorgensen M. Group Processes in Software Effort Estimation. Empirical Software Engineering. 2004
- [93] Molokken K., Jorgensen M. Software Effort Estimation: Unstructured Group Discussion as a Method to reduce Individual Biases. Workshop of the Psychology of Programming Interest Group. 2003
- [94] Montgomery D., Peck E., Vining G. Introduction to linear regression analysis. John Wiley. 2001.
- [95] MoProSoft. <http://www.software.net.mx>. Disponible: Junio 2006

## Referencias (continuación)

- [96] Munson J. C. Software Engineering Measurement. Auerbach Publications. 2003
- [97] Musflek P., Pedrycz W., Succi G. Reformat M. (2000), Software Cost Estimation with Fuzzy Models, Applied Computing Review, Vol. 8, No. 2, pages 24-29.
- [98] Naur P., Randell B., Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968. January 1969.
- [99] Oppenheimer A. Cuentos Chinos. Plaza Janes. 2005
- [100] Passing U. A pilot experiment on software project size and effort estimation. Proceedings of International Symposium on Empirical Software Engineering, ISESE. 2003
- [101] Passing U., Strahringer S. Process-oriented effort estimation of software projects. European Conference on Software Process Improvement. Proceedings EuroSPI. 2002
- [102] Paulk M., Weber Ch., Garcia S., Chrissis M.B., Bush M. Key Practices of the Capability Maturity Model Version 1.1. CMU/SEI-93-TR-025. 1993
- [103] Pedrycz W., Computational Intelligence as an Emerging Paradigm of Software Engineering. SEKE ACM. 2002
- [104] Pedrycz W., Gomide F. An Introduction to Fuzzy Sets. The MIT Press. 1998
- [105] Peeples M. A. Software Development Cost Estimation Model for Higher Level Language Environments. [http://www.sba.muohio.edu/abas/2003/vancouver/peeples\\_cosmo4gl%20abas03.pdf](http://www.sba.muohio.edu/abas/2003/vancouver/peeples_cosmo4gl%20abas03.pdf) Disponible: Junio 21, 2006
- [106] Pressman R. S., Ingeniería de *Software*, Un Enfoque Práctico. McGraw Hill. 2005.
- [107] Reaman C.B. Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transactions On Software Engineering, Vol. 25, No. 4, 1999
- [108] Reformat M., Pedrycz W., Pizzi N. Building a software experience factory using granular-based models. Fuzzy Sets and Systems. Elsevier. 2004.
- [109] Ruhe M., Jeffery R., Weiczorek I. Cost Estimation for Web Applications. IEEE 2003
- [110] Ryder J. Fuzzy modeling of Software Effort Prediction. Information Technology Conference. IEEE. 1998
- [111] Schofield C. (1998): Non-Algorithmic Effort Estimation Techniques, *ESERG*, TR98-01
- [112] Secretaría de Economía, (2002): Programa para el Desarrollo de la Industria del Software, <http://www.economia.gob.mx/pics/p/p1128/Texto.pdf>, disponible: junio del 2006
- [113] Shan Y., McKay R.I., Lokan C.J., Essam D.L. Software project Effort Estimation Using Genetic Programming. International Conference on Communications, Circuits and Systems and West Sino Expositions. IEEE. 2002
- [114] Shepperd M., Schofield C. Estimating Software Project Effort Using Analogies. IEEE transactions on Software Engineering, vol. 23, núm 12. 1997
- [115] Sjoberg D.I.K., Anda B., Arisholm E., Dyba T., Jorgensen M., Karahasanovic A., Koren E. F., Vokác M. Conducting Realistic Experiments in Software Engineering. Proceedings of the International Symposium on Empirical Software Engineering (ISESE). 2002
- [116] Software Process Improvement and Capability dTermination. SPICE. <http://www.sqi.gu.edu.au/SPICE/> Disponible: Junio 21, 2006
- [117] Sommerville I., Ingeniería de *Software*. Séptima Edición. Pearson. 2005.
- [118] Srinivasan K. Machine Learning Approaches to Estimating Software Development Effort. IEEE Transactions on Software Engineering. Vol. 21, No. 2. 1995
- [119] Statgraphics Plus for Windows. Versión 4.0. Statistical Graphics Corporation. 1999.
- [120] Stensrud E, Foss T, Kitchenham B, Myrtveit I. An Empirical Validation of the Relationship Between the Relative Error and Project Size. Eighth IEEE Software Metrics Symposium. 2002
- [121] Stutzke R. D. Estimating Software-Intensive Systems, Projects, Products and Processes. SEI Series in Software Engineering. Addison Wesley. 2005
- [122] UNCTAD, Changing Dynamics of Global Computer Software and Services Industry: Implication for Developing Countries. Naciones Unidas, 2001.

## Referencias (continuación)

- [123] Vesterinen P. Issues in Calibrating Effort Estimation Models. Software Engineering Notes. Vol 24, No. 3. ACM SIGSOFT, 1999
- [124] Weimer C. R. Estadística. CECSA. 2002.
- [125] Weiss N.A. Introductory Statistics. Addison Wesley. 1999.
- [126] Xu Z., Khoshgoftaar T. M. Identification of fuzzy models of software cost estimation. Elsevier Fuzzy Sets and Systems. Volume 145, Issue 1, Pages 141-163. 2004
- [127] Zadeh L. A. Fuzzy Logic, Neural Networks, and Soft Computing. Communications of ACM. Vol. 37, No. 3. March 1994
- [128] Zhang D. Applying Machine Learning Algorithms in Software Development. Proceedings of Monterey Workshop on Modeling Software Systems. 2000

# Anexo A Formas de Registro

## Sumario del Plan del Proyecto

Desarrollador \_\_\_\_\_ Fecha \_\_\_\_\_  
Programa \_\_\_\_\_

Tamaño del Programa (LDC)	Plan	Actual	A la fecha	Porcentaje a la fecha
Base (B)				
Eliminado (E)				
Modificado (M)				
Adicionado (A)				
Reusado (R)				
Nuevo y Cambiado (N)				
Total (T)				

### Tiempo por Fase (min.)

Planeación				
Diseño				
Revisión de diseño				
Codificación				
Revisión de código				
Compilación				
Pruebas				
Postmortem				
Total				

### Defectos introducidos

Planeación				
Diseño				
Revisión de diseño				
Codificación				
Revisión de código				
Compilación				
Pruebas				
Total				

### Defectos Eliminados

Planeación				
Diseño				
Revisión de diseño				
Codificación				
Revisión de código				
Compilación				
Pruebas				
Total				

### Descripción de Líneas de Código:

Base (B)                    Tamaño del programa(s) previo(s), antes de ser modificado por el nuevo programa.  
 Eliminado (E)            Código base que es eliminado dentro del nuevo programa  
 Modificado (M)          Código base que es modificado dentro del nuevo programa  
 Adicionado (A)          Código escrito para el nuevo programa que es adicionado al programa base  
 Reusado (R)              Código base que es utilizado, sin modificación alguna, dentro del nuevo programa  
 Nuevo y Cambiado (N y C)    Código adicionado más el modificado  
 Total (T)                  Tamaño en LDC final del programa.

### Descripción de tiempos:

Planeación            Entendimiento de los requerimientos del programa y tiempo de estimación de fases siguientes (diseño a postmortem)  
 Diseño                Algoritmo representado ya sea en diagrama de flujo o pseudocódigo  
 Codificación          Traducción del diseño a código respetando estándar de codificación  
 Compilación          Corrección de defectos detectados por el compilador  
 Pruebas                Corrección de defectos no detectados por el compilador. Asegurarse que el programa cumple con los requerimientos.  
 Postmortem          Registro de datos reales en el Sumario del Plan del Proyecto.

## Anexo A Formas de Registro (continuación)

### Registro de Tiempos

Desarrollador \_\_\_\_\_ Fecha \_\_\_\_\_  
Programa \_\_\_\_\_

Fecha	Fase	Inicio	Término	Tiempo de Interrupción	Tiempo Delta	Comentarios

#### Descripción de Registro de Tiempos:

Fecha	Día, mes y año del registro del programa
Fase	Fase (planeación...postmortem) que se registrará
Inicio	Hora en que se inicio o reinició la fase
Término	Hora en que se concluyó parcial o totalmente la fase
Tiempo de interrupción	Lapso que duró una interrupción dentro de una fase o lapso de esta
Tiempo Delta	Tiempo efectivo de una fase o lapso de esta ( <i>tiempo delta = fin - inicio - tiempo de interrupción</i> )
Comentarios	Anotaciones que se consideren relevantes alusivas al registro (motivo de interrupción, por ejemplo)

### Registro de Defectos

#### Tipos de Defectos

Número	Nombre	Descripción
10	Documentación	Comentarios, mensajes
20	Sintaxis	Ortografía, puntuación, formato de instrucciones
30	Construcción, Paquetes	Bibliotecas, control de versión
40	Asignación	Declaración, nombres duplicados, límites
50	Interfaz	Llamadas a procedimientos y referencias, E/S, formatos de usuario
60	Chequeo	Mensajes de error
70	Datos	Estructura
80	Función	Lógica, punteros, bucles, recursión
90	Sistema	Memoria, sistema
100	Entorno	Diseño, compilación, pruebas y otros problemas que soporta el sistema

Desarrollador \_\_\_\_\_ Fecha \_\_\_\_\_  
Programa \_\_\_\_\_

Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
/ /						

Descripción:

Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
/ /						

Descripción:

## Anexo A

### Formas de Registro

*(continuación)*

#### Descripción de Registro de Defectos:

Fecha	Día, mes y año del registro del programa
Número	Número consecutivo (1..n) de cada defecto del programa
Tipo	Clave (10...100) de tipo de defecto
Introducido	Fase (planeación, diseño, codificación) en la que se atribuye que se introdujo el defecto
Eliminado	Fase (revisión de diseño, revisión de diseño, compilación, pruebas) en la que se corrigió el defecto
Tiempo de corrección	Número de minutos que consumió la corrección del defecto
Defecto corregido	Número del defecto del que se derivó el que se está registrando-
Descripción	Causa que originó el defecto (no colocar el efecto sino lo que le dio origen)

#### Reporte de Prueba

Desarrollador \_\_\_\_\_ Fecha \_\_\_\_\_  
 Programa \_\_\_\_\_

Número de prueba	
Objetivo	
Descripción	
Condiciones de prueba	
Resultado esperado	
Resultado generado	

#### Propuesta de Mejoramiento de Proceso

Desarrollador \_\_\_\_\_ Fecha \_\_\_\_\_  
 Programa \_\_\_\_\_

1.	Descripción del Problema	
	Propuesta de mejoramiento	

## Anexo B

### Datos de desarrolladores

#### a) Desarrolladores cuyos programas fueron considerados para este estudio:

##### A) Comisión Federal de Electricidad

- 1) Alatorre Carranza Natalia
- 2) Barraza Arellano Irma Judith
- 3) De la Cruz Amezcua Hugo César
- 4) De la Cruz Preciado Omar Alejandro
- 5) Flores Gómez Carmen Adriana
- 6) Galindo Gauna Rogelio
- 7) García Ramos Manuel Arturo
- 8) Guerra Martínez Azor
- 9) Guzmán Martínez Alberto Adolfo
- 10) Hernández Hernández Paola Vanessa
- 11) Hernández Ramos Adriana Berenice
- 12) Larios Pérez José Luis
- 13) Partida Menchuca Laura

##### B) Grupo PAFTI-Alfa

- 1) Alegría Bobadilla Juan Manuel
- 2) Escamilla Rodríguez Jorge Alej.
- 3) Gutiérrez Ramírez Francisco José
- 4) Montesinos S. José Carlos
- 5) Morales López Diego Jesús
- 6) Plascencia Sánchez Juan Manuel
- 7) Reynoso Rojas Ricardo
- 8) Rivera Vega Bolivar Ulises
- 9) Vega Baray Fermín Martín
- 10) Viramontes Cortés Armando

##### C) Grupo PAFTI-Beta

- 1) Castro Gómez Teddy Raúl
- 2) Cordero Baltasar Diego
- 3) Davis Alcaraz Robert
- 4) Díaz Infante Montes Jorge
- 5) Domínguez Zárata Sahara
- 6) Duarte Lobo Miguel Angel
- 7) Jiménez Galicia Nayeli
- 8) Montero Silva Alberto
- 9) Martínez Sotelo Nicandro
- 10) Nuño Amézquita Jorge
- 11) Rocha Hernández José
- 12) Solís Orozco Miguel
- 13) Vega Ávalos César
- 14) Romero Rosas Alfonso
- 15) Vega Ávalos César

##### D) UNIVA

- 1) Becerril Ramírez Jorge Luis
- 2) Caro Guerra Rafael
- 3) De la Torre Gómez José Luis
- 4) Gutiérrez Hernández Ana Christina
- 5) Herrera Rábago Francisco Javier
- 6) Medina Estrada Christian
- 7) Navarro Rodríguez Jorge Arturo
- 8) Ramírez Carrasco Eduardo
- 9) Ramírez Castro Francisco
- 10) Reséndiz Martínez Jaime Ricardo
- 11) Sánchez Sánchez Francisco Javier
- 12) Santana Ruelas José Luis
- 13) Tamayo Emmanuel
- 14) Vargas Mora Dulce María

**Anexo B**  
*(continuación)*  
**Datos de desarrolladores**

**b) Desarrolladores cuyos programas fueron excluidos del experimento:**

<b>Desarrollador</b>	<b>Causa</b>
1) Alatorre Carranza Natalia	Cambio de entorno de desarrollo, inició con un lenguaje poco familiarizado para ella (Pascal) y terminó con C.
2) De la Cruz Amezcua Hugo César	Ausencia en dos sesiones
3) Larios Pérez José Luis	Ausencia en una sesión
4) Castro Gómez Teddy Raúl	Ausencia en dos sesiones
5) Nuño Amézquita Jorge	Inconsistencia en sus registros
6) Romero Rosas Alfonso	Inconsistencia en sus registros
7) Solís Orozco Miguel	Inconsistencia en sus registros
8) Vega Ávalos César	Inconsistencia en sus registros
9) De la Torre Gómez José Luis	Inconsistencia en sus registros
10) Navarro Rodríguez Jorge Arturo	Ausencia en dos sesiones
11) Ramírez Carrasco Eduardo	Ausencia en tres sesiones
12) Ramírez Castro Francisco	Al inicio del curso manifestó deficiencia en su habilidad para programación
13) Reséndiz Martínez Jaime Ricardo	Ausencia en tres sesiones
14) Santana Ruelas José Luis	Al inicio del curso manifestó deficiencia en su habilidad para programación
15) Vargas Mora Dulce María	Decidió codificar en un lenguaje de poca familiaridad para ella (JAVA)

## Anexo C

### Datos para la generación de los modelos de estimación

Datos reales de desarrolladores para generar los modelos (D: Desarrollador, P: Programa, RD: Revisión de Diseño; RC: Revisión de Código).

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
1	CFE1	1	49	0	27	0	50	0	36	26	139
		2	17	46	20	0	25	0	25	45	115
		3	98	0	40	0	62	20	56	49	227
		4	22	58	35	0	47	22	44	38	186
		5	153	0	25	0	70	15	20	15	145
		6	89	0	25	0	34	20	28	18	125
		7	22	84	21	0	17	15	19	10	82
2	CFE2	1	42	0	15	0	25	0	8	5	53
		2	62	40	14	0	20	0	6	30	70
		3	126	0	31	0	88	8	31	68	226
		4	17	77	15	1	14	5	10	11	56
		5	77	0	47	1	44	6	14	12	124
		6	96	0	35	1	37	7	5	49	134
		7	17	89	11	2	18	3	1	28	63
3	CFE3	1	74	0	9	0	39	0	10	47	105
		2	18	66	6	0	29	0	3	61	99
		3	142	0	10	0	88	5	60	30	193
		4	33	64	14	0	18	5	2	31	70
		5	125	0	15	2	64	2	4	37	124
		6	58	21	23	5	36	9	8	18	99
		7	35	63	12	4	30	3	16	35	100
4	CFE4	1	27	0	17	0	31	0	6	7	61
		2	45	19	8	0	33	0	2	21	64
		3	26	9	12	0	26	5	0	18	61
		4	5	25	5	1	14	2	2	23	47
		5	75	23	15	5	89	2	5	8	124
		6	32	17	17	1	26	2	3	43	92
		7	20	40	14	5	28	3	0	9	59
5	CFE5	1	53	0	14	0	40	0	7	25	86
		2	16	53	8	0	14	0	9	25	56
		3	89	0	10	0	50	7	30	12	109
		4	15	50	8	5	13	7	2	23	58
		5	84	0	31	14	50	10	40	40	185
		6	51	0	10	3	28	5	3	4	53
		7	3	47	5	2	10	3	2	6	28

## Anexo C (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
6	CFE6	1	41	0	7	0	25	0	5	26	63
		2	15	39	8	0	11	0	4	20	43
		3	84	0	22	0	47	3	10	13	95
		4	13	32	7	3	9	3	3	8	33
		5	129	0	22	4	36	4	17	5	88
		6	30	25	18	1	19	4	1	49	92
		7	8	48	7	5	8	2	1	26	49
7	CFE7	1	50	0	9	0	15	0	10	5	39
		2	6	50	10	0	10	0	5	5	30
		3	129	0	20	0	30	5	10	10	75
		4	10	73	15	0	10	3	10	10	48
		5	102	0	3	2	12	3	3	5	28
		6	81	0	20	2	12	5	22	15	76
		7	12	69	5	2	10	3	15	10	45
8	CFE8	1	110	0	11	0	35	0	8	10	64
		2	23	66	11	0	16	0	3	15	45
		3	139	0	5	0	63	9	3	23	103
		4	20	75	5	1	27	3	1	35	72
		5	117	0	4	3	39	8	4	13	71
		6	136	0	3	3	68	3	10	33	120
		7	21	115	5	1	21	4	1	40	72
9	CFE9	1	36	0	15	0	35	0	5	20	75
		2	42	0	10	0	30	0	35	35	110
		3	129	0	10	0	95	0	60	30	195
		4	34	10	10	0	40	0	15	30	95
		5	137	12	20	0	70	0	30	35	155
		6	34	5	10	0	70	0	15	45	140
		7	20	0	10	0	25	0	20	30	85
10	CFE10	1	59	0	12	0	43	0	25	30	110
		2	32	46	14	0	20	0	15	45	94
		3	60	74	25	0	60	5	30	20	140
		4	22	68	15	5	25	5	10	25	85
		5	111	20	30	5	50	5	25	20	135
		6	87	12	45	5	40	5	10	20	125
		7	28	68	25	5	20	5	15	25	95
11	A1	1	30	0	7	0	21	0	13	15	56
		2	29	16	10	0	16	0	2	20	48
		3	30	14	13	0	31	4	3	22	73
		4	17	28	8	2	26	6	2	21	65
		5	22	27	13	5	33	4	2	27	84
		6	30	17	16	4	28	4	1	18	71
		7	13	37	13	4	16	4	1	15	53

## Anexo C (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
12	A2	1	51	0	11	0	64	0	70	60	205
		2	79	0	30	0	75	0	10	50	165
		3	23	58	5	0	39	5	3	55	107
		4	47	42	15	6	54	4	4	44	127
		5	30	33	18	3	100	1	0	55	177
		6	47	59	10	2	54	4	2	18	90
		7	17	96	2	1	25	0	1	3	32
13	A3	1	72	0	6	0	37	0	1	17	61
		2	31	68	4	0	12	0	1	22	39
		3	32	68	5	0	21	10	1	14	51
		4	34	78	5	0	19	4	1	14	43
		5	35	38	3	0	25	5	1	18	52
		6	54	52	5	0	27	7	1	16	56
		7	50	80	5	0	25	5	1	16	52
14	A4	1	31	0	15	0	30	0	15	40	100
		2	26	25	10	0	8	0	3	19	40
		3	14	28	16	0	18	5	16	21	76
		4	11	33	5	3	19	5	3	9	44
		5	13	28	18	3	22	2	10	65	120
		6	25	8	10	3	10	2	1	24	50
		7	14	27	7	3	15	2	1	4	32
15	A5	1	33	0	40	0	40	0	5	20	105
		2	27	24	35	0	21	0	1	25	82
		3	19	27	30	0	25	4	1	15	75
		4	31	33	22	1	19	4	1	14	61
		5	12	25	13	1	14	1	1	8	38
		6	23	22	20	1	19	2	1	11	54
		7	10	42	12	1	8	1	1	12	35
16	A6	1	52	0	17	0	39	0	1	22	79
		2	36	34	23	0	20	0	7	18	68
		3	52	19	28	0	37	6	12	49	132
		4	19	56	13	3	13	11	4	15	59
		5	26	40	13	13	22	9	4	41	102
		6	23	11	14	4	15	3	2	33	71
		7	11	33	12	2	15	3	3	39	74
17	A7	1	29	0	11	0	20	0	9	41	81
		2	13	25	18	0	20	0	5	25	68
		3	20	22	30	0	23	5	24	9	91
		4	11	42	9	0	4	5	3	2	23
		5	15	42	10	4	34	3	34	37	122
		6	15	8	11	2	12	2	3	12	42
		7	23	33	17	1	15	1	10	7	51

## Anexo C (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
18	A8	1	38	0	7	0	40	0	11	6	64
		2	25	30	5	0	37	0	2	49	93
		3	21	26	8	0	42	0	30	48	128
		4	26	34	14	0	40	0	1	36	91
		5	17	41	10	0	44	0	4	35	93
		6	19	23	8	0	23	0	2	33	66
		7	13	34	8	0	32	0	1	21	62
19	A9	1	42	0	8	0	30	0	2	7	47
		2	42	28	11	0	22	0	1	21	55
		3	27	39	7	0	25	2	2	9	45
		4	11	57	4	0	12	1	1	16	34
		5	12	52	7	0	30	1	2	22	62
		6	12	40	6	0	20	1	1	11	39
		7	10	46	4	0	15	1	1	9	30
20	A10	1	30	0	17	0	19	0	19	13	68
		2	58	38	17	0	24	0	10	59	110
		3	54	26	15	0	23	3	8	24	73
		4	60	51	12	2	17	2	1	24	58
		5	45	18	57	4	16	3	35	30	145
		6	49	0	14	2	27	3	3	31	80
		7	81	65	7	2	30	2	4	14	59
21	B1	1	50	0	19	0	21	0	1	21	62
		2	58	15	40	0	44	0	11	43	138
		3	77	7	12	0	41	20	20	45	138
		4	50	24	16	2	28	3	6	28	83
		5	93	0	12	6	18	2	2	28	68
		6	143	0	12	8	52	5	30	15	122
		7	91	143	15	6	25	4	19	24	93
22	B2	1	33	0	50	0	15	0	20	10	95
		2	78	6	50	0	140	0	20	30	240
		3	51	29	50	0	30	10	20	20	130
		4	36	24	20	0	50	10	10	10	100
		5	17	21	5	0	20	5	5	30	65
		6	117	0	15	0	55	10	40	30	150
		7	42	111	20	0	35	5	10	30	100
23	B3	1	63	0	27	0	50	0	10	25	112
		2	38	30	17	0	39	0	9	31	96
		3	44	33	14	0	32	5	4	41	96
		4	75	30	7	0	34	6	5	52	104
		5	71	0	18	0	45	7	5	17	92
		6	79	0	23	0	54	10	11	30	128
		7	157	32	10	0	80	5	8	20	123

## Anexo C (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
24	B4	1	49	0	33	0	132	0	12	45	222
		2	32	27	33	0	150	0	24	30	237
		3	59	22	15	0	76	20	20	60	191
		4	48	41	42	15	55	14	10	55	191
		5	45	17	38	17	58	23	6	44	186
		6	81	32	12	4	68	18	7	46	155
		7	119	100	14	13	67	14	6	54	168
25	B5	1	27	0	9	0	28	0	15	23	75
		2	44	0	15	0	46	0	15	50	126
		3	39	5	6	0	34	8	13	50	111
		4	54	0	22	5	50	4	23	30	134
		5	28	0	11	4	15	5	20	30	85
		6	103	8	22	7	75	10	17	40	171
		7	97	30	17	10	50	8	16	30	131
26	B6	1	30	0	16	0	27	0	11	11	65
		2	28	15	44	0	22	0	6	18	90
		3	32	25	35	0	18	5	1	31	90
		4	37	21	32	1	39	8	8	35	123
		5	16	12	12	1	12	2	1	36	64
		6	49	18	51	3	30	6	54	15	159
		7	123	46	109	2	40	7	7	30	195
27	B7	1	49	0	23	0	60	0	32	30	145
		2	38	23	20	0	40	0	10	60	130
		3	26	22	23	0	53	5	3	30	114
		4	41	35	37	0	32	9	3	40	121
		5	20	15	16	0	55	4	9	30	114
		6	81	69	30	0	30	14	25	30	129
		7	84	150	7	0	100	4	3	10	24
28	B8	1	67	0	11	0	76	0	1	55	143
		2	47	9	11	0	46	0	1	48	106
		3	29	53	8	0	38	10	3	39	98
		4	29	0	10	0	22	3	4	10	49
		5	29	63	8	38	0	10	3	39	98
		6	182	50	8	0	48	30	35	20	141
		7	63	150	3	0	50	20	20	20	113
29	B9	1	43	0	15	0	29	0	2	11	57
		2	45	20	17	0	33	0	11	42	103
		3	34	39	17	0	24	9	0	41	91
		4	75	46	29	0	33	6	0	47	115
		5	44	5	7	1	52	4	15	18	97
		6	100	17	29	3	72	8	2	39	153
		7	121	111	26	2	69	8	7	32	144

### Anexo C (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
30	B10	1	49	0	11	0	50	0	4	20	85
		2	40	29	20	0	52	0	4	65	141
		3	41	26	4	0	34	19	3	37	97
		4	79	43	4	0	62	10	1	56	133
		5	51	0	10	0	25	10	1	24	70
		6	112	8	3	0	57	11	6	15	92
		7	84	119	6	0	51	11	1	9	78

## Anexo D

### Datos para la validación de los modelos de estimación

Datos reales de desarrolladores para validar los modelos (D: Desarrollador, P: Programa, RD: Revisión de Diseño; RC: Revisión de Código)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
31	U1	1	39	0	10	0	28	0	7	43	88
		2	49	0	10	0	24	0	1	36	71
		3	95	0	44	0	77	0	16	77	214
		4	86	0	29	0	23	0	5	48	105
		5	75	0	37	0	67	0	1	25	130
		6	58	0	7	0	9	0	1	15	32
		7	56	0	48	0	67	0	5	40	160
32	U2	1	37	0	15	0	35	0	12	40	102
		2	48	0	15	0	23	0	15	31	84
		3	104	0	33	0	42	0	2	37	114
		4	69	0	21	0	20	2	1	23	67
		5	107	0	47	0	56	1	10	27	141
		6	91	0	19	0	20	1	1	18	59
		7	53	0	30	0	55	1	1	35	122
33	U3	1	31	0	15	0	22	0	1	79	117
		2	20	23	30	0	32	0	1	70	133
		3	87	3	30	0	115	0	1	44	190
		4	19	41	19	0	20	0	1	40	80
		5	92	7	26	0	50	0	1	100	177
		6	27	8	14	0	30	0	1	27	72
		7	7	31	4	0	9	0	1	20	34
34	U4	1	30	0	10	0	50	0	5	10	75
		2	7	27	5	0	50	0	5	15	75
		3	109	0	20	0	180	0	10	30	240
		4	78	0	10	0	50	0	10	30	100
		5	155	0	20	0	120	0	10	30	180
		6	29	0	15	0	30	0	10	20	75
		7	40	0	15	0	45	0	10	20	90
35	U5	1	17	0	15	0	25	0	2	25	67
		2	4	16	15	0	20	0	3	10	48
		3	36	0	20	0	60	0	5	50	135
		4	2	20	15	0	40	10	5	20	90
		5	57	0	10	0	30	10	5	20	75
		6	50	0	10	0	20	5	5	25	65
		7	55	0	10	0	15	5	5	20	55

### Anexo D (continuación)

D	P	Código		Esfuerzo (en minutos)						Esfuerzo Total	
		NyC	Reusado	Diseño	RD	Codificación	RC	Compilación	Prueba		
36	U6	1	37	0	7	0	59	0	4	3	73
		2	21	32	2	0	4	0	31	6	43
		3	86	0	30	0	135	0	7	18	190
		4	55	0	18	0	19	0	7	10	54
		5	53	0	16	0	22	0	16	8	62
		6	60	0	17	0	26	0	8	9	60
		7	6	60	6	0	13	0	6	5	30
37	U7	1	31	0	7	0	59	0	4	3	73
		2	48	6	10	0	55	0	5	13	83
		3	53	1	10	0	45	0	7	10	72
		4	13	20	7	0	32	0	3	7	49
		5	109	0	25	0	69	0	3	9	106
		6	86	4	8	0	104	0	2	7	121
		7	6	88	8	0	38	0	1	9	56

## Anexo E

### Predicción del esfuerzo mediante el modelo de regresión ( $10 \leq \text{NyC} \leq 182$ )

*NyC: líneas de código nuevas y cambiadas*  
*Esfuerzo: Duración en minutos (diseño a pruebas)*

NyC	Esfuerzo	NyC	Esfuerzo	NyC	Esfuerzo	NyC	Esfuerzo
10	60.16	56	91.76	102	123.37	148	154.97
11	60.85	57	92.45	103	124.05	149	155.65
12	61.54	58	93.14	104	124.74	150	156.34
13	62.22	59	93.82	105	125.43	151	157.03
14	62.91	60	94.51	106	126.11	152	157.72
15	63.60	61	95.20	107	126.80	153	158.40
16	64.28	62	95.89	108	127.49	154	159.09
17	64.97	63	96.57	109	128.17	155	159.78
18	65.66	64	97.26	110	128.86	156	160.46
19	66.34	65	97.95	111	129.55	157	161.15
20	67.03	66	98.63	112	130.24	158	161.84
21	67.72	67	99.32	113	130.92	159	162.52
22	68.41	68	100.01	114	131.61	160	163.21
23	69.09	69	100.69	115	132.30	161	163.90
24	69.78	70	101.38	116	132.98	162	164.59
25	70.47	71	102.07	117	133.67	163	165.27
26	71.15	72	102.76	118	134.36	164	165.96
27	71.84	73	103.44	119	135.04	165	166.65
28	72.53	74	104.13	120	135.73	166	167.33
29	73.21	75	104.82	121	136.42	167	168.02
30	73.90	76	105.50	122	137.11	168	168.71
31	74.59	77	106.19	123	137.79	169	169.39
32	75.28	78	106.88	124	138.48	170	170.08
33	75.96	79	107.56	125	139.17	171	170.77
34	76.65	80	108.25	126	139.85	172	171.46
35	77.34	81	108.94	127	140.54	173	172.14
36	78.02	82	109.63	128	141.23	174	172.83
37	78.71	83	110.31	129	141.91	175	173.52
38	79.40	84	111.00	130	142.60	176	174.20
39	80.08	85	111.69	131	143.29	177	174.89
40	80.77	86	112.37	132	143.98	178	175.58
41	81.46	87	113.06	133	144.66	179	176.26
42	82.15	88	113.75	134	145.35	180	176.95
43	82.83	89	114.43	135	146.04	181	177.64
44	83.52	90	115.12	136	146.72	182	178.33
45	84.21	91	115.81	137	147.41		
46	84.89	92	116.50	138	148.10		
47	85.58	93	117.18	139	148.78		
48	86.27	94	117.87	140	149.47		
49	86.95	95	118.56	141	150.16		
50	87.64	96	119.24	142	150.85		
51	88.33	97	119.93	143	151.53		
52	89.02	98	120.62	144	152.22		
53	89.70	99	121.30	145	152.91		
54	90.39	100	121.99	146	153.59		
55	91.08	101	122.68	147	154.28		

## Anexo F

### Predicción del esfuerzo mediante el modelo difuso con funciones de membresía triangulares ( $10 \leq \text{NyC} \leq 182$ )

*NyC: líneas de código nuevas y cambiadas; Esfuerzo: Duración en minutos (diseño a pruebas)*

NyC	Esfuerzo	NyC	Esfuerzo	NyC	Esfuerzo
10	50.2	56	50.2	102	137
11	50.1	57	50.1	103	138
12	50.2	58	94.9	104	140
13	50.2	59	96.2	105	141
14	50.2	60	97.6	106	142
15	50.2	61	99	107	144
16	50.2	62	101	108	145
17	50.2	63	102	109	147
18	50.2	64	104	110	148
19	54.3	65	106	111	150
20	57.7	66	106	112	152
21	60.4	67	106	113 - 162	153
22	62.7	68	106	163 - 182	154
23	64.8	69	106		
24	66.5	70	106		
25	68	71	106		
26	69.4	72	106		
27	70.6	73	107		
28	71.7	74	108		
29	72.7	75	109		
30	73.6	76	110		
31	74.4	77	111		
32	75.2	78	112		
33	76	79	113		
34	76.7	80	114		
35	77.4	81	115		
36	78	82	116		
37	78.6	83	117		
38	79.2	84	118		
39	79.8	85	119		
40	80.4	86	120		
41	81	87	121		
42	81.7	88	122		
43	82.3	89	123		
44	82.9	90	124		
45	83.5	91	125		
46	84.2	92	126		
47	84.9	93	127		
48	85.6	94	128		
49	86.3	95	129		
50	87	96	130		
51	87.8	97	131		
52	88.7	98	132		
53	89.6	99	133		
54	90.5	100	135		
55	91.5	101	136		

## Anexo G

### Predicción del esfuerzo mediante el modelo difuso con funciones de membresía gaussianas ( $10 \leq \text{NyC} \leq 182$ )

*NyC: líneas de código nuevas y cambiadas; Esfuerzo: Duración en minutos (diseño a pruebas)*

NyC	Esfuerzo	NyC	Esfuerzo	NyC	Esfuerzo
10	68.6	56	88.8	102	150
11	68.4	57	89.6	103	151
12	68.2	58	90.5	104	152
13	68.1	59	91.3	105	153
14	68	60	92.2	106	154
15	67.9	61	93.1	107	155
16	67.9	62	94.1	108	155
17	67.8	63	95.1	109	156
18	67.8	64	96.2	110 - 112	157
19	67.8	65	97.4	113 - 115	158
20	67.9	66	98.5	116 - 121	159
21	68	67	99.8	122 - 174	160
22	68.1	68	101	175 - 182	159
23	68.2	69	102		
24	68.4	70	104		
25	68.6	71	105		
26	68.8	72	107		
27	69	73	108		
28	69.3	74	110		
29	69.6	75	111		
30	69.9	76	112		
31	70.2	77	114		
32	70.6	78	115		
33	71	79	117		
34	71.5	80	118		
35	72	81	119		
36	72.5	82	121		
37	73.1	83	122		
38	73.7	84	124		
39	74.4	85	125		
40	75.2	86	126		
41	75.9	87	128		
42	76.7	88	129		
43	77.5	89	131		
44	78.4	90	132		
45	79.3	91	134		
46	80.2	92	135		
47	81.1	93	137		
48	82.1	94	138		
49	83	95	140		
50	83.9	96	141		
51	84.8	97	143		
52	85.6	98	144		
53	86.4	99	146		
54	87.2	100	147		
55	88	101	149		

## Anexo H

### Predicción del esfuerzo mediante el modelo difuso con funciones de membresía trapezoidales ( $10 \leq \text{NyC} \leq 182$ )

*NyC: líneas de código nuevas y cambiadas; Esfuerzo: Duración en minutos (diseño a pruebas)*

NyC	Esfuerzo	NyC	Esfuerzo	NyC	Esfuerzo
10	49.5	56	92.3	102	138
11	49.5	57	93.5	103	139
12	49.4	58	94.8	104	141
13	49.4	59	96.2	105	142
14	49.3	60	97.7	106	144
15	49.3	61	99.3	107	145
16	49.2	62	101	108	147
17	49.2	63	103	109	148
18	49.1	64	105	110 - 182	150
19	49.1	65	106		
20	49	66	106		
21	49	67	106		
22	48.9	68	106		
23	52.4	69	106		
24	55.6	70	106		
25	58.3	71	106		
26	60.8	72	106		
27	63	73	108		
28	65	74	109		
29	66.8	75	110		
30	68.4	76	111		
31	69.9	77	112		
32	71.2	78	113		
33	72.4	79	114		
34	73.5	80	115		
35	74.5	81	116		
36	75.5	82	116		
37	76.3	83	117		
38	77.1	84	118		
39	77.8	85	119		
40	78.4	86	120		
41	79	87	121		
42	79.6	88	122		
43	80.1	89	123		
44	80.6	90	124		
45	81.5	91	125		
46	82.4	92	126		
47	83.3	93	127		
48	84.3	94	128		
49	85.2	95	129		
50	86.2	96	130		
51	87.2	97	132		
52	88.2	98	133		
53	89.3	99	134		
54	90.2	100	135		
55	91.2	101	137		

Podría llamar la atención el hecho de que en los sistemas difusos presenta un valor estable para el esfuerzo a partir  $NyC = 113$  hasta  $NyC = 182$ . A continuación se hace una comparación entre ambos modelos (comprobación de la adecuación de los modelos o verificación) para este rango de valores de  $NyC$  (ER: Esfuerzo Real; ERL: Esfuerzo por Regresión Lineal; ELDT: Esfuerzo por Lógica Difusa Triangular; ELDG: Esfuerzo por Lógica Difusa Gausiana; ELDP: Esfuerzo por Lógica Difusa Trapezoidal):

	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
1	129	88	141.91	0.61	0.38	153	0.74	0.42	160.00	0.82	0.45	150.00	0.70	0.41
2	136	120	146.72	0.22	0.18	153	0.28	0.22	160.00	0.33	0.25	150.00	0.25	0.20
3	143	122	151.53	0.24	0.19	153	0.25	0.2	160.00	0.31	0.24	150.00	0.23	0.19
4	157	123	161.15	0.31	0.24	153	0.24	0.2	160.00	0.30	0.23	150.00	0.22	0.18
5	125	124	139.17	0.12	0.11	153	0.23	0.19	160.00	0.29	0.23	150.00	0.21	0.17
6	182	141	178.33	0.26	0.21	154	0.09	0.08	159.00	0.13	0.11	150.00	0.06	0.06
7	121	144	136.42	0.05	0.06	153	0.06	0.06	159.00	0.10	0.09	150.00	0.04	0.04
8	153	145	158.4	0.09	0.08	153	0.06	0.05	160.00	0.10	0.09	150.00	0.03	0.03
9	117	150	133.67	0.11	0.12	153	0.02	0.02	159.00	0.06	0.06	150.00	0.00	0.00
10	137	155	147.41	0.05	0.05	153	0.01	0.01	160.00	0.03	0.03	150.00	0.03	0.03
11	119	168	135.04	0.20	0.24	153	0.09	0.1	159.00	0.05	0.06	150.00	0.11	0.12
12	123	195	137.79	0.29	0.42	153	0.22	0.27	160.00	0.18	0.22	150.00	0.23	0.30

MMRE	0.21		0.19		0.23		0.18
MMER	0.19		0.15		0.17		0.15

Se observa que la MMER resultó favorecedor para los tres modelos difusos con relación al modelo de regresión.

Tras un un análisis de heterocedasticidad, se observa su existencia. Previa comparación de medias se procede a dividir el grupo en dos submuestras C y D justas (capítulo 5):

### Submuestra C

	NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER
1	129	88	141.91	0.61	0.38	153	0.74	0.42	160.00	0.82	0.45	150.00	0.70	0.41
2	136	120	146.72	0.22	0.18	153	0.28	0.22	160.00	0.33	0.25	150.00	0.25	0.20
3	143	122	151.53	0.24	0.19	153	0.25	0.2	160.00	0.31	0.24	150.00	0.23	0.19
4	157	123	161.15	0.31	0.24	153	0.24	0.2	160.00	0.30	0.23	150.00	0.22	0.18
5	125	124	139.17	0.12	0.11	153	0.23	0.19	160.00	0.29	0.23	150.00	0.21	0.17
6	182	141	178.33	0.26	0.21	154	0.09	0.08	159.00	0.13	0.11	150.00	0.06	0.06
7	121	144	136.42	0.05	0.06	153	0.06	0.06	159.00	0.10	0.09	150.00	0.04	0.04
8	153	145	158.4	0.09	0.08	153	0.06	0.05	160.00	0.10	0.09	150.00	0.03	0.03

MMRE	0.24		0.24		0.30		0.22
MMER	0.18		0.18		0.21		0.16

## Submuestra D

NyC	ER	ERL	MRE	MER	ELDT	MRE	MER	ELDG	MRE	MER	ELDP	MRE	MER	
9	117	150	133.67	0.11	0.12	153	0.02	0.02	159.00	0.06	0.06	150.00	0.00	0.00
10	137	155	147.41	0.05	0.05	153	0.01	0.01	160.00	0.03	0.03	150.00	0.03	0.03
11	119	168	135.04	0.20	0.24	153	0.09	0.1	159.00	0.05	0.06	150.00	0.11	0.12
12	123	195	137.79	0.29	0.42	153	0.22	0.27	160.00	0.18	0.22	150.00	0.23	0.30

<b>MMRE</b>		<b>0.16</b>				<b>0.09</b>				<b>0.08</b>				<b>0.09</b>				<b>0.11</b>	
<b>MMER</b>				<b>0.21</b>				<b>0.10</b>				<b>0.09</b>							

Se observa que en la primer submuestra (C) la mejor MMER lo obtuvo el modelo difuso del tipo trapezoidal seguido del triangular y de la regresión, mientras que en la segunda submuestra (D) los modelos difusos superaron (por tener las MMRE y MMER más bajos) a la regresión lineal.