



Transacciones, Objetos de Entidad y Caminos:
métricas de software basadas en casos de uso, que
mejoran la estimación temprana de esfuerzo

Mg. Gabriela Robiolo

Director: **Dr. Ricardo Orosco**
Co-Director: **Dra. Claudia Pons**

**Tesis presentada para obtener el grado de Doctor en
Ciencias Informáticas**

Mayo 2009

Agradecimientos

A Ricardo y Claudia Pons, quienes confiaron en la viabilidad de la propuesta que les hice, me acompañaron y motivaron.

A todas las personas, tanto alumnos como profesionales, que me facilitaron datos de proyectos.

A Benjamín Del Sastre, Santiago Ceria, Cristina Badano, Juan Ale, Victor Basili, Tore Dyba, Sandro Morisca y Gabriela Arévalo, con cuyos oportunos comentarios y aportes fue posible construir este trabajo.

A familiares y amigos que me sostuvieron y apoyaron.

1. Introducción	5
1.1. Motivación	5
1.2. Nuestra Propuesta.....	8
1.2.1. <i>Objetivo de la Tesis</i>	8
1.2.2. <i>Aporte de la Tesis</i>	8
1.3. Publicaciones.....	9
1.4. Organización del trabajo	10
2. Estado del arte	11
2.1 Evolución de las métricas de tamaño y su convergencia con UML ..	11
2.2. Métricas basadas en casos de uso	14
2.3. Técnicas de estimación de esfuerzo	17
2.4 Conclusiones del capítulo	19
3. Marco teórico de las nuevas métricas	20
3.1. Conceptos subyacentes.....	20
3.1.1 <i>Casos de uso y Transacciones</i>	20
3.1.2. <i>Objetos de Entidad</i>	21
3.1.3. <i>Complejidad Ciclomática</i>	21
3.2. Lecciones aprendidas.....	22
3.2.1. <i>FP y sus limitaciones</i>	22
3.2.2. <i>Características comunes de métodos derivados de los FP</i>	25
3.2.3. <i>Puntos casos de uso y sus limitaciones</i>	27
3.2.4. <i>Unidades de medida de la precisión de la estimación de</i> <i>esfuerzo</i>	29
3.3. Conclusiones del capítulo	30
4. Descripción de las unidades de medida propuestas	32
4.1. Basadas en casos de uso.	32
4.2. Expresan funcionalidad, datos y complejidad	33
4.2.1. <i>Transacciones</i>	34
4.2.2. <i>Objetos de entidad</i>	36
4.2.3. <i>Caminos</i>	38
4.3. Ejemplo de cómputo de las métricas propuestas en un sistema de consulta de viajes.....	41
4.4. Ejemplo de cómputo de las métricas propuestas usando documentos complementarios	45
4.5. Consistencia con la teoría de medición.....	48
4.6. Conclusiones del capítulo	50
5. Demostración empírica de la disminución del error de la estimación de esfuerzo	52
5.1. Características de los proyectos	52
5.2. Productividad media histórica	54
5.2.1. <i>Primer escenario</i>	56
5.2.2. <i>Segundo Escenario</i>	58
5.2.3 <i>Análisis de los resultados</i>	60

5.3. Regresión lineal	62
5.3.1. Simple	62
5.3.2. Múltiple	63
5.3.3. Análisis de los resultados	65
5.4. Redes neuronales.....	67
5.4.1. Definición de los modelos	69
5.4.2. Análisis de los resultados	72
5.5. Descripción del contexto empírico	73
5.6. Conclusiones de la demostración empírica	75
6. Diseño e implementación de una herramienta para el	
seguimiento y control de productos de software	78
6.1 Informe de Visión.....	78
6.1.1 Posicionamiento	79
6.1.2 Descripción de Usuarios.....	79
6.1.3 Descripción global del producto	79
6.1.4 Otros requerimientos del producto	80
6.2. Casos de uso del Meta-Diseño	83
6.2.1 Actualizar información desde los Diagramas de secuencia ...	83
6.2.2 Actualizar información del Diagrama de Clases	85
6.2.3. Calcular métricas de Producto	86
6.2.4. Ingresar horas trabajadas.....	86
6.2.5 Modificar y Consultar horas trabajadas	87
6.2.6 Consultar avance de construcción de Producto	87
6.2.7. Consultar Métricas de Producto	89
6.2.8 Ingresar el Tamaño Pactado de un Producto y las	
identificaciones de los módulos	91
6.2.9 Editar Usuario	91
6.2.10. Editar Métricas de Producto	91
6.2.11. Editar Producto	92
6.2.12 Modificar estado de las clases.....	92
6.3. Descripción de una herramienta para el seguimiento y control de	
productos (SSCP)	93
6.4 Conclusiones del capítulo	97
7. Conclusiones.....	97
7.1 Contribuciones principales.....	98
7.2. Trabajos futuros	iError! Marcador no definido. 99
Referencias bibliográficas	102
Glosario de siglas y términos.....	106
Apéndices	111
Apéndice I. MRE para diferentes métodos de estimación.....	112
Apéndice II. Definición matemática de Tamaño y Complejidad	114
Apéndice III. Análisis de los supuestos de los modelos de la regresión	
lineal	116
Apéndice IV. Tabla de datos modelo multi-relacional.....	122

1. Introducción

1.1. Motivación

Las cuantiosas pérdidas que se registran, con más frecuencia de la deseada, en proyectos de software (Gibbs, 1994 y Charette, 2005) reclaman una constante mejora del control de los proyectos, de los procesos de desarrollo y de los métodos de estimación. Es especialmente interesante apuntar a los métodos que permiten a los líderes de proyectos realizar estimaciones durante la primera fase del ciclo de vida de un proyecto, eso es, una vez que está definido el alcance del producto. Esto se debe a que es en ese preciso momento en el que se toman las decisiones sobre las inversiones económicas a realizar.

Los errores en las estimaciones de esfuerzo de los productos de software suelen estar estrechamente relacionados con las pérdidas de los proyectos. Por lo tanto, disminuir el error en la estimación de esfuerzo implica evitar grandes pérdidas económicas. Si se trata de proyectos muy grandes, de 100 millones de dólares, por ejemplo, un pequeño error, digamos del 1%, se torna significativo.

El análisis de los errores en la estimación de esfuerzo de los productos de software reportados por diferentes autores es preocupante. Los resultados de las estimaciones están lejos de satisfacer las necesidades de la industria del software, puesto que los mismos suelen presentar un valor que no es aceptable para los inversores, esto es, una Magnitud del Error Relativo (Magnitude Relative Error, MRE) mayor a 0.25 (Fenton, 1997).

En el cálculo de la estimación temprana de esfuerzo, es posible identificar como variables críticas el *tamaño* y/o la *complejidad* de una aplicación. Su importancia ha sido reconocida por la definición de los estándares de métricas de tamaño funcional (ISO/IEC 14143:1998; ISO/IEC 19761:2003, ISO/IEC 20968:2002, ISO/IEC 20926:2003). Al mismo tiempo, el hecho de que los métodos de cálculo deban ser frecuentemente mejorados o adaptados a los nuevos contextos de desarrollo muestra que existe una necesidad subyacente que reclama un enfoque diferente para abordar la estimación de esfuerzo.

En la búsqueda de este nuevo enfoque, se puede trazar un paralelismo con la industria de la construcción, en la que se utiliza el metro cuadrado para medir los inmuebles. Los constructores pueden establecer un precio para su unidad de medida, y luego estimar el costo de la futura construcción, basándose en el tamaño de la misma. Esto conduce a la siguiente pregunta: *¿es posible que la industria del software adopte*

INTRODUCCION

una unidad de medida similar, que sea más simple e intuitiva que las que se están usando en la actualidad, y que permita obtener una estimación temprana de esfuerzo más eficiente?

Al igual que los autores (Fenton y Pfleeger, 1997), se piensa que es posible que la industria de software adopte una unidad de medida similar al metro cuadrado, siempre y cuando dicha unidad de medida sea fácil de comprender, simple y consistente con la teoría de representación de la medición.

Ahora bien, el hecho de que tal hipotética medida sea adoptada por la industria del software es un desafío a futuro, que está fuera del alcance de este trabajo. Sin embargo, en la búsqueda de tal medida, también hay que considerar que la adopción es un aspecto que se espera lograr, por lo tanto, las características enunciadas anteriormente, tales como facilidad de comprensión y simplicidad, son claves para lograr dicho objetivo. Otro aspecto clave para lograr tal adopción es modificar lo menos posible la forma de trabajo de las personas que van a utilizar esta unidad de medida. No hay que olvidar que las unidades de medida son usadas por los profesionales de la industria como un medio, no como un fin en sí mismo. Es probable que una unidad de medida que implique un esfuerzo importante de comprensión sea desechada: por lo tanto, *se impone la búsqueda de métricas que posibiliten una comprensión intuitiva.*

Para mejorar los resultados obtenidos con otras métricas en la medición productos de software, se plantea que en este enfoque innovador la nueva métrica sea:

- a. *Simple*, la misma debe ser creada a partir de un solo elemento, a diferencia de Puntos Función (Function Points, FP), que está constituida por cinco elementos (Entrada Externa, Salida Externa, Consulta Externa, Archivo Interno, Archivo Externo) y Puntos Casos de Uso (Use Case Points, UCP) que tiene dos elementos (Actor y Caso de uso). También la misma debe ser directa, lo que significa que es una métrica que mide un atributo¹ y que no depende de la métrica de ningún otro atributo (Olsina, 2004) y familiar al ambiente de UML.
- b. *Temprana*, debe ser posible usarla en un momento temprano del ciclo de vida de desarrollo de software, porque el inversor decide su inversión antes del inicio de la especificación del diseño. Es importante para él poder contar con una buena estimación una vez que se han definido los requerimientos.

¹ Un atributo es una propiedad física o abstracta mensurable, de una entidad.

INTRODUCCION

- c. *Precisa*, debe reducir la MRE. En el Apéndice I se muestran diferentes valores de la MRE reportados por distintos autores, quienes utilizaron diferentes métodos de estimación. El rango de variabilidad de MRE reportado es muy amplio y preocupante [0.0 – 6.94]. La Tabla muestra la dispersión actual, tanto en las técnicas de estimación (bayesian network, stepwise regresión, case-based reasoning, CART, COCOMO, COSEKMO, analogy based, regresión lineal simple y múltiple, productividad, estimación de expertos, vector prediction) como en las unidades de medida utilizadas (variables de proyectos, web sizes, UCP, FP, Use case rouge, de tamaño y complejidad) y la variedad de resultados alcanzados. Cabe destacar que el menor valor de MRE fue obtenido con los siguientes métodos de predicción:
- Case-based Reasoning and Stepwise Regresión, para dos conjuntos de aplicaciones Web (15 y 37 proyectos, respectivamente). El MRE reportado fue [0.02 – 0.32].
 - Vector Prediction para un conjunto de ocho proyectos de diferentes dominios, organizaciones y tipos de aplicación. El MRE reportado fue 0.20.

Esta dispersión disminuye la confianza en los resultados de las estimaciones. La industria del software necesita trabajar con un error más acotado y resultados repetibles.

- d. *Uniforme*, la nueva métrica debe resolver las limitaciones presentadas por los casos de uso, en lo referente a la variabilidad de tamaño. Como el tamaño de los casos de uso depende en parte del criterio que adopta la persona que los escribe, el caso de uso no es una unidad de medida apropiada. En parte, este problema puede ser controlado mediante la definición de un formato común o plantilla. Sin embargo, ésta no sería una solución convincente, especialmente si la aplicación fuera compleja, y si fuera necesario utilizar casos de usos que no hubiesen sido escritos siguiendo el mismo formato. Además, no es inherente a la definición del caso de uso la aplicación de una plantilla, lo que hace difícil su estandarización.
- e. *Intuitiva*, de tal forma que la métrica permita al estimador comprender instantáneamente, sin razonamiento, la característica o atributo a medir. Es posible que el dueño de un producto a desarrollar no tenga un perfil técnico, por lo tanto, es importante emplear una unidad de medida que él pueda comprender.
- f. *Factible de automatización*, como también es significativo para los desarrolladores, líderes e inversores, conocer el nivel de desarrollo durante la construcción de un producto de software, en lo que se refiere a la comprensión del progreso realizado, y dado el aumento del tamaño de las aplicaciones, dicha métrica debe permitir ser

INTRODUCCION

computada automáticamente, aspecto que es esencial para su adopción por parte de la industria.

También cabe destacar que las métricas a ser propuestas miden atributos internos de los productos de software, por lo tanto, los requerimientos no funcionales están fuera del alcance de este trabajo, ya que éstos especifican las propiedades de los sistemas, como ser restricciones de ambiente e implementación, desempeño, facilidad para el mantenimiento, extensibilidad, confiabilidad, etc. Como sólo es posible medir estos atributos externos cuando el producto está terminado, y el foco de interés de este trabajo está puesto en una etapa temprana del ciclo de vida del desarrollo de software, quedan claros los motivos de su exclusión del presente trabajo.

1.2. Nuestra Propuesta

Se propone un conjunto de tres unidades de medida de software que se identificarán con los nombres Transacciones (T), Objetos de Entidad (Entity Objects, EO) y Caminos (Paths, P). Cada una de ellas capturará un atributo interno del software: funcionalidad, datos y complejidad, respectivamente.

Es importante aclarar que aquí no se presenta un método de estimación, sino un conjunto de unidades de medida, que pueden ser usadas en un método de estimación temprana.

1.2.1. Objetivo de la Tesis

El objetivo de la Tesis es *definir tres nuevas unidades de medida de software, que estén basadas en casos de uso, y que reduzcan el error de la estimación temprana de esfuerzo.*

Para lograr tal objetivo, fue necesario trabajar en los siguientes sub-objetivos:

- a. Definir las unidades de medida para la base de cálculo seleccionada: casos de uso.
- b. Definir la forma de cálculo de las nuevas unidades de medida.
- c. Cumplir con los requisitos que se impusieron en 1.1.
- d. Probar, por medio de casos de estudio que incluyan varios proyectos, que es posible reducir el error de la estimación temprana de esfuerzo para las técnicas: Productividad² Media Histórica (Historical Mean Productivity, HMP), Regresión Simple y Múltiple, y Redes Neuronales.

1.2.2. Aporte de la Tesis

² Cociente entre el Tamaño y el Esfuerzo o cociente entre la Complejidad y el Esfuerzo.

INTRODUCCION

El aporte de la tesis consiste en *contribuir a la madurez de las métricas de estimación temprana*.

Específicamente este aporte se concreta en:

- a. Brindar a la Industria del Software un conjunto de unidades de medida de tamaño³ y complejidad⁴, que reduce el error en la estimación temprana de esfuerzo.
- b. Aportar unidades de medida de software de tamaño y complejidad, base para el cálculo de productividad.
- c. Introducir unidades de medida que facilitan el control del avance de la construcción de un producto de software, con un grado de traceabilidad que va desde la definición de requerimientos al diseño.
- d. Aportar las bases para la definición de una herramienta de cálculo automatizado de tamaño y complejidad de productos de software.

1.3. Publicaciones

Fue posible publicar artículos con referato en varios congresos afines a la Ingeniería de software en Argentina y en el exterior. Además, cabe destacar que el "Doctoral Symposium on Empirical Software Engineering, IDoESE" se desarrolló dentro del marco del "Internacional Symposium on Empirical Software Engineering and Measurement", que nuclea a las personas que están trabajando en el tema de métricas. Fue una oportunidad para validar los contenidos de este trabajo con personalidades tales como: Victor Basili, Dieter Rombach, Tore Dyba, Sandro Morasca. Los trabajos publicados son:

- a. *A Preliminary Study on the Early Estimation of Effort based on Use Cases*

Autores: Robiolo, G. y Orosco, R.

Publicado en: ASSE 2006 Argentine Symposium on Software Engineering, Mendoza, Argentina, September 6 - 7, 2006

- b. *An alternative method employing uses cases for early effort estimation*

Autores: Robiolo, G, Orosco, R.

³ Cantidad de elementos que tiene un entregable. El aumento o disminución de tamaño está unido a la mayor o menor cantidad de elementos.

⁴ Cantidad de relaciones que existen entre los elementos de un entregable⁴. El aumento o disminución de complejidad está unido a la mayor o menor cantidad de relaciones entre los elementos.

INTRODUCCION

Publicado en: Software Engineering Workshop SEW 2007, Baltimore, MD, March 2007.

c. *Employing use cases to early estimate effort with simpler metrics.*

Autores: Robiolo, Gabriela; Orosco, Ricardo.

Publicado en: Innovations in Systems and Software Engineering, Volume 4, Number 1, April 2008 , pp. 31-43(13), Springer 2008

d. *A simple approach that improves early effort estimation based on use cases*

Autores: Robiolo, Gabriela.

Publicado en: Proceedings of the Third International Doctoral Symposium on Empirical Software Engineering, IDoESE, 2008, Kaiserslautern.

1.4. Organización del trabajo

A continuación se describen los contenidos de los próximos capítulos.

El Capítulo 2 presenta el estado del arte, concentrándose en tres cuestiones: la evolución de las métricas de tamaño y su convergencia con UML, la existencia de métricas basadas en casos de uso y las técnicas de estimación de esfuerzo más usadas.

El Capítulo 3 describe los conceptos subyacentes en la definición de las nuevas métricas (*casos de uso, transacciones, objetos de entidad y complejidad ciclométrica*) y las lecciones aprendidas con el uso de dos métricas referentes para este trabajo: FP y UCP.

El Capítulo 4 describe las tres nuevas unidades de medidas que se utilizarán en la medición de tamaño y complejidad de las aplicaciones. Además, presenta dos ejemplos de cómputo de las métricas propuestas en dos aplicaciones de software tomadas de la industria y analiza la consistencia de las métricas propuestas con la teoría de la medición.

El Capítulo 5 demuestra, a través de un caso de estudio de 13 proyectos reales de negocios pequeños, que el error de la estimación temprana de esfuerzo fue reducido usando las técnicas HMP, Regresión Lineal Simple y Múltiple y Redes Neuronales.

El Capítulo 6 describe el diseño y la implementación de una herramienta para el seguimiento y control de productos de software.

Finalmente, el Capítulo 7 presenta las conclusiones, destacando las contribuciones y los trabajos futuros.

2. Estado del arte

La primera pregunta que surge antes de analizar el estado del arte es el motivo por el cual se piensa en nuevas métricas. Es necesario justificar este planteo dilucidando las limitaciones de las métricas actuales, las necesidades de los contextos tecnológicos vigentes y las lecciones aprendidas de las métricas anteriores.

Por lo tanto, un resumen del estado del arte, que define la base para el planteo de la nueva propuesta de métricas, ayudará a encontrar una respuesta a las siguientes preguntas:

- a. ¿Cuál es la evolución que han sufrido las métricas de tamaño en los últimos años? ¿Cómo ha impactado la UML en la definición de métricas de tamaño y complejidad?
- b. ¿Existen métricas basadas en casos de uso?
- c. ¿Cuáles son las técnicas de estimación de esfuerzo más usadas?

Antes de responder estas preguntas, es importante destacar que lo primero que se advierte es que las métricas de tamaño y/o complejidad vigentes no han surgido en los contextos tecnológicos actuales, sino que son anteriores. Por ejemplo, el peso a asignar por complejidad Baja, Media, Alta, en FP se mide en términos de datos, puesto que esta métrica surgió en contextos donde las aplicaciones son frecuentemente orientadas a bases de datos. A pesar de que en la actualidad las bases de datos siguen vigentes, ciertas tecnologías como la orientación a objetos, las aplicaciones Web o la integración de dispositivos a las aplicaciones han variado las arquitecturas de las aplicaciones, modificando el impacto que los datos tienen en la complejidad de los sistemas.

Otros ejemplos evidentes son las líneas de código y los puntos objetos. En el primer caso, si bien ellas se siguen utilizando por ser una medida de tamaño objetiva, y medible en forma automatizada, no cabe duda de que es necesario mejorarla. Esto se debe a que las medidas de tamaño suelen ser empleadas como base para la estimación de costo o medición de productividad, y es evidente que a mayor cantidad de líneas de código, no necesariamente hay mayor productividad. Es más, los programadores más experimentados, son capaces de resumir en una línea de código varias escritas por programadores novatos.

En el segundo caso, se llaman puntos objetos y no se tratan de "objetos" propios de la orientación a objetos, sino que los puntos objetos son calculados usando los siguientes elementos: pantallas, salidas impresas y componentes 3GL.

2.1 Evolución de las métricas de tamaño y su convergencia con UML

Jorgensen & Shepperd (Jorgensen & Shepperd, 2007) destacan que casi un tercio de los artículos de investigación escritos sobre estimación de costo en los años 90 tratan sobre FP, registrándose en este período el pico más alto de interés en esta métrica. Además, hoy día los usuarios no están completamente satisfechos con los resultados obtenidos con los métodos de estimación; los errores en la estimación de esfuerzo son demasiado grandes (cfr. Apéndice I), lo que lleva a reflexionar sobre si las métricas actuales son adecuadas a las necesidades de los líderes de proyectos, inversores o dueños de las aplicaciones.

A pesar que FP es una métrica probada, la investigación en este ámbito muestra que ha evolucionado (Levesque et al., 2008) tendiendo a simplificarse (cfr. Tabla 4, filas 2, 3 y 4) y a superar las dificultades que se hallaban en los estándares ISO/IEC 20968:2002 e ISO/IEC 20926:2003. Con esta finalidad surge ISO/IEC 19761:2003 COSMIC-FFP.

Sin embargo, aunque CÓSMIC-FFP no presente las dificultades antedichas, aún presenta la desventaja de que no sido planteada para calcular el tamaño funcional de productos de software que son caracterizados por algoritmos matemáticos complejos u otras reglas especializadas o complejas o variables de procesos continuos, como ser los sonidos de audio o imágenes de vídeo, encontradas, por ejemplo, en el software de juegos por computadora. Esta limitación se pone en evidencia en la propuesta de Levesque (Levesque et al., 2008), puesto que propone añadir el concepto de complejidad al método CÓSMICO-FFP, para modelos UML, midiendo la complejidad en un diagrama de secuencia.

También otros autores encuentran necesario ajustar los FP al contexto actual de desarrollo. Por ejemplo, algunos introdujeron el concepto de "web objects" (Ruhe et al., 2003). Ellos investigaron sobre la aceptabilidad de una nueva métrica de tamaño propuesta para proyectos desarrollados en la web, creando una extensión de FP, que llaman "objetos web". En su artículo concluyen que el tamaño de un sistema, medido en FP o en objetos web, no contribuye a hacer una estimación más exacta que la que realiza un experto. Así mismo, verifican que los objetos web contribuyen a obtener mejores estimaciones de esfuerzo en aplicaciones web, que las obtenidas usando FP.

Al mismo tiempo, hay varios autores que han adaptado las métricas existentes, de forma tal que puedan contar los elementos de las métricas en modelos de UML. Otros han convergido en la idea de *usar la UML como forma para resolver las dificultades para el conteo automatizado de las métricas, y estar en condiciones de diseñar una herramienta automatizada.*

Actualmente, los métodos que han sido adaptados a UML obligan al uso de elementos de UML en una determinada manera, que limita el uso de UML, ya que impiden el uso de la información histórica. Esto hace que la aceptación extendida de tales métodos sea bastante improbable. Los ejemplos de tales adaptaciones son detallados a continuación:

- a. Fetcke (Fetcke et al., 1999) define un excelente mapeo del método de Jacobson "Object-Oriented Software Engineering (OOSE)" y el método de cálculo de los FP detallado en el manual desarrollado por International Function Point User Group. Su artículo muestra una forma práctica y fácil de calcular FP, basada en los modelos del método de Jacobson. Además, Harput (Harput et al. 2005) define reglas que especifican la transformación semi-automática de un modelo de Requerimientos orientado a objetos a un modelo de Análisis de FP, para aplicar FP a una especificación de requerimientos orientada a objetos. Ahora bien, si bien es posible calcular los FP en el contexto de UML, al ser FP, siguen manteniendo sus limitaciones.
- b. Uemura (Uemura et al., 1999) desarrolla una herramienta para la medición automática de FP, basada en los modelos generados por el Rational Rose. Esta es una buena solución para obtener un cálculo cuantitativo de la funcionalidad de un sistema; pero, al mismo tiempo, el cálculo de la complejidad de la Función Transaccional (Transactional Function) está resuelto de forma tal que no siempre es posible reproducirla, ya que miden la complejidad contando los argumentos presentes en los mensajes de los diagramas de secuencia. Este es un aspecto que se considera débil debido a que los programadores utilizan diferentes estrategias para definir los argumentos, por ejemplo, los pueden encapsular en un objeto, de esa forma distorsionando el cálculo de la complejidad.
- c. Ceddia y Dick (Ceddia y Dick, 2004) automatizaron el conteo de FP usando el Diagrama de Entidad y Relación, y el Diagrama de Caso de Uso, con la siguiente limitación: cada caso de uso debe representar una sola transacción, tanto una entrada como una salida, y cada transacción es modelada como un solo caso de uso. *También se impuso esta última limitación, a la herramienta que se desarrolló para el seguimiento y control de productos (SSCP, cfr.6.3)*
- d. Issa (Issa et al., 2005) adapta Puntos Objeto, de modo tal que puedan ser contados en forma automática en un meta-modelo de casos de uso, con la intención de que se puedan eliminar la mayor parte de las carencias del modelo de caso de uso. El meta-modelo impone una forma de unificar los diferentes modos de escribir los casos de uso que difícilmente sea aceptada.

- e. Levesque (Levesque et al., 2008) declara que la razón principal por la cual ninguna de las normas de FP ISO Standard han sido totalmente automatizadas se debe a que cada método tiene su propio vocabulario y su propio modo de modelar el software. Consideraron para la automatización del método COSMIC-FFP y en la adición de una medida objetiva de complejidad a la medida estándar de COSMIC-FFP, el uso de modelos del UML. Divulgaron que es posible estimar el número de FP según COSMIC contando el número de mensajes UML intercambiados dentro de un diagrama de secuencia UML. Por otro lado, también, Kim (Kim et al., 2006) introduce puntos UML, que es la combinación de puntos de caso de uso y puntos clase, y propone una herramienta automatizada de análisis de métricas de software. Lamentablemente, ambos autores recién podrán realizar dicho cálculo en una etapa posterior al modelo de casos de uso.
- f. Lavazza (Lavazza et al., 2008) construye FP orientados a modelos UML que no tienen que incluir más información que la requerida en forma general en el proceso de desarrollo. Sin embargo, para asegurar que la información requerida por el FPA es incluida correctamente, los autores imponen un cierto modo de trabajar que no es habitual entre los desarrolladores.

Se observa una tendencia a integrar los FP y sus sucesivas adaptaciones con UML. Ante esta realidad, surge la siguiente pregunta: *¿por qué no pensar en la posibilidad de definir métricas que sean inherentes a los modelos de UML, más que adaptar las existentes?*

2.2. Métricas basadas en casos de uso

Un concepto inherente al UML es el caso de uso. Karner (Karner, 1993) define puntos caso de uso que son similares a FP, como una forma de medir tamaño en aplicaciones basadas en casos de uso. Diev (Diev, 2006) define una heurística para la aplicación de UCP y reglas para los modelos de UML, para hacer posible el conteo de FP. Lamentablemente, estos trabajos tienen las mismas limitaciones que Kitchenham (Kitchenham, 1997) señala con respecto a FP.

Briand (Briand et al., 1996) piensan que la investigación sobre la medición también debería abarcar entregables⁵ desarrollados en las tempranas fases del proceso de software, dado que ellos son los más importantes debido a que el resto del desarrollo depende de estos entregables. Además, la mayoría de las veces, los conceptos tamaño y complejidad, que, son relevantes en el código, son también para otros entregables. *Esta opinión confirma la elección de casos de uso como base para la medición.*

⁵ Traducción del término "artifacts".

Anda (Anda et al., 2005) presenta una experiencia que usa UCP para la estimación de esfuerzo de un sistema específico. Cuatro diferentes empresas, basándose en una misma definición de requerimientos, desarrollaron cuatro aplicaciones similares. Cada empresa usó un proceso de desarrollo diferente, con el propósito de comparar distintos valores de esfuerzo. El artículo muestra una diferencia significativa entre el esfuerzo estimado calculado en base a los puntos casos de uso y el esfuerzo real de los proyectos, lo que lleva a la conclusión de que el método no está lo suficientemente maduro para realizar buenas predicciones y/o de que los procesos afectan las estimaciones de costos más de lo que habitualmente se espera.

Dos aspectos importantes con respecto a la utilización de los casos de uso como una medida de tamaño son la variación de dimensión y el nivel de detalle, ya que diferentes desarrolladores escriben casos de uso con modalidades distintas. Además, Anda (Anda et al., 2001) señala que el uso de UCP es afectado por aspectos diferentes de la estructura del modelo de caso de uso, por ejemplo: el uso de generalizaciones entre actores, la utilización de casos de uso incluidos y extendidos, el nivel de detalle en las descripciones de caso de uso. Para mejorar la estimación de modelos basados en caso de uso, Anda (Anda et al., 2002) recomienda que un nivel balanceado de detalle sea usado en la escritura de los casos de uso, cuando el modelo de caso de uso deba utilizarse como base para la estimación. Aconseja usar los buenos ejemplos de casos de uso o definir directrices específicas para el proceso de modelado de casos de uso. Estas sugerencias conducen a una reducción del error en la estimación de esfuerzo, pero no eliminan el problema, dado que no hay ninguna forma objetiva y práctica para fijar, para todos los casos de uso -históricos o nuevos-, un nivel balanceado de detalle.

Vinsen et al. (2004) advierten que los FP han sido aplicados satisfactoriamente a sistemas de caso de uso en pequeña escala, pero que puede haber dificultades con sistemas en gran escala, dado que la funcionalidad detallada del sistema no puede ser especificada en el caso de uso, sino que es necesario usar documentos complementarios que incluyen reglas, condiciones, exclusiones, etc. Además, Issa (Issa, et al. 2005) resalta que, cuanto más detallado sea un modelo de caso de uso, más exacta será la estimación de esfuerzo, por lo cual, *se torna necesario encontrar una unidad de medida que pueda medir el tamaño del caso de uso en documentos complementarios.*

Mohagheghi (Mohagheghi et al., 2005) introduce algunas variaciones en la utilización de los casos de uso de un desarrollo industrial de software a gran escala, adaptando el método UCP. Él considera que cada transacción y cada flujo alternativo de un caso de uso, es a su vez, un nuevo caso de uso.

También, Braz y Vergilio (Braz & Virgilio, 2006) adaptan los UCP, para ser usados con casos de uso más detallados, introduciendo los USP (Use Case Size Points), como una métrica de tamaño que considera la estructura interna de un caso de uso. El USP mide la funcionalidad en las estructuras y las secciones de un caso de uso, contando el número y el peso de escenarios, actores, condiciones previas y posteriores. También introducen los FUSP (Fuzzy Use Case Size Points), otra métrica que considera conceptos de la Teoría "Fuzzy Set" para crear una clasificación gradual que trabaje mejor con la incertidumbre. Sin embargo, *el hecho de que los UCP deban ser adaptados para ser usados en circunstancias diferentes, demuestra que éste no es un método maduro para medir tamaño funcional.*

Además, vale la pena destacar que otros autores, como por ejemplo, Bernárdez (Bernárdez et al. 2004) aplican el concepto complejidad ciclomática a un caso de uso, pero con distinta intención. El objetivo de Bernárdez es medir la complejidad ciclomática para validar la definición de caso de uso, mientras que en este trabajo se busca una unidad de medida que mejore la estimación de esfuerzo.

Por otra parte, cabe mencionar que en un artículo proporcionado por el Rational Unified Process, Smith (Smith, 1999) propone un modo de estimar esfuerzo basado en líneas de código. Además, el autor también incluye una jerarquía estructural, que permite la descomposición de un sistema, la definición de la cantidad de casos de uso en cada nivel jerárquico, y la definición de un número de escenarios para cada caso de uso. Sin embargo, no queda clara la razón por la cual así define a la jerarquía estructural y la cantidad de casos de uso. Aunque a primera vista, éste parezca ser un buen ejemplo de estimación de esfuerzo basado en casos de uso, no se han tenido en cuenta para este estudio ni el procedimiento ni las métricas líneas de código. Esto se debe a que el procedimiento que Smith desarrolla no está claramente justificado, y a que las líneas de código no han demostrado ser útiles para medir la productividad, dado que éstas no pueden ser contadas en una etapa temprana del ciclo de vida de desarrollo de software.

Con respecto a los casos de uso, se puede concluir que si bien es un concepto inherente a UML, y es posible usarlo en una etapa temprana del ciclo de vida de desarrollo de software, *los casos de uso como unidad de medida no mejoran el error en la estimación de costo. Además, las adaptaciones hechas para que los FP puedan ser contados en casos de uso comparten las limitaciones de FP. Además, los UCP comparten con los casos de uso las limitaciones para manejar el nivel de detalle propio de las aplicaciones grandes y complejas. Así mismo, se resalta la necesidad de encontrar una métrica que también pueda ser aplicada en documentos complementarios.*

2.3. Técnicas de estimación de esfuerzo

En el artículo "A Systematic Review of Software Development Cost Estimation Studies", (Jorgensen & Shepperd, 2007), analizaron 304 artículos divulgados en 76 publicaciones. La búsqueda de los artículos finalizó en el año 2004. Ellos resaltan que predominan las técnicas basadas en la regresión, un 49% del total de los artículos, y en FP, con un 22%, aunque cabe destacar que ambas demuestran un interés decreciente. Por otro lado, se registra la utilización de técnicas de clasificación y árboles de regresión, 5%, y redes bayesianas, 2%, con un interés creciente; y de redes neuronales, con un 7%, y un interés constante.

En la Tabla 1 (para más datos cfr. Apéndice I), cuya ventana temporal abarca desde 1992 al 2007, se puede observar la tendencia existente entre las variables usadas y las técnicas de estimación aplicadas. Se sintetizan de la siguiente forma:

- a. *Los FP generalmente se usan con modelos de regresión lineal*
- b. *Los UCP generalmente se usan con productividad*
- c. Las Variables de Proyectos se usan con diferentes métodos, como por ejemplo, Bayesian Network, Stepwise Regresión, Case-based Reasoning, CART, árboles de Regresión

En una extensa revisión de estudios relativos a estimación realizada por expertos, Jorgensen (Jorgensen, 2004) afirma que la estimación hecha por expertos es la más frecuentemente aplicada, y que existen situaciones en las que se puede esperar que este tipo de estimación sea más exacta que los modelos de estimación formal. Al mismo tiempo, afirma que utilizar la combinación de los modelos de estimación y los juicios de expertos es mejor que la pura estimación de expertos. *En este último punto es donde se desea realizar una contribución, aportando nuevas métricas que puedan ser usadas en los modelos formales de estimación.*

Tabla 1. Relación entre variable y método a utilizar

Referencia	Métricas	Método de estimación
(Ferens & Gurner, 1992)	FP	Regresión lineal
(Matson, et al.,1994)	FP	Regresión lineal simple y múltiple
(Hasting, T.E. and Sajeev, A.S.M. 2001)	Operators and operands	Vector Prediction Model
(Anda, Angelvik, et al. 2002)	UCP, Estimación de expertos	Productividad
(Mendes et al. 2002)	8 de tamaño y complejidad	Stepwise regression, Case based reasoning
(Ruhe et al., 2003)	FP, Web objects	Regresión lineal, estimación de expertos
(Mendes, Watson et al. 2003)	8 variables de complejidad y tamaño	Cased Based Reasoning, Stepwise Regression, Regression Trees
(Mendes, et al., 2003)	Web size measures	Early Web size measures, Stepwise Regression y Case-based Reasoning
(Moløkken-Østvold, et al., 2004)	Datos de proyectos	84% Estimación de expertos, 16% Combinación de estimación de expertos y modelos de estimación
(Anda, et al., 2005)	UCP, Estimación de expertos	Productividad
(Braz, & Vergilio, 2006)	UCP, FP, USP, FUSP	Productividad
(Issa, et al., 2006)	Use Case rough	Productividad
(Chen, et al. 2006.)	De requerimientos, UML, SLOC	COCOMO, COSECKMO
(Ohsugi, et al. 2007)	14 variables, incluye FP	Analogy-based
(Di Martino, et al. 2007)	25 variables	Case-based Reasoning, Stepwise Regression
(Mendes, 2007)	Variables Base de datos Tukutuku	Bayesian Network, Stepwise Regression, Case-based Reasoning , CART

2.4. Conclusiones del capítulo

A modo de resumen, se destacan las siguientes ideas:

- a. Los estándares de tamaño funcional han sufrido sucesivos ajustes a los ambientes de desarrollo actuales.
- b. Los márgenes de error reportados por los trabajos de investigación no satisfacen las necesidades de los líderes de proyectos, inversores o dueños de las aplicaciones.
- c. Se observa una tendencia a integrar los FP y sus sucesivas adaptaciones con UML, en algunos casos con la intención de plantear una herramienta automatizada.
- d. Si bien los casos de uso son un elemento inherente a UML, no se ha podido solucionar el problema del nivel de detalle, puesto que la estrategia usada por la mayoría de las métricas basadas en casos de uso ha sido imponer la utilización de elementos de UML de una determinada manera, que limita dicho uso, lo que resulta en una improbable aceptación de esas métricas.
- e. En cuanto a los UCP, que tienen la característica de ser inherentes a UML, y la ventaja de no imponer una forma determinada de escribir el caso de uso, es posible usarlos en una etapa temprana del ciclo de vida del desarrollo de software. Así mismo, los UCP comparten algunas de las limitaciones de FP, y otras que se desprenden de la dificultad para manejar el nivel de detalle propio de las aplicaciones grandes y complejas.
- f. Además, sería necesario encontrar una unidad de medida que pueda medir el tamaño del caso de uso en documentos complementarios.

3. Marco teórico de las nuevas métricas

Este capítulo describe los conceptos subyacentes en la definición de las nuevas métricas y las lecciones aprendidas de las siguientes métricas: FP –estándar de medición de tamaño funcional-, UCP y las unidades de medida de la precisión de la estimación de esfuerzo.

El objetivo es clarificar las bases sobre las cuales se va a construir este trabajo.

3.1. Conceptos subyacentes

Para la definición de las métricas se seleccionaron conceptos que son propios o inherentes a ambientes de desarrollo actuales -*casos de uso, transacciones y objetos de entidad* (Jacobson et al., 1992)- y una métrica de complejidad ampliamente aceptada -*complejidad ciclométrica* (McCabe, 1976)-.

3.1.1 Casos de uso y Transacciones

Jacobson define el *caso de uso* como una forma especial de usar el sistema, ejecutando alguna parte de su funcionalidad; cada caso de uso constituye una secuencia completa de eventos disparados por un actor y especifica la interacción que existe entre el actor y el sistema (Jacobson et al., 1992). Por lo tanto, un caso de uso es una secuencia especial de transacciones realizadas por un actor y un sistema, en forma de diálogo. Se define el caso de uso desde la perspectiva del actor.

Desde el punto de vista del sistema, se puede decir que un caso de uso es un flujo completo del sistema: es decir, una *transacción*. Un caso de uso puede tener una o más transacciones, dependiendo del criterio usado por la persona que escribe el caso de uso. Esto ha sido tenido en cuenta por Karner (Karner, 1993) en su definición del método UCP, que define el tamaño de la aplicación en función del número de las transacciones que existen en cada caso de uso.

En la actualidad, se define a los casos de uso con un grado de formalidad muy bajo, debido a que siempre y cuando cumpla con la definición anteriormente dada, un caso de uso puede ser escrito de diferentes formas. Esta flexibilidad presenta la ventaja de que el mismo sea fácil de adoptar, pero también la desventaja de que, si se utiliza al caso de uso como unidad de medida de tamaño para estimar esfuerzo, el grado de exactitud de dicha estimación no será confiable debido a que el tamaño de los casos de uso puede ser muy diferente. Varios trabajos analizan ésta y otras limitaciones, pero ninguno de los autores plantea una solución convincente (cfr. 3.2).

3.1.2. Objetos de Entidad

Jacobson (Jacobson et al., 1992) define los *objetos de entidad* como aquellos objetos que manejan la información persistente del modelo. La mayoría de estos objetos se identifican en los casos de uso y, habitualmente, son los objetos evidentes que participan en el modelo de objetos del dominio del problema. Las entidades corresponden a conceptos de la vida real, las cuales existen fuera del sistema. Es preciso considerar solamente las entidades que son necesarias para realizar el caso de uso. La Figura 1 muestra un ejemplo donde se identifican los objetos de entidad de la descripción del caso de uso "Dar de alta a un producto en un catálogo". Los objetos de entidad son las palabras resaltadas en negrita.

1. El sistema muestra una pantalla con la lista de **categorías** y **subcategorías**.
2. El usuario debe *seleccionar* la **categoría** a la que pertenece el nuevo **producto**. Se muestra en la parte inferior de la pantalla la lista de las propiedades de las **categorías** y **subcategorías** seleccionadas. Si las **categorías** y **subcategorías** son nuevas, ENTONCES se desplegará una descripción detallada de ellas.
3. Presionando la tecla "enter", el usuario va a acceder a una nueva pantalla, en la cual debe completar la definición de las propiedades del **producto**.
4. El caso de uso finaliza cuando el usuario carga el **producto**, *graba* los cambios y deja el sistema o cancela la operación.

Fig. 1. Descripción de un caso de uso "Dar de alta a un producto en un catálogo"

3.1.3. Complejidad Ciclomática

La métrica "*Complejidad ciclomática*" fue introducida por Thomas McCabe en 1976 (McCabe, 1976). Esta métrica da un número que puede ser usado para definir la complejidad de los programas. Mide la cantidad de decisiones lógicas en el código fuente y tiene la característica de ser independiente del lenguaje.

La complejidad ciclomática de un módulo puede ser calculada desde los grafos conectados de un módulo, como se define en la fórmula 1:

$$CC = E - N + 2. \tag{1}$$

Donde E es la cantidad de arcos de un grafo, N los nodos de un grafo. Los nodos representan instrucciones o expresiones y los arcos la transferencia de control entre los nodos. La complejidad ciclomática se define en términos de caminos básicos, cuya combinación va a generar

todos los caminos posibles y conforma la noción intuitiva de "menor número de caminos".

Una forma simplificada de calcular la complejidad ciclomática es identificar las decisiones binarias. En este caso, la complejidad ciclomática va a ser igual a la cantidad de decisiones binarias más uno (Watson & Mac Cabe, 1996), como se describe en la fórmula 2.

$$CC = \text{Cantidad Decisiones binarias} + 1 \quad (2)$$

En caso de decisiones múltiples, el valor correspondiente es igual al número de alternativas menos una. Esto quiere decir que si hay tres opciones alternativas, se agregan dos, si hay cuatro, se agregan tres. En definitiva, la complejidad está dada por la cantidad de arcos agregados al flujo principal.

3.2. Lecciones aprendidas

Dado que el objetivo del presente trabajo es mejorar el error en la estimación de esfuerzo, se considera importante conocer y comprender algunos hechos: las limitaciones de FP -de tal forma que se realice un planteo que las supere-, las características y tendencias planteadas por los métodos derivados de FP, el método UCP -porque está basado en casos de uso- y la experiencia desarrollada con las unidades de medida de precisión para la estimación de esfuerzo.

3.2.1. FP y sus limitaciones

Albrecht introduce el concepto de FP en 1979 (Fenton & Pfleeger, 1997). FP es una medida del tamaño funcional de una aplicación. Como los FP se definen desde el punto de vista del cliente, se tiene en cuenta elementos que pueden ser identificados por un usuario externo. Esta propiedad permite definir una métrica que no depende de la tecnología. Además, los FP se utilizan para predecir el esfuerzo de una aplicación a ser desarrollada.

El método de FP define cinco elementos básicos: Entrada Externa (External Input, EI), Salida Externa (External Output, EO), Consulta Externa (External Inquiry, EQ), Archivo Interno (Internal File, IF), y Archivo Externo (External File, EF). La complejidad de cada elemento (Uemura et al., 1999) se clasifica en Baja, Media, o Alta, de la siguiente forma:

- a. Para cada IF y EF, la complejidad funcional se clasifica en términos del número "de tipos de elementos de datos" (Data Element Type, DET) y tipos de elementos de un registro " (Record Element Type, RET) presentes en cada IF o EF.

MARCO TEORICO DE LAS NUEVAS METRICAS

b. Para cada EI, EO, EQ, la complejidad funcional se clasifica en términos del número de referencias de archivo internas y externas (File Type References, FTR) y el número de DET.

La Tabla 2 muestra los valores de los factores de complejidad para cada elemento básico.

Tabla 2. Factores de complejidad que participan en el cálculo de FP

	Baja	Media	Alta
EI	3	4	6
EO	4	5	7
EQ	3	4	6
EF	7	10	15
IF	5	7	10

Por lo tanto, los puntos función no ajustados (Unadjusted Function Point, UFP) se definen como la suma de los productos de los factores de complejidad (W_{ij}) por el número de los elementos clasificados según su grado de complejidad (I_{ij}). Esto puede ser expresado de la siguiente forma (3):

$$UFP = \sum_{i=1}^5 \sum_{j=1}^3 W_{ij} * I_{ij}. \quad (3)$$

Los UFP pueden ser ajustados si se tiene en cuenta la tecnología empleada en el desarrollo de la aplicación. La Tabla 3 muestra las 14 características de complejidad técnica que se tienen en consideración. A cada característica se le asigna un grado de influencia dentro de un rango que va desde "sin influencia" -para la cual se asigna el valor 0-, hasta "fuerte influencia" -para la cual se asigna el valor 5-. El grado total de influencia es igual a la suma del grado de influencia (Influence Degree, ID) de cada característica de la aplicación.

En consecuencia, el Factor de Complejidad Técnica (Technical Complexity Factor, TCF) se calcula según la fórmula (4).

$$TCF = 0.65 + 0.01 \sum_{i=1}^{14} ID_i. \quad (4)$$

El tamaño de la aplicación se define en términos de FP, y se calcula como el producto de los UFP por el TCF, como se expresa en (5)

$$\text{Tamaño funcional}_{[FP]} = UFP * TCF. \quad (5)$$

Tabla 3. Características a ser consideradas en TCF

Características	ID
Comunicación de datos	
Funciones distribuidas	
Desempeño	
Configuración de usuario	
Transacciones	
Entrada de datos en línea	
Eficacia de usuario final	
Actualización en línea	
Procesamiento complejo	
Re-utilización	
Facilidad de instalación	
Facilidad de operación	
Sitios múltiples	
Facilidad de cambio	

Entre la bibliografía existente, se comprueba que Kitchenham (Kitchenham, 1997) ha resumido las limitaciones y dificultades que encontró en el uso FP. Ella señala que cuando Albrecht clasifica los elementos en complejidad Baja, Media y Alta, eso significa que se reduce la escala absoluta de conteo a una medida de escala ordinal. Formalmente, las medidas de escala ordinal no pueden ser sumadas: los elementos de complejidad Baja, Media y Alta son distintos ítems, a pesar de que se utilice un factor para realizar una supuesta conversión. Al medir el tamaño en una escala ordinal, se puede expresar que un tamaño es mayor a otro, pero no cuánto más grande es, como lo establece el método de FP. Después de analizar la construcción de FP desarrollada por Symons (Symons, 1988), quien ha simplificado el número de elementos a tener en cuenta, Kitchenham comenta que, *lamentablemente, no hay un factor de conversión estándar para igualar entradas, salidas y acceso a entidades*. Con respecto al TCF, *puntualiza que no hay evidencia de que el TCF mejore algún modelo predictivo que use FP*, y que en varios estudios empíricos, las predicciones basadas en UFP han sido tan buenas como las predicciones basadas en FP.

Otro aspecto importante a tener en cuenta es la existencia de correlación entre los elementos de FP, ya que para ser internamente

consistente, los FP deberían ser independientes. Tal correlación indica la existencia de algún tipo de falencia. Lokan (Lokan, 1999) reporta que hay evidencia de correlación entre los elementos de los datos usados en un experimento que incluye un amplio espectro de 269 proyectos. En otro caso de estudio de 40 proyectos, que muestra la existencia de correlaciones entre los elementos, Kitchenham y Känsälä (Kitchenham & Känsälä, 1993), concluyen que en ese caso, FP no tiene las características de una medida de tamaño válida. Esto se debe a que la existencia de correlación entre los elementos indica que algo podría haber sido contado más de una vez. *Es evidente que los elementos de datos afectan la complejidad de un sistema, pero parece que el método no refleja correctamente la magnitud de tal impacto.*

Otra debilidad del método FP es su inconsistencia. Después de investigar este aspecto, Kemerer informa que las diferencias reportadas en el conteo de FP de un mismo sistema son del 12%. Kitchenham y Känsälä (Kitchenham & Känsälä, 1993), *suponen que un método más simple podría reducir tal error de conteo.*

En un artículo que se consideró el punto de partida de este trabajo, Kitchenham (Kitchenham, 1997) clarifica algunas dificultades importantes que este método presenta, y que no han sido solucionadas aún, y establece algunos lineamientos sobre los cuales sería necesario trabajar, según se detalla a continuación:

- a. *Se debería considerar la simplificación del concepto de FP, de tal forma que un conteo básico pueda ser recolectado en forma automática, desde una representación temprana del sistema.*
- b. *No se debería agregar peso por complejidad.*
- c. *No se debería adicionar el conteo de cosas diferentes, sino expresarlo como un vector de medidas que describen el sistema.*

3.2.2. Características comunes de los métodos derivados de los FP

A modo de resumen, en la Tabla 4 se sintetiza un conjunto de métodos que derivan de FP. Para la comparación fueron seleccionadas las cuatro características que son esenciales del método de FP. *Se observa que dentro de los métodos definidos como estándares (filas 2, 3 y 4), existe una tendencia a simplificar el método, disminuyendo la cantidad de elementos y eliminando algunas de sus características esenciales. Esta misma tendencia (cfr. el resto de las filas) no se percibe en la comunidad de métricas en general.*

Tabla 4. Resumen de las características comunes de métodos derivados de FP

Fila	Métodos	Elementos	Clasificación de complejidad de elementos (Baja, Media, Alta)	Conversión a una unidad de medida usando un coeficiente	Ajuste por TCF
1	Puntos Casos de uso (Karner, 1993)	2	Sí	Sí	Si, agrega factores ambientales
2	Mark II (Symons, 1988; ISO/IEC 20968:2002)	3	No	Sí. Pesos promedios de la industria	Sí, agrega factores ambientales
3	FP (Albrecht 1979; ISO/IEC 20926:2003)	5	Sí	Sí	Sí
4	COSMIC-FFP (ISO/IEC 19761:2003)	1	No	No	No
5	Web Objects (Ruhe et al., 2003)	9	Sí	Sí	No
6	FP modificado (Ceddia & Dick, 2004)	2	Sí	Sí	Sí
7	UCP (Carroll, 2005)	3, agrega reportes	Sí	Sí	Sí, agrega factores de experiencia
8	Objects Points (Issa et al., 2005)	3	Sí	Sí	No, pero agrega re-uso.
9	USP (Braz & Virgilio, 2006)	6	Sí, pero no mantiene la relación de 3 a 1 (UCP)	Sí	Sí. Agrega factores ambientales.
10	UML Points (Kim et a. 2006)	UCP (2) + Class Point (7)	Sí	Sí	Sí, pero agrega factores ambientales

3.2.3. Puntos casos de uso y sus limitaciones

En 1993 Karner (Karner, 1993) introduce el concepto UCP, basado en FP y el Proceso de Objectory. El proceso de conteo se basa en el modelo de caso de uso. Los UCP identifican sólo dos elementos: Actores y Casos de uso. La complejidad de cada elemento se clasifica como Baja, Media o Alta. La misma se establece mediante el análisis de las características del Actor, y para los Casos de uso, a través del número de transacciones y el número de objetos de análisis involucrados en cada caso de uso. La Tabla 5 muestra los valores de los factores de complejidad para cada elemento básico.

Tabla 5. Factores de complejidad que participan en el cálculo de UCP

	Baja	Media	Alta
Actor	1	2	3
Caso de uso	5	10	15

Por lo tanto, los puntos casos de uso no ajustados (Unadjusted Use Case Point, UFP) se definen como la suma de los productos de los factores de complejidad (W_{ij}) por el número de los elementos clasificados según su grado de complejidad (I_{ij}). Esto puede ser expresado de la siguiente forma (6):

$$UUCP = \sum_{i=1}^2 \sum_{j=1}^3 W_{ij} * I_{ij} . \tag{6}$$

Los UUCP pueden ser ajustados si se tiene en cuenta la tecnología y el ambiente de la aplicación. La Tabla 6 muestra las 13 características de complejidad técnica y la Tabla 7, las 8 características ambientales, que se deben tener en cuenta. A cada característica se le asigna un grado de influencia dentro de un rango que va desde "sin influencia" -para la cual se asigna el valor 0-, hasta "fuerte influencia" -para la cual se asigna el valor 5-. En consecuencia, el TCF se calcula según la fórmula (7) y el Factor ambiental (Evironmental Factor, EF) según la fórmula (8).

$$TCF = 0.6 + 0.01 \sum_{i=1}^{13} D_{li} * W_i . \tag{7} \quad EF = 1.4 - 0.03 \sum_{i=1}^8 D_{li} * W_i . \tag{8}$$

MARCO TEORICO DE LAS NUEVAS METRICAS

Tabla 6. Características a ser consideradas en TCF

Características	W
Sistemas distribuidos	2
Objetivos de desempeño aplicados, tanto en la respuesta como en el rendimiento	1
Eficacia de usuario final	1
Procesamiento interno complejo	1
Re-uso	1
Facilidad de Instalación	0.5
Facilidad de operación y uso	0.5
Portabilidad	2
Facilidad de cambio	1
Concurrencia	1
Características de seguridad especiales	1
Acceso directo a terceras partes	1
Facilidad de entrenamiento para usuarios especiales	1

Tabla 7. Características del EF

Características	W
Familiaridad con "Objectory"	1.5
Trabajadores jornada reducida	-1
Capacidad del analista	0.5
Experiencia en la aplicación	0.5
Experiencia en orientación a objetos	1
Motivación	1
Dificultad del lenguaje de programación	-1
Requerimientos estables	2

Entonces, el tamaño de la aplicación se define en términos de UCP, y se calcula como el producto de los UUCP por el TCF y el EF, como se expresa en (9).

$$\text{Tamaño funcional}_{[UCP]} = UUCP * TCF * EF. \quad (9)$$

Sin embargo, se puede observar que *entre los autores que han usado los UCP no se distingue un criterio único aplicado para el cálculo de la complejidad de un caso de uso*: por ejemplo, Anda (Anda et al., 2005) y Diev (Diev, 2006) no tienen en cuenta los objetos de análisis. Además, también existen diferentes criterios para identificar una transacción. Unos definen la transacción como un conjunto atómico de actividades entre el actor y el sistema, que debe ser realizado en forma completa (Anda et al., 2001; Anda et al., 2002; Anda et al., 2005; Carroll, 2005; Diev, 2006). Otros claramente definen las diferencias entre transacciones y escenarios (Diev, 2006), mientras otros no hacen ninguna diferencia entre ellos (Anda et al. 2005). Hay otro que define el número de transacciones como el número de transacciones dentro de los escenarios de un caso de uso (Issa, 2006), y aún hay otro más que, en la búsqueda de un modo de automatizar el cálculo UCP, identifica las

transacciones a partir del estímulo de actor (Kusumoto et al., 2004). *En este trabajo, se ha utilizado el criterio de Kusumoto para identificar las transacciones debido a que tiene un grado de objetividad mayor, si se lo compara con las opciones discutidas anteriormente. Además, no se han considerado los objetos de análisis para facilitar la comparación de los resultados alcanzados con aquellos de los trabajos publicados.*

3.2.4. Unidades de medida de precisión en la estimación de esfuerzo

Existen diferentes formas para medir la precisión de la estimación de esfuerzo de una aplicación de software. En este trabajo se ha utilizado dos de ellas: MRE y Z.

- a. La MRE, se define como el valor absoluto del cociente entre, la diferencia entre el esfuerzo real (Actual Effort, AE) y el esfuerzo estimado (EE), y el esfuerzo real, como lo expresa la fórmula (10):

$$MRE = | AE - EE | / AE \quad (10)$$

Conte, Dunsmore, y Shen (Fenton & Pfleeger, 1997) sugieren que un nivel aceptable de error es algo menor o igual a 0.25. Definen la medida de Calidad de la predicción para un conjunto de n proyectos, si k es el número de proyectos cuya MRE es menor o igual que q, como se muestra en (11)

$$PRED(q) = k / n. \quad (11)$$

Sugieren que una técnica de estimación es aceptable si PRED(0.25) es al menos 0.75 (Fenton & Pfleeger, 1997).

- b. Z se define como el cociente entre EE y AE, según se expresa en la fórmula (12):

$$Z = EE / AE. \quad (12)$$

Kitchenham (Kitchenham et al., 2001) examina el concepto de precisión en algunas medidas estadísticas. Con esta finalidad, analiza la media de MRE (MMRE) y PRED(0.25), y demuestra que MMRE y PRED(0.25) son, respectivamente, medidas de la dispersión y curtosis de la variable Z. Por lo tanto, *se considera a Z una medida de precisión* mientras que medidas estadísticas tales como MMRE y PRED(0.25) se entienden como medidas de las propiedades de Z. Estos autores sugieren que son necesarias medidas de locación central y asimetría de Z, así como medidas de dispersión y curtosis. Además, dado que la distribución de Z es *no-normal*, pueden ser necesarias medidas no paramétricas de esta propiedad. Por esta razón, *los boxplots de Z son alternativas útiles para resumir las métricas en forma simple.*

Stensrud (Stensrud et al., 2002) afirma que la razón implícita en la utilización de una medida de error relativa como MRE, en vez de una absoluta, se debe a que se busca una medida que sea independiente del

tamaño del proyecto. Investigaron si esta afirmación implícita es verdadera para varios juegos de datos (Albrecht, Kemerer, Finnish, DMR y Accenture-ERP). *Los resultados sugieren que la MRE no es independiente del tamaño del proyecto.* Más bien, la MRE es más grande para pequeños proyectos que para proyectos grandes. Una consecuencia práctica de esto es que el estimador de un pequeño proyecto falsamente puede estimar un MRE demasiado bajo. De la misma manera, lo inverso puede suceder si predice un proyecto grande. Para estos estimadores, es importante saber que la MMRE no es una medida apropiada de la MRE esperada para proyectos pequeños y grandes. Ellos recomiendan, por lo tanto, que los conjuntos de datos sean divididos en dos o más sub-muestras, y que la MMRE sea reportada por sub-muestra.

Habiendo tomando conocimiento de las advertencias descritas en los párrafos anteriores, *se ha seleccionado a la variable z* para medir la precisión en la estimación de esfuerzo en el caso de estudio desarrollado en el presente trabajo.

3.3. Conclusiones del capítulo

En este capítulo se describieron las bases sobre las cuales se van a definir las nuevas métricas. Dichas bases están compuestas por un conjunto de conceptos definidos por otros autores (*casos de uso, transacciones, objetos de entidad y complejidad ciclométrica*), que forman parte del vocabulario común de la ingeniería de software, y un conjunto de lecciones aprendidas que se sintetizan de la siguiente forma:

- a. No hay un factor de conversión estándar para igualar entradas, salidas y acceso a entidades
- b. No hay evidencia de que el TCF mejore algún modelo predictivo que use FP
- c. Es evidente que los elementos de datos afectan la complejidad de un sistema, pero parece que los FP no reflejan correctamente la magnitud de tal impacto
- d. Se supone que un método más simple podría reducir el error de conteo
- e. Se debería considerar la simplificación del concepto de FP, de tal forma que un conteo básico pueda ser recolectado en forma automática, desde una representación temprana del sistema.
- f. No se debería agregar peso por complejidad.
- g. En el conteo, no se deberían sumar cosas diferentes, sino expresarlo como un vector de medidas que describen el sistema.

MARCO TEORICO DE LAS NUEVAS METRICAS

- h. Los métodos derivados de FP y hoy considerados estándares tienden a simplificar el método original
- i. El criterio de Kusumoto (Kusumoto et al., 2004) para identificar transacciones define una forma de realizarlas con un mayor grado de objetividad
- j. La distribución de Z no es normal y los boxplots de Z son alternativas útiles para graficar los resultados
- k. La MRE no es independiente de tamaño del proyecto

4. Descripción de las unidades de medida propuestas

En este capítulo se describen las tres unidades de medidas que se utilizarán en la medición de aplicaciones. Las mismas se calculan a partir de la descripción textual de los casos de uso, y expresan características diferentes de una aplicación: funcionalidad, datos y complejidad. Además, se presentan dos ejemplos de cómputo de las métricas propuestas utilizando dos aplicaciones de software provenientes de la industria. El segundo ejemplo incluye la utilización de documentos complementarios. Finalmente, se analiza la consistencia de las métricas propuestas con la teoría de la medición.

4.1. Basadas en casos de uso.

Se selecciona el modelo de casos de uso como base de cálculo, porque, al ser éste un *documento informal*, cuenta con una *amplia aceptación* entre los programadores y porque es posible usarlo en una etapa temprana del ciclo de vida del desarrollo de software. La Figura 2 muestra, en una secuencia temporal, los diferentes entregables que habitualmente se desarrollan durante este proceso. Obsérvese que una vez concluida la definición de los requerimientos, se puede definir una Lista de casos de uso.

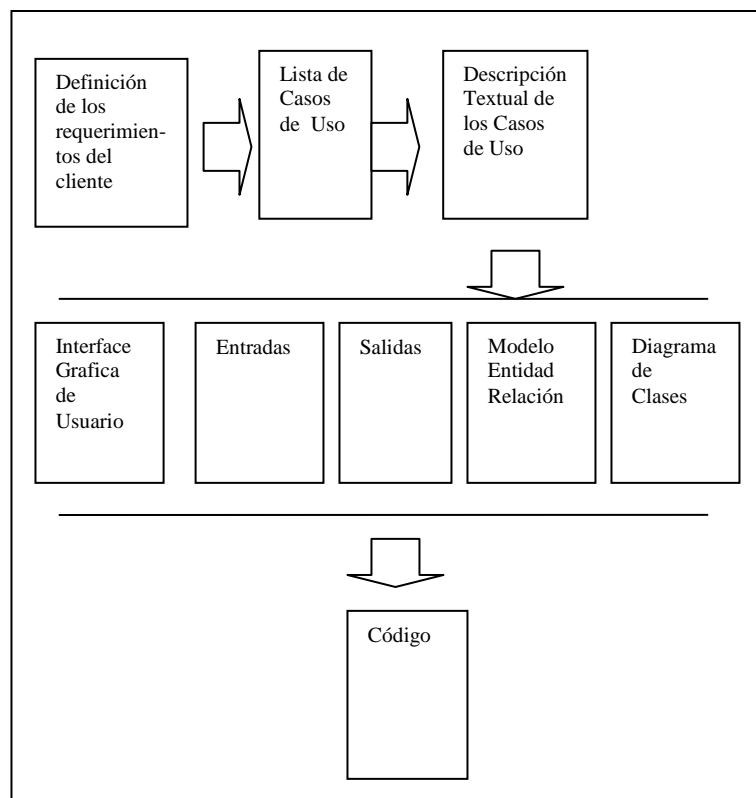


Fig. 2. Secuencia temporal de los entregables habitualmente desarrollados

Una vez definida la lista, se puede realizar una descripción textual de los casos de uso. Con esta información, el analista puede definir la Interfaz gráfica del usuario y las Entradas y Salidas del Sistema. Finalmente, se puede dibujar el Modelo Entidad Relación. Si, por ejemplo, el método de FP fuese usado para calcular el tamaño de la aplicación, sería necesario usar la Interfaz gráfica del usuario, la definición de las Entradas y Salidas del Sistema y del Modelo Entidad Relación, que estarán disponibles en un momento más tardío en el ciclo de desarrollo de la aplicación.

Si, en cambio, se utilizan las líneas de código para medir el tamaño de una aplicación, el tamaño podrá ser conocido recién al final del desarrollo del producto de software.

El hecho de que los casos de uso estén finalizados en una etapa temprana es una característica importante, debido a que posibilita realizar una temprana estimación de costo.

4.2. Expresan funcionalidad, datos y complejidad

Para cuantificar algo real con cierto grado de complejidad y abarcar con más profundidad esa realidad, puede ser necesario medir más de un atributo. Este es el caso, por ejemplo, del tamaño de una persona, que suele expresarse en términos de peso y altura. Por lo cual, resulta natural que para cuantificar un producto de software sea necesario medir más de un atributo interno. Por este motivo, se seleccionan tres atributos esenciales y al mismo tiempo básicos: funcionalidad, datos y complejidad.

Con el objetivo de identificar funciones, se seleccionan a las *transacciones* (T). Éstas son elegidas porque, al tratarse de una respuesta de un sistema a un estímulo dado, es posible identificarlas con cierto grado de objetividad a partir del estímulo del actor, logrando de esta forma la identificación de una funcionalidad concreta.

Con el objetivo de identificar datos, se utilizaron los *objetos de entidad* (Entity Objects, EO), porque contienen la información persistente de los sistemas, como fuera claramente definido por Jacobson (Jacobson et al., 1992).

Con el objetivo de identificar complejidad, se seleccionó a la complejidad ciclomática, porque ofrece un modo simple de cálculo, que puede ser adaptado para ser aplicado en la descripción textual de los casos de uso. Esta unidad de medida se llamará *camino*s (Paths).

Por lo tanto, se definen tres unidades de medida: T, EO y P. La cuantificación, tanto del caso de uso como de la aplicación, se mide y expresa tanto en T, como en EO, o en P. Éstas son medidas que

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

expresan características diferentes de una aplicación: funcionalidad, datos y complejidad.

4.2.1. Transacciones

Una transacción está formada por un estímulo, que es disparado por un actor en el inicio de la transacción, y una respuesta, la cual es realizada mediante las acciones del sistema. Esta respuesta debe agregar algún valor al actor. Para clarificar al actor y a la identificación del estímulo, se puede realizar un análisis sintáctico del texto. El actor es el sujeto de la oración y el estímulo emitido o la acción a ser realizada por el actor, va a estar representado por el verbo. Se identifican en forma similar a la planteada por Kusumoto (Kusumoto et al., 2004), quien sugiere que, del conjunto total de oraciones del caso de uso, se identifiquen como transacciones aquellas relativas a la operación del actor y a la respuesta del sistema.

Si la transacción se graficara en un diagrama de secuencia, sería el equivalente a una secuencia de mensajes, disparados en el sistema a partir del estímulo del actor. La transacción termina cuando el actor recibe del sistema la respuesta que estaba esperando.

El número de estímulos realizados por el actor va a ser el número de transacciones de un caso de uso. En la Figura 1 se muestra un caso de uso. Los dos verbos que definen el número de transacciones del caso de uso están resaltados con letra cursiva. Los dos estímulos que el actor (usuario) realiza sobre el sistema son identificados: seleccionar y grabar un nuevo producto. La Tabla 8 muestra los verbos, el hecho de si los verbos son contados, o no, y los motivos por los cuales se cuentan o se descartan.

Por lo tanto, en el ejemplo del caso de uso "Dar de alta a un producto en un catálogo" de la Figura 1, se identifican 2 T.

La cantidad de T de la aplicación va a ser la suma de la cantidad de transacciones de cada caso de uso de la aplicación. Por lo tanto, el tamaño funcional de una aplicación de n casos de uso va a ser expresada en T, como se muestra en (13).

$$\text{Tamaño Funcional de la Aplicación } [T] = \sum_{i=0}^n \text{Cantidad de T de los Casos de uso.} \quad (13)$$

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

Tabla 8 – Actor y verbos del caso de uso “Dar de alta a un producto en un catálogo”

Verbo	Contar	Justificación
<i>seleccionar</i>	√	Es un estímulo del sistema porque dispara una consulta a la base de datos. Si hubiera sido una selección entre un conjunto de alternativas que no implicara una consulta a la base de datos, no hubiera sido considerado como un estímulo
presionar enter	×	Esta acción es parte de la acción de selección; no define una acción diferente
acceder completar cargar	×	Este es un paso anterior a la realización del estímulo por parte del actor. No se consideran estímulos los ingresos de datos en una pantalla de entrada
<i>grabar</i>	√	Es un estímulo del sistema
cancelar	×	No se considera un estímulo
Dejar	×	Es una consecuencia de la transacción, no una transacción en sí

Intuitivamente, se reconoce al tamaño como un concepto de medida importante. Según el planteo de los autores (Briand et al., 1996), el tamaño no puede ser negativo (No negatividad), y se espera que sea nulo cuando un sistema no contiene ningún elemento (Valor Nulo). Cuando los módulos no tienen elementos en común, se espera que el tamaño sea aditivo (Aditividad). El hecho de que T satisface las propiedades de No negatividad, Valor Nulo y Aditividad –como se puede deducir del Apéndice II-, confirma que es correcto considerar a T como una unidad de medida de Tamaño del software.

A continuación, se detallan algunas particularidades a tener en cuenta en el cálculo de T:

- a. *Un caso de uso que extiende⁶ o usa⁷ otro caso de uso.* Se cuenta en forma independiente cada caso de uso, sin modificar las pautas establecidas en este documento. En principio, no tiene por qué haber repetición en las T de un caso de uso que extiende o usa otro caso de uso. Si la hubiera, no se cuenta más de una vez la misma transacción.

⁶ Representan una parte de la funcionalidad del caso que no siempre ocurre; no necesariamente se originan por un error o una excepción.

⁷ La misma funcionalidad del sistema es accedida por varios casos de uso.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

- b. *Flujo básico o alternativo.* No se realiza en el cálculo ninguna distinción entre el primero y/o los sub-flujos. Se cuentan los estímulos del actor, independientemente de que formen parte del flujo básico o alternativo o sub-flujos.
- c. *Diagramas de actividad.* En el caso de contar con estos diagramas, u otra forma de documentar una descripción de los casos de uso con un mayor nivel de detalle, se sigue el mismo criterio: identificar los estímulos del actor, agregar al caso de uso original las nuevas T, evitando contar una transacción más de una vez. Se encuentra un ejemplo en 4.5.
- d. *Direccionamiento a otra página.* Si es una página estática, se cuenta como una sola transacción. Si es una página dinámica, hay que evitar duplicar el cómputo de la transacción.
- e. *Funcionalidad repetida.* Hay opciones que se usan en muchas partes del sistema, por ejemplo, la selección de una fecha en un calendario, un marco de opciones en un sistema web, etc. No se debe repetir su cómputo, sólo se cuentan una vez.
- f. *Diseño de pantallas.* Es frecuente que la descripción textual de los casos de uso esté acompañada por un diseño detallado de pantalla. Éste puede facilitar el cálculo de las T. La cantidad de T de una pantalla es equivalente a la cantidad de botones que tiene una pantalla. Por ejemplo: botón de "grabar" y "borrar".
- g. *No se cuentan:*
 - 1. repetición
 - 2. las acciones Cancelar, Salir o Home
 - 3. validaciones
 - 4. atributos, check box, radio button
 - 5. alternativas planteadas dentro de la transacción.
 - 6. escenarios⁸
 - 7. pre-condiciones y post-condiciones

4.2.2. Objetos de entidad

Jacobson (Jacobson et al., 1992) define los objetos de entidad como aquellos objetos que manejan la información persistente del modelo. La mayor parte de estos objetos son identificados en los casos de uso y son por lo general, los objetos obvios que participan en el modelo de objetos

⁸ Instancias o específicas ocurrencias de un caso de uso.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

del dominio de problema. Corresponden a conceptos de la vida real, que por lo general existen fuera del sistema.

Para clarificar la identificación de objeto de entidades es conveniente realizar un análisis sintáctico del texto de los casos de uso. Primero, se identifican todos los sustantivos y sólo se seleccionan como objetos de entidad aquellos que constituyen la parte del el vocabulario del dominio del problema. En la Tabla 9 se muestran los sustantivos, si los sustantivos son contados, o no, y los motivos por los cuales se cuentan o se descartan.

Tabla 9 – Sustantivos del caso de uso “Dar de alta a un producto en un catálogo”

Sustantivo	Contar	Justificación
Categoría	√	Es una entidad que pertenece al vocabulario del dominio del problema
Sub-categoría	√	Es una entidad que pertenece al vocabulario del dominio del problema
Producto	√	Es una entidad que pertenece al vocabulario del dominio del problema
Propiedades	×	Es una entidad que identifica los atributos de producto, categoría-o sub-categoría
Pantallas Cambios Sistema Tecla Operación Lista	×	Son entidades que no pertenecen al vocabulario del dominio del problema, son comunes a todos los sistemas

El número de objetos de entidad en un caso de uso es igual al número de objetos de entidad diferentes en un caso de uso, por lo tanto, cada objeto de entidad es contado sólo una vez, no importa cuántas veces se repite. Esto es así porque, al medir con objetos de entidad, el esfuerzo – medido en horas trabajadas- depende de las entidades diferentes que están implicadas en un caso de uso, independientemente de que éstas sean usadas en más de una T. Por la misma razón, el número de objetos de entidad de un módulo –conjunto de casos de uso- es calculado de un modo similar. Entonces, el cálculo del número total de objetos de entidad por módulo se realiza de la forma siguiente:

- a. Se identifican todos los objetos de entidad que participan en los casos de uso.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

b. Los objetos de entidad se cuentan solamente una vez, incluso si están repetidos dentro de un caso de uso, o en cualquiera de los otros casos de uso que constituyen el módulo.

Por lo tanto, en el ejemplo del caso de uso "Dar de alta a un producto en un catálogo" de la Figura 1, se identifican 3 EO.

La cantidad de objetos de entidad por aplicación va a estar dada por la suma de objetos de entidad por módulo. Entonces, si una aplicación tiene n módulos, esto se expresa según la fórmula (14):

$$\text{Tamaño de datos [EO] de la Aplicación} = \sum_{m=0}^n \text{Cantidad de EO diferentes por modulo.} \quad (14)$$

Con respecto al criterio de modularización adoptado, se destacan los siguientes aspectos:

a. si la aplicación ya estaba dividida en módulos, se respetó el criterio definido por los desarrolladores

b. si la aplicación no estaba dividida en módulos, se definieron los módulos agrupando los casos de uso que compartían los mismos objetos de entidad y/o interfaces. Esto se aplicó con un criterio amplio, lo que quiere decir que no se excluye la posibilidad de que un objeto de entidad sea compartido por más de un módulo

Según el planteo de los autores (Briand et al., 1996), el tamaño no puede ser negativo (No negatividad), y se espera que sea nulo cuando un sistema no contiene ningún elemento (Valor Nulo). También, cuando los módulos no tienen elementos en común, se espera que el tamaño sea aditivo (Aditividad). En el caso de los EO, si bien existen EO comunes a más de un módulo, al pertenecer a módulos diferentes, se los consideran distintos. El motivo por el que así se hace es que el tamaño de un módulo, en términos de cantidad de datos, se expresa como la cantidad de EO diferentes del módulo.

El hecho de que los EO satisfaga las propiedades de No negatividad, Valor Nulo y Aditividad planteadas por Briand (Briand et al., 1996) y resumidas en el Apéndice II, confirma que es correcto considerar a los EO como una unidad de medida de Tamaño del software.

4.2.3. Caminos

Al definir la complejidad ciclomática, se dijo que el criterio de cálculo de complejidad en un grafo se puede definir como la cantidad de arcos agregados al flujo principal. Es posible aplicar este criterio a cada T de la descripción textual de un caso de uso si se tiene en cuenta lo siguiente:

a. cada transacción es un grafo.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

- b. el flujo principal de la transacción es el texto principal del caso de uso, por lo tanto, el valor de P es 1.
- c. cada expresión similar a "si... entonces" es considerada una decisión binaria.
- d. cada expresión similar a "en caso de: A... entonces, B... entonces, C... entonces" es considerada una decisión múltiple.

Por lo tanto, la complejidad de un caso de uso se mide de la siguiente forma:

- a. para obtener los P de una transacción se debe:
 - 1. contar la cantidad de decisiones binarias
 - 2. identificar las decisiones múltiples, y asignar un número igual a la cantidad de opciones, menos una.
 - 3. agregar 1 a la suma de los ítems 1 y 2.
- b. para obtener los P de un caso de uso: se debe sumar los P de cada transacción del caso de uso.

En la Figura 1 se muestra un caso de uso con dos T. En una de ellas está la expresión "SI ...ENTONCES" resaltada con letras mayúsculas. Es una decisión binaria, por lo tanto el caso de uso tiene 3 P.

Siendo n la cantidad de casos de uso y m –número que varía por caso de uso- la cantidad de T por caso de uso, la complejidad de la aplicación estará dada por la suma de los P de cada caso de uso, que es, a su vez, la suma de los P de las T que conforman ese caso de uso, como se define en (15)

$$\text{Complejidad de la Aplicación } [P] = \sum_{i=0}^n (\text{Cantidad de P para T1} + \dots + \text{Cantidad de P para Tm}) . \quad (15)$$

A continuación se detallan algunas consideraciones a tener en cuenta en el cálculo de P:

- a. *Se considera que un P es una relación entre dos nodos.* Los nodos son una parte del texto del caso de uso. La relación o secuencia entre los textos definen los caminos. En el caso del camino principal, se puede distinguir un texto inicial y otro final, y una relación secuencial.
- b. *Se identifican a las T, unidad elemental para calcular la complejidad,* de la misma manera en que se lo hace en el cálculo de T, es decir, a partir del estímulo del actor.
- c. *La transacción puede tener un camino principal y caminos alternativos (Alternative Path, AP).* Esta distinción es muy frecuente, puesto que

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

tanto en la escritura de los casos de uso, como en la programación, se suele escribir en primer lugar el flujo completo principal. Luego se le agregan los otros caminos o flujos alternativos. Normalmente, los caminos alternativos se definen a partir de una condición que se debe cumplir.

- d. *Si una T solamente tiene un camino principal, P es igual a 1.*
- e. *Si tiene caminos alternativos, el número de P será 1 (el camino principal) más el número de AP. Si el caso de uso es vacío, es decir, no se puede identificar ninguna transacción, entonces P es igual a cero.*
- e. *Cada transacción es considerada como una componente diferente.* Las relaciones entre T no son tenidas en cuenta, dado que no son consideradas componentes conectadas. Cada transacción comienza con el estímulo del actor y termina con la recepción del retorno o respuesta. Además, los caminos de cada transacción deben ser completamente diferentes. Si se repite un camino, esto es un error, por lo cual, el mismo no debe ser contado. Ahora bien, puede suceder que se repita una decisión dentro de un mismo caso de uso, por ejemplo, la validación de una fecha. En este caso, lo que se está repitiendo es un elemento. Entonces, si el mismo define un camino diferente, es decir, que plantea una relación distinta entre elementos, debe ser contado.
- f. *Las T son consideradas componentes disjuntas,* lo que significa que no comparten los mismos elementos (nodos), ni la relación entre ellos (caminos).

Briand (Briand et al., 1996) define la complejidad como un atributo intrínseco de un objeto, que depende de la existencia de una relación entre los elementos del sistema –nodos, desde la definición aquí propuesta- (cfr. Resumen en Apéndice II). También define que la complejidad no debe ser negativa (No negatividad) y que debe existir un valor nulo, en caso de que no existan relaciones entre los elementos de un sistema (Valor Nulo). Además, define que la complejidad no debe ser sensible a la representación del sentido de los arcos de las relaciones (Simetría).

Así mismo, la complejidad de un sistema debe ser mayor o igual a la suma de las complejidades de cualquier colección de sus módulos, si no existen módulos que comparten las relaciones entre elementos (Monotonidad). Se considera que los P no comparten elementos, aunque existan textos comunes a dos casos de uso, ya que son considerados elementos distintos, por pertenecer a casos de uso diferentes.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

Además, la complejidad de un sistema formado por componentes disyuntas es igual a la suma de las complejidades de cada componente (Aditividad).

El hecho de que P satisfaga las propiedades de No negatividad, Valor Nulo, Simetría, Monotonicidad del Módulo, y Aditividad de Módulos disjuntos, confirma que es correcto considerar a P como una unidad de medida de complejidad del software.

Es importante resaltar que P captura la *complejidad lógica del negocio*. La lógica, o reglas del negocio, suelen estar contenidas en frases tales como "si se cumple esta condición, entonces...", "si esto pasa, se hará...", "si sucede... se hará...", que no son otra cosa que decisiones binarias o múltiples. Por lo tanto, la complejidad lógica de un caso de uso puede ser medida en P.

4.3. Ejemplo de cómputo de las métricas propuestas en un sistema de consulta de viajes

En este punto se describe un ejemplo, basado en una aplicación real del ámbito de la industria del software, donde se muestra el cómputo de las métricas propuestas en un Sistema de consulta de viajes, que tiene dos casos de uso:

- Seleccionando tipo de consulta (cfr. Figura 3 y 4)
- Consultando sobre viajes ida sólo (cfr. Figura 5 y 6)

1. El usuario selecciona qué tipo de consulta de viajes desea realizar entre las opciones que se le presentan en la pantalla
 - *Ida y Vuelta*, opción por la cual la consulta a realizar tiene predefinido el origen del viaje y el destino de la vuelta.
 - *Ida Sólo*, opción mediante la cual se realiza la consulta de un tramo (ida)
 - *Múltiples Destinos*, opción por la cual se realiza la consulta de un viaje compuesto por varios tramos

Fig. 3. Descripción textual del caso de uso "Seleccionando tipo de consulta"



Fig. 4. Pantalla del caso de uso "Seleccionando tipo de consulta"

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

1. El usuario selecciona como tipo de viaje, la opción "*Ida Sólo*"
2. El sistema muestra, por debajo de las opciones de tipo de viaje, los campos que conforman este tipo de (pantalla "*Consulta viajes ida solo*"):
 - *Origen Salida*, ciudad o aeropuerto desde donde sale el vuelo. Este campo cuenta con la opción de ayuda para la consulta de los códigos de aeropuertos. (Ver Caso de Uso Obteniendo ayuda de código de aeropuertos)
 - *Destino Salida*, ciudad o aeropuerto a donde llega el vuelo. Este campo cuenta con la opción de ayuda para la consulta de los códigos de aeropuertos. (Ver Caso de Uso Obteniendo ayuda de código de aeropuertos)
 - *Fecha Salida*, con la posibilidad de ingresarla manualmente o a través del botón calendario, el cual muestra dos meses a la vez
 - *Hora Salida*, combo con la posibilidad de seleccionar, en lugar de una hora fija, "*Mañana*", "*Tarde*" o "*Noche*". Por default, estará seleccionada la opción "*Indistinta*"
Por debajo de los datos se muestran dos botones:
 - "**Borrar**", que permite limpiar los datos ingresados, simplemente haciendo clic sobre el mismo
 - "**Más Opciones de Búsqueda**", que permite ingresar a la pantalla de consulta avanzada (Ver Caso de Uso Especificando parámetros adicionales)
3. El usuario presiona "**Buscar Vuelos y Precios**"
4. El sistema asume, en caso de que no haya sido especificado lo contrario con parámetros adicionales (Ver Caso de Uso Especificando parámetros adicionales), que la consulta se realiza para un adulto y en clase turista.
5. El sistema valida:
 - Que estén completos los campos *Origen Salida*, *Destino Salida* y *Fecha Salida*.
 - En los campos *Origen Salida* y *Destino Salida*, que el código de aeropuerto sea válido, o en caso que fuera una ciudad, que dicha ciudad exista. En caso contrario, si el valor ingresado es de tres caracteres, se abre la ventana de ayuda de código de aeropuertos mostrando un listado con todos los aeropuertos cuyo código contenga el texto ingresado. Si no hubiera ninguna coincidencia, se muestra el listado vacío. (Ver Caso de Uso Obteniendo ayuda de código de aeropuertos)
 - Que *Fecha Salida* sea una fecha válida (formato fecha)
 - Que *Fecha Salida* sea posterior a la fecha actual
 - Que *Fecha Salida* sea menor o igual a 300 días a partir de la fecha del día en que se está realizando la consulta

Si alguna de estas validaciones no se cumple, se muestra el mensaje correspondiente a dicha validación.
1. Si las validaciones anteriores fueron correctas, el sistema muestra la pantalla popup "*Itinerarios*", con el resultado de la búsqueda, mostrando los vuelos que cumplen con los parámetros especificados. Cada una de las opciones se muestra con el nombre "*Itinerario n°#*" y con una tabla en donde se describe

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

cada uno de los tramos, con los siguientes datos: *Número, Compañía, Vuelo, Equipo, Fecha, Desde, Sale, Hasta, Llega, Fecha, Clase, Duración, Millas, Escala, Tiempo de espera en la escala y Cambio de avión*. Por encima, de esta tabla se muestran: *Precio del viaje, Impuestos y tasas, Total*

Por debajo de los resultados, se muestra el botón "Nueva Búsqueda", que permite realizar una nueva búsqueda, modificando los parámetros anteriormente ingresados. Es decir, que al volver a la pantalla de búsqueda, se muestran los valores ingresados en la búsqueda anterior.

Fig. 5. Descripción textual del caso de uso "Consultando viajes ida sólo"

The screenshot shows a web application interface for flight searches. On the left is a navigation menu with a 'Salir' button and links for 'Consulta de Vuelos y Precios' and 'Administración'. The main content area is titled 'CONSULTA DE VUELOS Y PRECIOS' and features three radio buttons for 'Tipo de Viaje': 'Ida y Vuelta', 'Ida Solo' (which is selected), and 'Múltiples Destinos'. Below this are input fields for 'Origen' and 'Destino', both with the placeholder 'Ciudad o Cód. Aeron.' and a 'Salida' label. The 'Fecha' field has a date format 'dd/mm/aaaa' and a calendar icon. The 'Hora' field has a dropdown menu currently set to 'Indistinta'. At the bottom, there are two red links: 'Borrar' and 'Más Opciones de Búsqueda', and a green 'Buscar Vuelos y Precios' button.

Fig. 6. Pantalla del caso de uso "Consulta viajes ida sólo"

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

Las Tablas 10, 11 y 12 muestran el detalle del cálculo de T, EO y P, respectivamente.

Tabla 10. Detalle del cálculo de T

Nombre del Caso de uso	T	Acciones realizadas por el actor	Justificación
Seleccionando tipo de consulta	1	selecciona	La única acción que realiza el usuario es la selección. Si bien hay tres opciones, no se cuentan como transacciones, puesto que se van a contar en el detalle de la consulta. Administración, que figura en la pantalla, es otro caso de uso, por lo tanto no se cuenta como transacción de éste.
Consultando viajes ida sólo	2	Borrar, Buscar Vuelos y Precios.	No se cuentan las validaciones, ni los atributos. Las referencias a casos de uso, por ejemplo, "Código de Aeropuerto" y "Más Opciones de Búsqueda", se contarán cuando se cuente el caso de uso. No se cuenta la repetición de la acción: "Nueva búsqueda". En la pantalla también se encuentra el botón para acceder a la selección de la fecha: se debe contar una sola vez. En este caso, como ya ha sido contado, no se tiene en cuenta. Tampoco se tiene en cuenta el link a salir.
Total	3		

Tabla 11. Detalle del cálculo de EO

Nombre del Caso de uso	EO	Objetos de entidad	Justificación
Seleccionando tipo de consulta	1	Viajes	
Consultando viajes ida sólo	4	Viajes Ciudad Aeropuerto Itinerarios	No se tiene en cuenta Vuelos porque es lo mismo que Itinerarios. También, Itinerario o tramo se está usando para describir lo mismo. Escalas es similar a Ciudad/Aeropuerto.
Total	4		Si se considera que estos dos casos de uso forman un módulo

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

Tabla 12. Detalle del cálculo de P

Nombre del Caso de uso	P	Palabras que identifican los pasos alternativos	Justificación
Seleccionando tipo de consulta	1	--	Consiste en una selección entre opciones diferentes, no está marcando flujos alternativos. Además, en este caso, las opciones no se cuentan como caminos alternativos porque cada una es una transacción alternativa, la cual es evaluada oportunamente. No se cuentan las relaciones entre T.
Consultando viajes ida sólo	4	"Valida" "Si el valor ingresado"	Existen 2 T: a. Borrar: no tiene caminos alternativos. b. "Buscar Vuelos y Precios" se identifican 2 decisiones binarias. El conjunto de validaciones está contado como una unidad, puesto que se deduce del texto que hay un único mensaje de error. En el caso "si el valor ingresado es de tres caracteres" se identifica como decisión binaria porque está generando un paso alternativo que es: "se abre la ventana de ayuda de código de aeropuertos"
Total	5		

4.4. Ejemplo de cómputo de las métricas propuestas usando documentos complementarios

En este punto se describe un ejemplo, basado en una aplicación real del ámbito de la industria del software, donde se muestra el cómputo de las métricas propuestas en un sistema de solicitud de crédito por internet, para un caso de uso complejo: Gestionando crédito en línea (cfr. Figura 7). Para facilitar su descripción, se utiliza un diagrama de actividad (cfr. Fig. 8), para completar y clarificar la información que brinda el caso de uso. Se debe interpretar cuidadosamente el caso de uso y el diagrama de actividad, de tal forma que se evite contar dos veces lo mismo. En el texto del caso de uso se destacan las T con verbos en *itálica*, los OE en **negrita** y los AP en **mayúsculas**.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

1. El **interesado registrado** accede al sitio y selecciona la opción para iniciar un trámite on-line. SI EN LA PÁGINA principal el usuario se registra, se retoma el trámite al punto en que fue abandonado. (ver caso de uso: "Identificándose como usuario en el Sistema")
2. SI EL USUARIO no viene del calculador de cuotas, la primera parte de la pantalla mostrará el calculador sin datos. Si el usuario viene del calculador de cuotas, se incorporan esos datos a la primera parte de la pantalla de originación de créditos. Ver Caso de Uso: Simulando Cuotas de un crédito, para más detalles. En este último caso, el usuario podrá corregir los datos ingresados y recalcular el plan de cuotas.
3. Una vez que el usuario *confirma* los datos del simulador, se inicia el **trámite** que da origen a créditos, que tiene cuatro partes que aquí se describen en términos generales luego se detallan diagramas de actividad (ver Diagrama de actividad)
4. Pre-Scoring: en esta etapa, a partir de los datos básicos del crédito (**producto**, plazo, tipo de tasa, monto, etc.) ingresados en el calculador, y los datos básicos del solicitante (ingresos, edad)-que ingresan en esta etapa, se valida el LTV, la relación cuota ingresos, y el límite de edad según el plazo.
5. Scoring: esta etapa tiene dos partes. En la primera *se calcula* la fórmula de scoring para los dos titulares, y en la segunda *se realiza* una consulta a Veraz en línea. Según los resultados, se aprueba o no el trámite.
6. Tasación Electrónica: esta etapa tiene como objetivo *validar* que, según las bases de datos del Banco, el valor de la **propiedad** declarado por el solicitante esté dentro de un rango aceptable que no afecta el LTV del crédito.
7. Finalización del Trámite: en esta etapa, si el usuario lo desea, se piden datos adicionales que figuran en el tríptico de la solicitud para que no tenga que llenarlos a mano, y se pide al usuario que *indique* la forma en la que quiere continuar con el trámite.

Fig. 7. Gestionando Crédito en línea

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

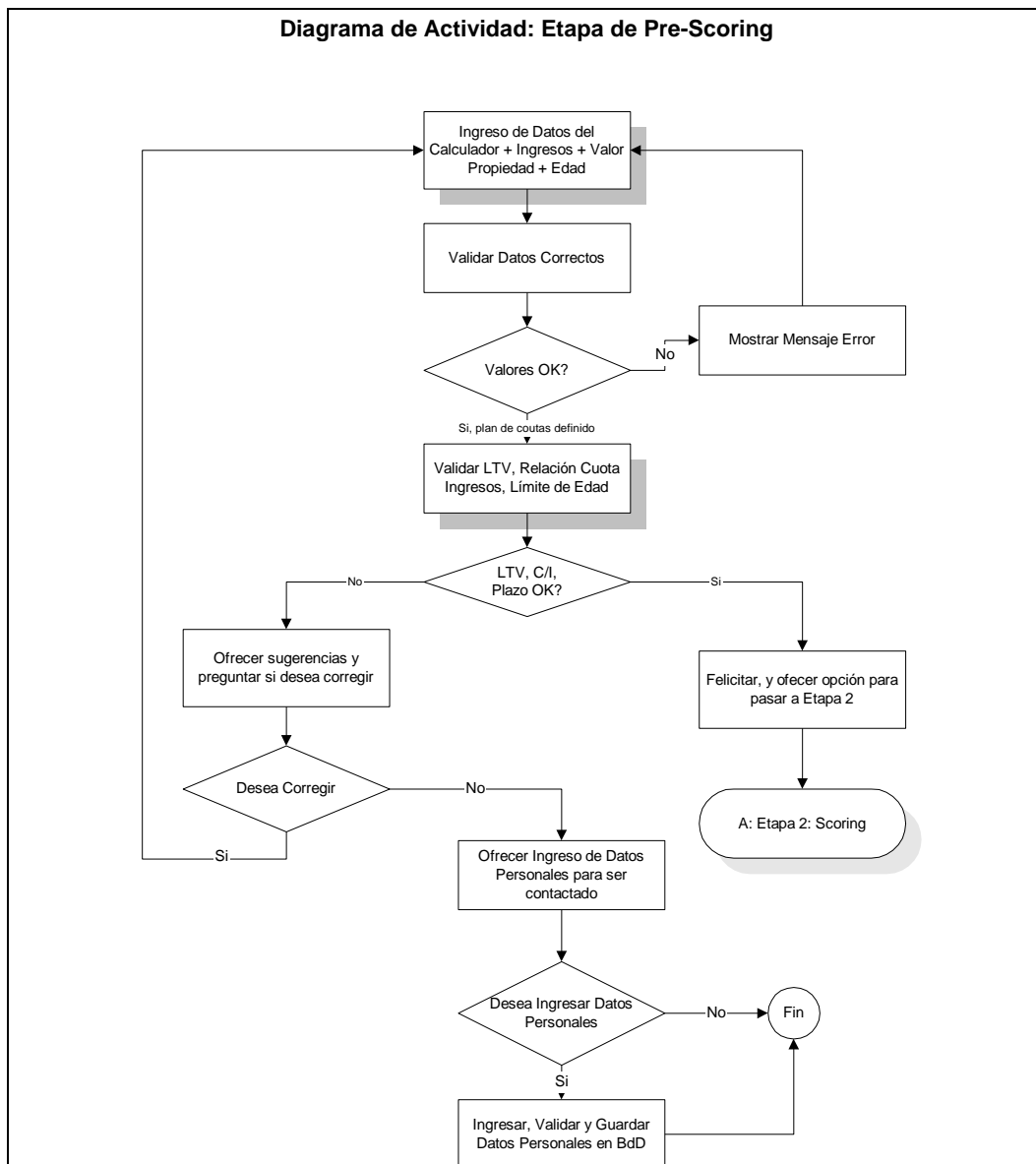


Fig. 8. Diagrama de actividad

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

En la Tabla 13 se detalla el cómputo de las métricas que proponemos, basado en Casos de uso y el Diagrama de actividad.

Tabla 13. Detalle del cómputo usando Casos de uso y Diagrama de actividad

Fuente de identificación	Métricas	Justificación
Caso de uso	T ₁ T ₄ T ₅ T ₆ T ₇	El punto 1 corresponde al acceso al sistema. El punto 2 no se considera, porque el cálculo del plan de cuotas es otro caso de uso. <i>confirmar</i> <i>se calcula</i> <i>se realiza</i> <i>validar</i> <i>indique</i>
	AP ₁ AP ₂	SI EN LA PÁGINA SI EL USUARIO
	EO ₁ EO ₂ EO ₃ EO ₄	Interesado registrado Tramite Producto propiedad
Diagrama de actividad	AP ₃ , AP ₄ , AP ₅ , AP ₆	4 decisiones binarias que no figuran en el caso de uso y se identifican claramente en el diagrama de actividad
	T ₂ T ₃	“Preguntar si desea corregir” y “Preguntar si desea ingresar datos”, implican una respuesta del actor, que es un estímulo de una transacción que no figura en el caso de uso

Por lo tanto, este caso de uso tiene 7T, 4EO y 13P. Se Obtuvieron los P sumando los AP y las T. Las acciones identificadas en el caso de uso no se contaron en el diagrama de actividad. La numeración de las métricas muestra la secuencia temporal con la cual fueron identificadas, en el texto del caso de uso o en el diagrama de actividad.

Se desarrolló este ejemplo con la intención de mostrar que es posible utilizar documentos complementarios para realizar el cómputo de las métricas. Se supone que al usar documentos complementarios, que introducen un mayor nivel de detalle, se reduce el error de la estimación.

4.5. Consistencia con la teoría de medición

En uno de sus libros, Fenton y Pfleeger (1997) describen la “Teoría de la representación de la medición”. Esta teoría formaliza la intuición sobre

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

cómo las medidas son asignadas a cosas. Esto quiere decir que, el dato que se obtiene como medición, debe representar atributos de la entidad observada, y la manipulación del dato debe preservar las relaciones que se observan entre las entidades. De esta forma, nuestra intuición es el punto de partida para todas las mediciones. El primer aspecto a identificar es el atributo de alguna entidad del mundo real: para este trabajo, la entidad seleccionada es la aplicación -definida en la descripción textual de los casos de uso- y el atributo, para el caso de T, es el tamaño de la aplicación. Entonces, es necesario identificar la relación empírica para el atributo tamaño de la aplicación, como se muestra en la Tabla 14, para transacciones. La medición se define como un mapeo del mundo empírico en el mundo relacional formal. Por lo tanto, una medición es un número o símbolo asignado a una entidad en este mapeo, con el objetivo de caracterizar un atributo.

Tabla 14. Relaciones empíricas y numéricas para la métrica de tamaño T

Relaciones empíricas	Relaciones numéricas	Relaciones aplicadas en este trabajo
X es mayor que Y	$X > Y$	$P2 > P1; 63 T > 57 T$
X es mucho mayor que Y	$X > Y + 20 T$	$P2 > P4 + 20; 63 T > 55T$
X es grande	$X > 60 T$	$P2 > 60; 63 T > 60 T$
X + Y es mayor que C	$X + Y > C$	$P4 + P3 > P2; 83 T > 63 T$

De la misma manera, una medición debe especificar un dominio, el rango y las reglas para realizar el mapeo. El dominio es la descripción textual de las aplicaciones desarrolladas con casos de uso; el rango es el conjunto de enteros; las reglas se especificaron en este trabajo cuando se describieron las formas de cálculo de las métricas propuestas.

La condición de representación afirma que el mapeo debe mapear entidades en números, y relaciones empíricas en relaciones numéricas, de tal forma que la relación empírica preservada en la relación numérica. La tabla 14 muestra el mapeo de la relación empírica en la relación numérica, suponiendo que una aplicación es más grande si ésta tiene 20 T más, y que una aplicación es grande si tiene más de 60 T, y que el tamaño en T de los proyectos P1, P2, P3 y P4 es, respectivamente, 57, 63, 48 y 35.

El problema está en el hecho de que es difícil saber con seguridad si el número asignado al tamaño realmente representa el tamaño de la aplicación, dado que es imposible realizar una representación mental del tamaño del software. Para construir una referencia real, podría servir usar la cantidad de hojas o palabras utilizadas en cada una de las descripciones textuales.

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

También la métrica T puede ser caracterizada con una escala ratio porque tiene las siguientes características (Fenton y Pfleeger, 1997):

- a. Es un mapeo de medición que preserva el orden, el tamaño de los intervalos entre entidades, y ratios entre entidades.
- b. Hay un elemento cero, que representa la ausencia total de atributos.
- c. El mapeo de la medición debe comenzar con el cero e incrementarse por intervalos iguales, conocidos como unidades.
- d. Toda aritmética puede ser aplicada con sentido a las clases en el rango del mapeo.

Todas estas consideraciones pueden ser aplicadas también a EO y a P. Por lo tanto, es posible afirmar que tanto T, como EO y P, son métricas consistentes con la teoría de medición.

4.6. Conclusiones del capítulo

A modo de conclusión, se describen las ventajas de las unidades de medidas propuestas. La propuesta de tres unidades de medida persiguió tres objetivos claros: simplificar el método de conteo; definir métricas que sean inherentes a UML; obtener unidades de medida diferentes que, individualmente, puedan capturar un aspecto clave de las aplicaciones de software, para, de esa forma, poder reducir el error de estimación de esfuerzo y, al mismo tiempo, ser consistente con la teoría de la medición.

Es posible afirmar que la simplificación del método de conteo ha sido alcanzada porque:

- a. cada métrica tiene sólo un elemento, lo que evita contar más de una vez una misma cosa
- b. al no utilizar los pesos de complejidad usados en UFP Y UUCP se evita reducir el cálculo realizado en una escala absoluta a una escala ordinal.
- c. Los Factores de Complejidad Técnica y Ambiental no son tenidos en cuenta.

Si bien los casos de uso son propios de UML, se puede afirmar que las unidades de medida propuestas -T, EO y P- son inherentes a UML, puesto que por su naturaleza están unidas a UML. Esta es la razón por la cual no ha sido necesario introducir nuevos conceptos; se han tomado elementos que ya existían en UML o en su contexto. Cabe destacar que al seleccionar los nombres de las métricas, también se ha querido resaltar esta característica, que se considera esencial, dado que va a facilitar la adopción de las métricas propuestas por parte de la industria del software. Al mismo tiempo, como los elementos a ser contados en

DESCRIPCION DE LAS UNIDADES DE MEDIDA PROPUESTAS

las métricas propuestas no son algo exclusivo de UML, podrían también ser identificadas en otros contextos.

Otra condición importante es que cada unidad de medida captura solamente un aspecto de una aplicación: funcionalidad, datos o complejidad. No caben dudas de que estos aspectos deben capturarse separadamente para facilitar el entendimiento intuitivo de lo que expresan.

También es posible contar las métricas propuestas en documentos complementarios, como el que se muestra en el punto 4.4. Se espera que su utilización, en los casos en que es necesario volcar un mayor nivel de detalle que no esté incluido en los casos de uso, optimice aún más el cálculo de la estimación de esfuerzo.

Por las características de las métricas propuestas se puede afirmar que se logró reducir el error de la estimación de esfuerzo. Para comprobarlo, se define un caso de estudio para poder comparar el comportamiento de las métricas propuestas con otras extensamente usadas.

5. Demostración empírica de la disminución del error de la estimación de esfuerzo

Para demostrar empíricamente la disminución del error en la estimación de esfuerzo se utiliza una muestra de 13 proyectos, a la que se aplican tres técnicas diferentes de estimación: productividad media histórica, regresión lineal simple y múltiple, y redes neuronales.

Se comparan los resultados obtenidos con T, EO y P, por un lado, y con UFP y UUCP, por el otro. Se selecciona FP por ser un estándar ampliamente reconocido y UCP por ser una métrica basada en casos de uso. Cabe aclarar que también se analizaron otras métricas basadas en casos de uso (Braz y Virgilio, 2006; Carroll, 2005; Ceddia y Dick, 2004; Chen et al., 2004.; Fetcke et al., 1997; Issa et al., 2006; Kim et al., 2006; Lavazza et al., 2008 ; Levesque et al., 2008), pero los resultados no se pudieron comparar porque esas métricas definen una forma particular de escribir los casos de uso y/o porque basan el cálculo en otros documentos que no son la descripción textual de los casos de uso.

A continuación se describen las características de los proyectos, las técnicas empleadas, los resultados alcanzados con las técnicas empleadas, las amenazas de validez de los resultados alcanzados y las conclusiones de la demostración empírica.

5.1. Características de los proyectos

Los trece proyectos son pequeñas aplicaciones de negocios, que fueron desarrollados en tres contextos diferentes: un ambiente académico de estudiantes avanzados de grado de la Facultad de Ingeniería de la Universidad Austral, en el Departamento de Tecnología y Sistemas (S&T) de la Universidad Austral, y en una empresa de CMM nivel 4. Los recursos humanos involucrados tienen un perfil similar porque fueron estudiantes avanzados los que trabajaron tanto en el Departamento S&T como en la Empresa de CMM nivel 4. Se seleccionaron estos trece proyectos porque los mismos satisfacen los siguientes requisitos:

- a. Las definiciones de requerimientos estaban basadas en casos de uso.
- b. Había información disponible sobre las horas trabajadas.
- c. Eran desarrollos nuevos.
- d. El tamaño de los proyectos había sido calculado con FP, usando el método de Albrecht.
- e. Los casos de uso fueron implementados en forma completa.

Las características de los proyectos se encuentran en la Tabla 15. Además, la Tabla 16 muestra el número de casos de uso (Use Case, UC)

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Tabla 15. Características de los proyectos

Proyecto	Descripción de la aplicación	Ambiente de implementación
P1	Controla viajes usando teléfonos celulares	Java, Tomcat, J2ME Mobile Phone Emulator
P2	Administra un sistema de compras centralizado	Java, Tomcat, Hibernate
P3	Administra el sistema de información de un hotel	Java, Struts, Hibernate
P4	Seguimiento de proyectos	Jave, Hibernate, Tomcat
P5	Administra información de bandas de música	Java, Apache, Tomcat
P6	Vende con una tarjeta autorizada	MS (Visual Studio, Office Sharepoint Portal Server)
P7	Vende entradas de cine usando un teléfono celular	MS Windows XP Professional, Apache, Tomcat
P8	Seguimiento de expedients	JSP, J2EE, Hibernate, Tomcat
P9	Administra un sistema de Activos Fijos	JSP, J2EE, Hibernate
P10	Administra información de alumnos graduados	Power Builder, Javascript
P11	Controla la deuda de clientes	Power Builder
P12	Administra información de viajes	Web application *
P13	Administra información de viajes	Web application *

* Por restricciones de confidencialidad no se muestran más datos

Tabla 16. Datos de los proyectos

Contexto	Proyecto	UC	AE	T	EO	P	UFP	UUCP
Académico	P1	40	410.00	57	18	71	144	201
	P2	26	473.50	63	21	73	266	149
	P3	15	382.40	48	22	60	171	84
	P4	10	285.00	35	8	49	142	72
	P5	12	328.00	21	5	34	89	72
	P6	8	198.00	23	11	35	75	62
	P7	6	442.02	33	10	50	57	71
Departamento S&T	P8	27	722.65	79	17	97	210	175
	P9	15	392.00	56	19	83	311	111
	P10	19	272.00	39	16	42	155	119
	P11	13	131.00	18	9	18	117	68
Empresa de	P12	20	1042.00	72	13	118	130	169

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

CMM nivel 4	P13	12	348.00	25	6	32	93	71
-------------	-----	----	--------	----	---	----	----	----

Esfuerzo Real (Actual Effort, AE), las T, P, UFP y UUCP de cada proyecto. El AE es medido en horas hombre (Man Hours, MH).

P1 a P7 fueron desarrollados en un contexto académico, P8 a P11 en el Departamento S&T, y P12 y P13 en una Empresa de CMM nivel 4. P5 a P7 fueron desarrollados al mismo tiempo, a partir de agosto de 2007, una vez que fueron concluidos los otros diez proyectos.

Si bien se puede observar cierta dispersión en las características de los proyectos, los mismos no son totalmente heterogéneos. Son proyectos reales de negocios pequeños. Como sucede en todo proceso de estimación basado en datos históricos, a mayor similitud entre los datos históricos y los estimados, menor será el error de la estimación. Si los proyectos a usar como información histórica y aquellos a estimar tuvieran características externas muy distintas -tales como experiencia del grupo de desarrollo, herramientas utilizadas no equiparables, nivel de compromiso, etc.- el error de la estimación estaría aumentado por este motivo.

5.2. Productividad media histórica

La técnica que calcula la media de la productividad histórica (Historical Mean Productivity, HMP) se seleccionó debido a su facilidad de uso y a su comprensión intuitiva. Se definieron dos escenarios para obtener resultados con los cuales se pudiera verificar si el error de estimación había disminuido. En el primer escenario, se utilizó un criterio cronológico, es decir, los 10 primeros proyectos fueron empleados como información histórica y los últimos tres, como proyectos a ser estimados. Se construyó el segundo escenario para obtener una mayor evidencia:

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

todas las combinaciones posibles de los 13 proyectos, tomados en grupos de 3. Luego se validaron los resultados con el coeficiente de correlación de Spearman.

Se calculó la productividad (Pr) de cada proyecto como el cociente entre el Tamaño del proyecto y el AE, para T, EO, UFP y UUCP. Así mismo, se calculó la HMP como la media aritmética de la Pr de los proyectos históricos -para esta técnica, siempre hubieron 10 proyectos que fueron considerados históricos-. En cuanto a P, se definió la Pr como el cociente entre P y AE. *Aunque P sea una medida de complejidad, se consideró que este cálculo es razonable porque P captura la complejidad de las reglas de negocios, y por lo tanto, si el AE es un valor fijo, y los P aumentan, entonces la Pr también aumentará.*

En las fórmulas que se describen a continuación (16 a 20), i es el conjunto de proyectos históricos, j son los proyectos a ser estimados, y la n es el número de proyectos históricos, por lo tanto:

$$Pr i_{[T, EO, UFP, UUCP]} = \text{Tamaño } i_{[T, EO, UFP, UUCP]} / AE i_{[MH]} \quad (16)$$

$$Pr i_{[P]} = \text{Complejidad } i_{[P]} / AE I_{[MH]} \quad (17)$$

$$HMP_{[T, EO, UFP, UUCP, P]} = \sum Pr i_{[T, EO, UFP, UUCP, P]} / n \quad (18)$$

Se calculó el esfuerzo estimado para cada uno de los 3 proyectos en todos los grupos como el cociente entre Tamaño o Complejidad de cada proyecto y de HMP. Se realizó esto quince veces: una por cada proyecto y por cada métrica (T, EO, P, UFP y UUCP), aplicando las siguientes fórmulas (19 y 20):

$$EE j_{[T, EO, UFP, UUCP]} = \text{Size } j_{[T, EO, UFP, UUCP]} / HMP_{[T, EO, UFP, UUCP]} \quad (19)$$

$$EE j_{[P]} = \text{Complexity } j_{[P]} / HMP_{[P]} \quad (20)$$

Otros autores usaron el factor de productividad para estimar el esfuerzo en una manera diferente, lo cual demuestra que aún no ha sido adoptada una forma estándar. Se resumen a continuación dichas formas.

- a. Karner (Karner, 1993) definió un número constante de horas-hombre por UCP, sin tener en cuenta procesos de desarrollo diferentes o las distintas características de los proyectos.
- b. Anda (Anda, 2002), Anda (Anda et al. 2001, 2002), Mohagheghi et al. (Mohagheghi et al., 2005) usaron el número de Karner de horas-hombre por UCP corregido considerando la cantidad de transacciones para calcular el esfuerzo estimado.
- c. Braz y Vergilio (Braz y Virgilio, 2006) calcularon la productividad como el cociente entre el Tamaño y el Esfuerzo, y luego lo usaron para estimar el esfuerzo.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

d. Caroll (Caroll, 2005) aplicó UCP en un contexto industrial complejo. Para estimar el esfuerzo, usó la productividad histórica, añadió un coeficiente de riesgo para cubrir los factores de riesgo que intrínsecamente no son incorporados al modelo, y estimó la creación de informes en horas-hombre, en vez de UCP.

Además, Anda (Anda et al., 2005) divulgó que había una correlación fuerte (Correlación de Pearson = 0.729, el P-valor = 0.000) entre el esfuerzo aplicado en la realización de los casos de uso y el número de transacciones de cada uno de ellos. *Este resultado demuestra que las transacciones pueden ser una buena unidad de medida para ser usada en el cálculo del tamaño de una aplicación.*

5.2.1. Primer escenario

Se construyó este escenario con los 10 proyectos que fueron desarrollados antes que los proyectos P5, P6 y P7, más estos tres últimos proyectos, que fueron desarrollados posteriormente. Se utilizaron los 10 primeros proyectos mencionados como datos históricos, con la intención de imitar una situación real, y se definió que se estimaría el esfuerzo de los tres restantes. El Box-plot de la Figura 9 muestra los resultados obtenidos para los tres proyectos estimados usando al resto como datos históricos. La Tabla 17 muestra la media de Z, la desviación estándar, y el error cuadrático medio. Se aplicó el coeficiente de correlación no paramétrico Spearman a las métricas y AE, al conjunto de 13 proyectos, con el objetivo de validar los resultados. También se aplicó el coeficiente de correlación de Pearson para T (Pearson correlation = 0.796, P-Value = 0.01) para comparar resultados con los de trabajos realizados previamente.

Como el error de la estimación de esfuerzo puede ser afectado por las características de los proyectos seleccionados y el conjunto de proyectos históricos, se construyó otro escenario para analizar el impacto de estos aspectos.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

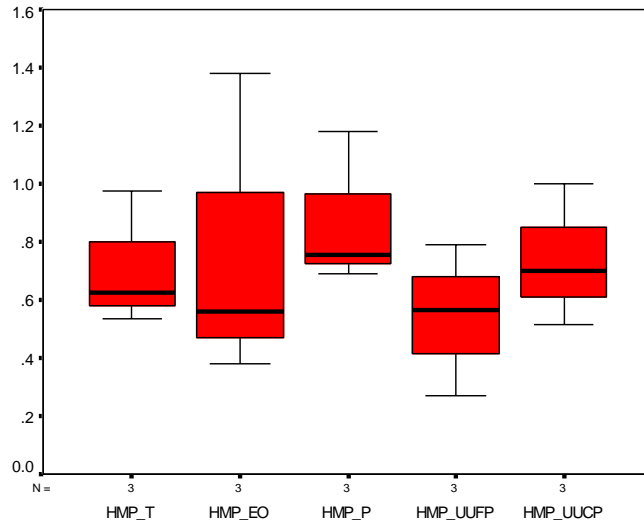


Fig. 9. Box-plot de valores de Z obtenidos para 3 proyectos estimados, usando otros 10 como información histórica (1^{er} escenario)

Tabla 17. Valores obtenidos para 3 proyectos estimados, usando otros 10 como información histórica (1^{er} escenario)

Métricas	Media de Z	Desviación Std. Z	Error Cuadrático Medio	Coefficiente Correlación Spearman (Métricas & AE)
T	0.7113	0.2307	129.92	0.824**
EO	0.7726	0.5325	168.14	0.451 (Sig.0.122)
P	0.8750	0.2652	88.07	0.857**
UFP	0.5413	0.2612	205.34	0.352 (Sig.0.239)
UUCP	0.7394	0.2461	136.26	0.697**

** La correlación es significativa en el nivel 0.01

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

5.2.2. Segundo Escenario

Se formó un nuevo conjunto con todas las combinaciones posibles de los 13 proyectos, tomados de a 3, lo que resultó en 286 combinaciones. Se utilizaron los 10 proyectos restantes en cada combinación como datos históricos. De esta forma, se estimó el error para cada uno de los 3 proyectos, en cada una de las 286 combinaciones, lo que dio como resultado 858 valores. Una vez que se obtuvieron las estimaciones individuales, se calculó la media de Z y la desviación estándar para el conjunto de 858 valores. Se realizó este proceso cinco veces, una vez para cada métrica. La figura 10 muestra el Box-plot de los 858 casos. Además, la Tabla 18 muestra la media de Z, la desviación estándar, el error cuadrático medio y los valores de coeficiente de correlación de Spearman.

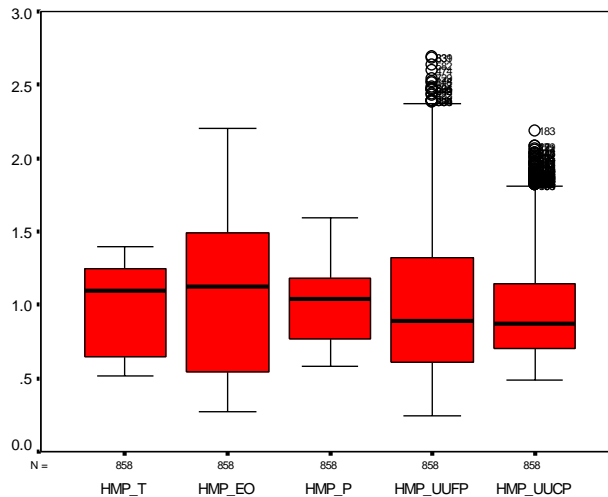


Fig. 10. Box-plot de valores de Z para cada uno de los 3 proyectos en las 286 combinaciones (2do escenario)

Tabla 18. Valores obtenidos para cada uno de los 3 proyectos en las 286 combinaciones (2do escenario)

Métricas	Media de Z	Desviación Std. Z	Error Cuadrático Medio	Coefficiente Correlación Spearman (Métricas & AE)

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

T	1.0075	0.2855	148.92	0.824**
EO	1.0255	0.5328	257.26	0.451 (Sig.0.122)
P	1.0055	0.2476	111.26	0.857**
UFP	1.0307	0.5933	278.06	0.352 (Sig. 0.239)
UUCP	1.0169	0.4386	191.09	0.697**

* * La correlación es significativa en el nivel 0.01

Para analizar la precisión de las métricas propuestas, se compararon T, EO y P, con UFP Y UCCP. Para ello, se planteó una Tabla con 5 columnas, y 858 filas. Se calculó para cada fila, y para cada métrica, la función lógica "es más pequeño que el error obtenido con". En la Tabla 19 y 20 se encuentran los resultados de la comparación de cada una de las métricas propuestas con UFP o UUCP.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Tabla 19. Comparación de los valores de Z obtenidos con las métricas propuestas y UFP en el 2do escenario

Error calculado con	Es menor que el error obtenido con	Cantidad de veces que es cierto, expresado en porcentaje
UFP	T, EO, y P	12.5%
T	UFP	72.8%
EO	UFP	45.2%
P	UFP	75.7%

Tabla 20. Comparación de los valores de Z obtenidos con las métricas propuestas y UUCP en el 2do escenario

Error calculado con	Es menor que el error obtenido con	Cantidad de veces que es cierto, expresado en porcentaje
UUCP	T, EO y P	30.7%
T	UUCP	63.6%
EO	UUCP	19.6%
P	UUCP	60.6%

5.2.3 Análisis de los resultados

En ambos escenarios, el mejor resultado se obtuvo con P, puesto que presenta la media y el coeficiente de correlación más cercanos a 1 y el menor error cuadrático medio, como muestran los box-plots de las Figuras 9 y 10, y la Tabla 17 y 18. Si bien P no obtuvo en el primer escenario el menor valor de la desviación estándar, sí lo hizo en el segundo escenario. Es importante destacar que para este trabajo es más significativo el error cuadrático medio que la desviación estándar, puesto que el primero calcula las diferencias entre AE y EE y el segundo entre Z y la media de Z. Este resultado fue validado con el coeficiente de correlación de Spearman.

Además, en el primer escenario, los resultados obtenidos con UFP mostraron los valores más alejados de 1 para la media y el coeficiente de correlación, y el valor más grande en el error cuadrático medio. Esto fue validado con el resultado obtenido con el coeficiente de correlación, lo cual indica que no hay correlación entre UFP Y AE.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Por otro lado, en la Tabla 17, no hay ninguna diferencia significativa entre los resultados obtenidos con T y UUCP, aunque los resultados del coeficiente de correlación de Spearman indicaran que T pudo haber obtenido un mejor resultado que UUCP. En cuanto a EO, tuvo una media similar a T y UUCP, pero al mismo tiempo, se destacó por la mayor desviación estándar. Este último valor junto con el valor de error cuadrático medio y el coeficiente de correlación indican que EO podría tener un comportamiento similar a UFP. Tal es así que se pudieron confirmar estas suposiciones cuando se hicieron los cálculos de todas las combinaciones posibles, como lo muestra la Tabla 18. Es importante tener en cuenta que el primer escenario es sólo una de las 286 combinaciones posibles del segundo escenario.

Al analizar el segundo escenario, se observó que las cinco métricas pueden ser consideradas como estimadores posibles, ya que las medias tienen valores cercanos a 1 (cfr. Tabla 18). Las diferencias relevantes no fueron encontradas en el análisis de las medias de Z, pero sí en la desviación estándar y en el error cuadrático medio, como lo muestra la Tabla 18, donde claramente se ve que P tiene los mejores valores.

Como se supuso por el análisis de la Tabla 17, en la Tabla 18 EO tiene valores más cercanos a UFP. Esto puede deberse al alto contenido de datos que tiene la métrica FP.

Los análisis de resultados arriba detallados muestran información global sobre los datos. Sin embargo, fue interesante analizar cada uno de los 858 casos para confrontar el comportamiento de las métricas propuestas y el de las seleccionadas. La Tabla 19 muestra que UFP obtuvo el error más pequeño sólo en el 12.5 % de los casos, y la Tabla 20 muestra que UUCP lo hizo en el 30.7 % de los casos. Estos valores bajos confirman las limitaciones planteadas en 2.2.1. Se cree que las diferencias existentes entre UFP y UUCP se deben a que el último método usó una menor cantidad de elementos y a que la complejidad de cada elemento fue medida aplicando un criterio diferente al de UFP, ya que no fue calculada contando principalmente los elementos de datos. También se cree que UUCP obtuvo resultados más cercanos a T y P debido a que la forma en que las transacciones fueron identificadas fue la misma que la usada para T y P. Además, cabe mencionar que los *valores obtenidos con EO la posicionan como una unidad de medida mejor que UFP pero, inferior a UUCP.*

Por otra parte, se cree que T y P obtuvieron mejores resultados, entre el 60 % y el 76 %, porque tienen una relación lineal más fuerte con AE y, probablemente, *estas unidades de medida capturen mejor la realidad.*

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Además, el valor obtenido aplicando el coeficiente de Correlación de Pearson entre T y AE, (Pearson la correlación = 0.796, el P-valor = 0.001) fue similar al obtenido por Anda et al. (cfr. 5.2), lo que corrobora los resultados obtenidos con T. Las pequeñas diferencias con los valores reportados por Anda pueden ser debidas a los criterios utilizados para identificar las transacciones.

Se puede afirmar que, si se comparan las unidades de medida T y P con UFP y UUCP, usando la técnica HMP, se logró:

- a. reducir el error de estimación para el juego de datos descripto.
- b. reducir el rango de los errores, aumentando la confiabilidad de las métricas. Por el contrario, en la figura 10 se puede apreciar que UFP y UUCP tienen una cantidad importantes de outliers, lo cual significa que con estas últimas unidades de medida existe una mayor probabilidad de obtener un error mayor a lo esperado.
- c. aumentar la precisión de T y P.

5.3. Reregresión lineal

El atractivo del método de regresión lineal (simple y múltiple) se encuentra en la posibilidad de calcular el valor de una variable dependiente, a partir de la función de una recta, que describe la relación lineal entre esta variable dependiente y la o las variables independientes. Por lo tanto, seleccionaron los métodos de regresión lineal⁹ simple y múltiple, puesto que es posible usarlos para estimar esfuerzo. Estos métodos explican la variación de la variable dependiente al variar una o más de las variables independientes, debido a su relación lineal. Se busca en ambos casos el mejor valor de los parámetros, reduciendo al mínimo la suma de los cuadrados de los errores.

5.3.1. Simple

Al igual que en el primer escenario de la técnica HMP, se utilizaron los 10 proyectos desarrollados en una etapa anterior a P5, P6 y P7 como información histórica. Se usaron estos valores históricos para calcular las rectas de regresión lineal -una para cada métrica- como se indica en la fórmula (21):

$$AE = a_j + \beta_j V_j. \quad (21)$$

Donde AE es la variable dependiente, a y β son los coeficientes de la recta y V_j es la variable independiente, siendo j cada una de las métricas (T, EO, P, UFP y UUCP). Se calcularon dichos coeficientes aplicando el

⁹ Se usó para este cálculo el software SPSS para Windows versión 11.5.1, método introducir

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

método de regresión lineal, usando los valores de la Tabla 16 para los proyectos P1 a P4 y P8 a P13.

5.3.2. Múltiple

Por otro lado, se definieron dos nuevos modelos con la finalidad de estudiar si era posible aplicar el método de regresión lineal múltiple a un modelo que integrara las tres métricas propuestas, como se muestra en las fórmulas 22 y 23:

$$AE = a_1 + \beta_2 T + \kappa_3 EO + u_4 P. \quad (22)$$

$$AE = a_5 + \beta_6 T + \kappa_7 EO + u_8 AP. \quad (23)$$

Donde AE es la variable dependiente, α , β , κ y u son los coeficientes de la recta y T, EO, P y AP son las variables independientes. Se calcularon dichos coeficientes aplicando el método de regresión lineal, usando los valores de la Tabla 21 para los proyectos P1 a P4 y P8 a P13.

La Tabla 21 muestra los valores de R, R^2 para cada uno de los modelos. Como se puede observar, no hay una relación lineal entre las variables independientes EO, UFP y AE. Por este motivo, no se utilizaron estas variables para estimar el esfuerzo. Además, se observó que se obtenían los mismos valores R, R^2 y R^2 Ajustada, para los modelos de regresión lineal múltiple T-EO-P y T-EO-AP, pero diferentes valores del Factor de Inflación de la Varianza (FIV). Analizando estos datos se llegó a la conclusión de que el primer modelo tiene problemas de multicolinealidad, lo que significa que los valores de los coeficientes de regresión son muy sensibles a cualquier cambio de la muestra. Por el contrario, los FIV del segundo modelo evidenciaron que el segundo modelo es más estable. Esto se debe a que AP es la variable que realmente introduce nueva información en el modelo, debido a que P es igual a la suma de T más AP.

Tabla 21. Valores de los modelos de regresión lineal para los proyectos históricos

Análisis Estadístico	T	P	UUCP	T-EO-P	T-EO-AP	EO	UFP
Anova Sig.	0.004	0.000	0.042	0.002	0.002	0.601	0.804
R	0.815	0.905	0.649	0.946	0.946	0.189	0.09
R^2	0.664	0.819	0.421	0.895	0.895	0.036	0.008
R^2 Ajustado				0.842	0.842		
Colinealidad FIV				16.33 2.17 12.73	3.89 2.17 2.41		

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Fue posible estimar el valor del esfuerzo al definir la recta de estimación con los valores de los coeficientes calculados con la información histórica, según se describen en las fórmulas 24 y 25:

$$EE_{ij} = a_j + \beta_j V_{ij} . \quad (24)$$

$$EE_i = a_5 + \beta_6 T_i + \kappa_7 EO_i + u_8 AP_i . \quad (25)$$

Donde EE_{ij} son los valores estimados de esfuerzo para cada uno de los i proyectos –P5, P6 y P7- y para cada una de las j métricas – T, P y UUCP-, a_j β_j son los coeficientes de las j rectas calculadas con valores históricos y V_{ij} son los valores de T, P y UUCP, para cada uno de los i proyectos a estimar –P5, P6 y P7- de la Tabla 16. Además, los EE_i son los valores estimados de esfuerzo para cada uno de los i proyectos –P5, P6 y P7-; a_5 , β_6 , κ_7 y u_8 son los coeficientes de la recta planteada en la fórmula 25 y las T_i , EO_i y AP_i son los valores de la Tabla 16, para cada uno de los i proyectos a estimar.

Una vez calculados los valores de los EE y los valores de AE, fue posible calcular los valores de Z, dado que Z es el cociente entre EE y AE. En el Box-plot de la figura 11, se muestran los Z de cada uno de los proyectos –P5, P6 y P7-, calculados para cada uno de los modelos de regresión lineal simple y múltiple. Además, en la Tabla 22 se muestran la media y la desviación estándar de Z y el error cuadrático medio.

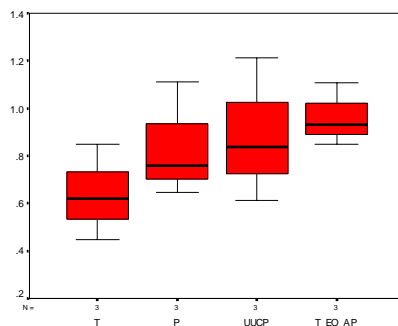


Figura 11. Box-plot de los Z de los proyectos a estimar

Tabla 22. Resultados del análisis de los errores de los proyectos estimados

Variables independientes	Media Z	Desviación Std. Z	Error Cuadrático Medio
T	0.6381	0.2009	143.83
P	0.8396	0.2422	91.56
UUCP	0.8871	0.3027	106.32
T-EO-AP	0.9628	0.1331	42.66

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

5.3.3. Análisis de los resultados

Se obtuvo el mejor valor para el modelo de regresión lineal múltiple, donde la variable dependiente es Esfuerzo y las independientes son T, EO y AP. En la Fig. 11 se observa que la distancia entre los valores máximo y mínimo es menor para este caso, y además, que los valores son los más cercanos a 1, lo que indica que los valores de los errores son menores. En la Tabla 22 también se observa que este caso tiene la media de Z más cercana a 1 y el menor valor para desviación estándar y el error cuadrático medio.

Se pudo obtener este resultado porque dicho modelo explica mejor la forma en que la variación de cada una de las variables independientes afecta a la variación del esfuerzo, obteniendo un modelo que integra más información sobre la realidad de una aplicación. Es importante tener presente que T mide la funcionalidad, EO los datos de la aplicación y AP la complejidad, entonces este modelo explica, para este conjunto de datos, cómo varía el esfuerzo al variar la funcionalidad, los datos y la complejidad.

Además, es interesante analizar los coeficientes de regresión estandarizados¹⁰, cuyos valores –en valor absoluto– para T, EO y AP son 0.784, 0.357 y 0.378, respectivamente, puesto que proporcionan información sobre la importancia relativa de cada variable independiente en la ecuación de regresión. De ellas se puede decir que la que más peso tiene en el modelo es T, luego AP y por último, en un valor cercano, es EO. Es muy interesante conocer esta composición de la relación, puesto que estos coeficientes están caracterizando, para el conjunto de datos dado, la composición de funcionalidad, complejidad y datos, y el efecto que producen en el esfuerzo.

De los modelos de regresión simple se puede decir que, si bien la media de Z más cercana a 1 es la de UUCP, P tiene una media similar, un menor valor de desviación estándar y del error cuadrático medio (cfr. Tabla 22). Por lo que se deduce que P, para este conjunto de datos, es una mejor variable predictora que UUCP, puesto que al ser muy pequeña la diferencia en la media, pesa más la evaluación de la dispersión de los errores.

Con el propósito de comparar estos resultados con los obtenidos en el segundo escenario de la técnica HMP, se calcularon para la totalidad de los datos de los 13 proyectos, los coeficientes de correlación de Pearson (T=0.795, Sig.=0.01 bilateral; P=0.897, Sig.=0.01 bilateral; UUCP=

¹⁰ Estos coeficientes están basados en valores estándares, que se producirán en la variable dependiente por cada cambio de una unidad en la correspondiente variable independiente, manteniendo constante el resto de las variables independientes.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

0.657, Sig.=0,05 bilateral). Si se comparan estos coeficientes con los calculados anteriormente con el coeficiente de Spearman (cfr. Tabla 17), se observan que son valores similares. Del análisis de estos datos se puede entrever que si se considera el conjunto de los 13 proyectos, se espera que P sea un mejor estimador que T y UUCP, y a su vez, que T sea mejor que UUCP. Al mismo tiempo, se debe resaltar que el valor de los coeficientes de Pearson para los 13 proyectos es muy similar al valor obtenido para los datos históricos.

Por otro lado, cabe destacar que el modelo de regresión lineal asume los siguientes supuestos¹¹ :

- a. Normalidad. Para cada valor de la variable independiente, o combinación de valores de las variables independientes, los residuos se distribuyen normalmente con media cero.
- b. Independencia. Los residuos son independientes entre sí, es decir, que constituyen una variable aleatoria.
- c. Homocedasticidad. Para cada valor de la variable independiente, o combinación de valores de las variables independientes, la varianza de los residuos es constante.
- d. Linealidad. Lo que implica que la variable dependiente es la suma de un conjunto de elementos: el origen de la recta, una combinación lineal de variables independientes o predictoras y los residuos
- e. No-Colinealidad. No existe relación lineal exacta entre ninguna de las variables independientes.

En el Apéndice III se desarrolla un análisis detallado de estos supuestos. Las conclusiones de este análisis son las siguientes:

- a. Ninguno de los modelos cumple el supuesto de Normalidad.
- b. T no cumple el supuesto de independencia
- c. No hay evidencia, en ninguno de los casos, para rechazar el supuesto de homocedasticidad ni de linealidad, ni no-colinealidad en el modelo de regresión múltiple.

En cuanto a la existencia de casos atípicos, lo que requeriría perfeccionar el modelo, no se identificaron. Si bien existen residuos más grandes para el modelo de regresión lineal simple con la variable independiente T, este número corresponde al proyecto que tiene el mayor número de horas trabajadas de la muestra, por lo tanto, al tener objetivamente un residuo más grande, aunque no en forma proporcional a las horas trabajadas (si se lo compara con otros casos) no se considera que sea necesario

¹¹ <http://www2.uca.es/serv/ai/formacion/spss/Inicio.pdf>

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

descartarlo. Además, resulta de interés para el análisis de las métricas contar con un dato, que a pesar de provenir de un proyecto chico, muestra un mayor valor en las horas trabajadas.

Dado que no se cumplen todos los supuestos que garantizarían la validez del modelo, los resultados del análisis deben ser interpretados con cautela. Se puede afirmar que bosquejan una interesante posibilidad para estimar Esfuerzo en función de T o P, y T-EO-AP, pero es importante encontrar caminos alternativos para validarla, ya sea replicando el caso de estudio en condiciones similares o agregando más datos.

El valor de estos resultados también está basado en el hecho de que no existe una muestra de datos más grande, donde se apliquen las métricas propuestas. Por lo tanto, la natural dificultad para armar un conjunto de datos similar al presentado u obtener más datos lleva a apreciar todo esfuerzo de análisis.

Como otros autores (Kitchenham et al., 2001) piensan que los residuos para estimaciones de esfuerzo no cumplen con el supuesto de normalidad, se plantea la necesidad de trabajar con modelos no-paramétricos.

5.4. Redes neuronales

Debido a que con el método de regresión lineal el supuesto de normalidad no pudo ser comprobado, se decidió analizar modelos no-paramétricos. El primer inconveniente que surgió fue la cantidad de datos, lo que hacía imposible la aplicación de estos métodos. Por lo tanto, se trabajó con un nuevo conjunto de datos (cfr. Apéndice IV), compuesto por una tabla que sintetiza datos a nivel de proyecto y otra que registra los datos por caso de uso. Lamentablemente, este detalle de información sólo estaba disponible para los proyectos P1 a P11.

La Tabla 1 del Apéndice IV contiene las variables AE, T, EO, P y UFP a nivel de proyecto y la Tabla 2 las mismas variables citadas anteriormente, pero esta vez registradas a nivel de caso de uso. De este grupo de variables cabe destacar el esfuerzo que supone el registro de horas reales trabajadas por caso de uso. Además, la Tabla 2 contiene un conjunto de variables que caracterizan a los proyectos, que se describen a continuación:

- a. *Web*. Se le asigna el valor 0 si no es un proyecto web, 1 si es web y 2 si parte del proyecto es web.
- b. *Tipo*. Se le asigna el valor 0 si es académico, 1 si pertenece al Departamento de T&S de la Universidad Austral, 2 si es un proyecto de la empresa nivel 4 de CMM.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

c. *Clasificación.* Se le asigna el valor 0 si AE es menor a 2000 y 1 si AE es mayor a 4000.

Se planteó un trabajo final de la carrera de grado de Ingeniería Informática de la Universidad Austral (Sagardía y Ale, 2008), donde se seleccionó y aplicó la técnica más adecuada para calcular la estimación de esfuerzo. Un modelo de Data Mining Multi-relacional fue el elegido, puesto que la información está concentrada en las dos Tablas descriptas en el Apéndice IV. La primera Tabla sintetiza información a nivel de proyecto, y la segunda, información de cada caso de uso. Como, a su vez, un proyecto se describe como un conjunto de casos de uso, esta relación fue la que motivó el uso del modelo multi-relacional, que plantea una dependencia “una a muchas” entre las tablas. Además, se buscó la forma de comparar las métricas propuestas con el estándar UFP. También este modelo facilitó la inclusión de UFP, expresado solamente a nivel de proyecto.

Se utilizó la herramienta SQL Server en su versión 2008. Si bien el conjunto de datos tiene un tamaño considerable (181 filas por 11 columnas), es más bien pequeño para ser procesado con técnicas de Data Mining. Sin embargo, a pesar de esta limitación, fue posible usar la técnica *redes neuronales*.

Las redes neuronales son un modelo no-paramétrico. Son un paradigma de aprendizaje y procesamiento automático, inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida.

Esta red (Sagardía y Ale, 2008) consiste en neuronas organizadas en tres capas: capa de entrada, capa oculta y capa de salida. Las neuronas en la capa de entrada se corresponden con los atributos de entrada y las neuronas en la salida se corresponden con los atributos objetivos. Las neuronas ocultas están conectadas simultáneamente con las neuronas de entrada y las neuronas de salida, y son esenciales para la clasificación. La mayoría de los algoritmos de redes neuronales se basan en la unidad llamada *perceptrón*. El perceptrón calcula una combinación lineal de sus entradas, y produce un resultado entre dos valores posibles. La función de activación devuelve la suma de los pesos de las entradas como valor de la salida (sólo existen dos valores de salida posibles).

El algoritmo de red neuronal combina cada posible estado del atributo de entrada con cada posible estado del atributo de predicción, y usa los datos de entrenamiento para calcular las probabilidades. Posteriormente, puede usar estas probabilidades para la clasificación o la regresión, así

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

como para predecir un resultado del atributo de predicción, basándose en los atributos de entrada.

5.4.1. Definición de los modelos

Los autores (Sagardía y Ale, 2008), eligieron el 70% de los datos para el entrenamiento del modelo -8 proyectos- y el 30% de ellos para las pruebas -3 proyectos-. Los motivos para ello fueron que 8 fue el número menor de proyectos que soportó la herramienta para obtener modelos significativos y que se consideró una distribución de datos razonable. La herramienta realizó una selección de 3 proyectos al azar, a saber: P1, P5 y P7.

Los modelos a ser generados con la técnica redes neuronales fueron comparados con aquellos obtenidos con la técnica HMP, para el mismo conjunto de datos, usando para el entrenamiento del modelo los mismos proyectos que fueron usados como información histórica y como proyectos a estimar los mismos que se utilizaron en las pruebas.

Se plantearon tres preguntas de investigación:

- a. ¿La técnica de redes neuronales, aplicada a un modelo multi-relacional, mejora el error de la estimación de esfuerzo, si se lo compara con la técnica HMP, para las métricas propuestas T, EO y P?
- b. ¿Es más significativa la incidencia de las métricas propuestas T, EO y P, en la mejora del error de la estimación de esfuerzo, que la incidencia de UFP, usando la técnica de redes neuronales, aplicada a un modelo multi-relacional?
- c. ¿Se justifica el esfuerzo de registrar las horas trabajadas por casos de uso?

En forma conjunta con los autores (Sagardía y Ale, 2008), definieron los 16 modelos que están descritos en la Figura 12. Con cada uno se intentó resolver una cuestión particular. Además, la Tabla 23 (Sagardía y Ale, 2008) muestra las variables usadas en cada modelo.

En la Tabla 24 (Sagardía y Ale, 2008), se detalla la media, la desviación estándar y el error cuadrático medio obtenido para cada modelo, usando la técnica redes neuronales. Además, en la Tabla 25 se detallan la media, la desviación estándar y el error cuadrático medio, usando la técnica HMP, para cada una de las métricas propuestas y regresión lineal múltiple para el conjunto de las tres métricas.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

- M1. Identificación de la mejor técnica.**
M2. Incidencia de UFP en el modelo, a nivel de proyecto y de caso de uso.
M3. Uso de T como unidad de medida, a nivel de proyecto y de caso de uso.
M4. Uso de T como unidad de medida, a nivel de proyecto.
M5. Uso de P como unidad de medida, a nivel de proyecto y de caso de uso.
M6. Uso de P como unidad de medida, a nivel de proyecto.
M7. Uso de EO como unidad de medida, a nivel de proyecto y de caso de uso.
M8. Uso de EO como unidad de medida, a nivel de proyecto.
M9. Uso de AP como unidad de medida, a nivel de proyecto y de caso de uso.
M10. Uso de AP como unidad de medida, a nivel de proyecto.
M11. Uso de T, EO, AP como unidades de medida, a nivel de proyecto y de caso de uso.
M12. Uso de T, EO, AP como unidades de medida, a nivel de proyecto.
M13. Peso de la Tabla de casos de uso.
M14. Variaciones a nivel de proyecto en el mejor modelo (M11): sin T
M15. Variaciones a nivel de proyecto en el mejor modelo (M11): sin T y agregando UFP.
M16. Variaciones a nivel de proyecto en el mejor mo(M11): sin T y AP.

Fig. 12. Modelos definidos para la técnica redes neuronales

Tabla 23. Variables usadas en cada modelo con la técnica redes neuronales.

Modelo	Nivel de Proyecto					Nivel de Caso de uso							
	EO	UFP	T	P	AP	EO	AP	T	P	Tipo	Web	Clasif	AE
M1	X	X	X	X	X	X	X	X	X	X	X	X	X
M2	X		X	X	X	X	X	X	X	X	X	X	X
M3			X					X		X	X	X	X
M4			X							X	X	X	X
M5				X					X	X	X	X	X
M6				X						X	X	X	X
M7	X					X				X	X	X	X
M8	X									X	X	X	X
M9					X		X			X	X	X	X
M10					X					X	X	X	X
M11	X		X		X	X	X	X		X	X	X	X
M12	X		X		X					X	X	X	X
M13	X	X	X	X	X								
M14	X				X	X	X	X		X	X	X	X
M15	X	X				X	X	X		X	X	X	X
M16	X					X	X	X		X	X	X	X

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

Tabla 24. Valores calculados para los modelos usando la técnica redes neuronales

Modelo	Z Media	Z Desviación Standard	Error Cuadrático Medio
M1	0.862	0.082	69.719
M2	0.869	0.085	67.98
M3	0.858	0.097	74.657
M4	0.858	0.099	74.72
M5	0.865	0.1	72.734
M6	0.865	0.102	73.042
M7	0.86	0.1	73.851
M8	0.875	0.114	72.789
M9	0.879	0.111	70.532
M10	0.877	0.112	71.345
M11	0.893	0.118	67.748
M12	0.886	0.118	70.027
M13	0.655	0.221	158.836
M14	0.88	0.111	70.301
M15	0.866	0.113	75.24
M16	0.871	0.114	74.031

Tabla 25. Valores calculados para las métricas usando la técnica HMP y regresión lineal múltiple

Métrica	Z media	Z Desviación Estándar	Error cuadrático medio
T	0.504	0.072	196.06
P	0.621	0.087	152.34
EO	0.448	0.122	214.49
UFP	0.351	0.130	269.86
T-EO-AP	0.755	0.316	136.49

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

5.4.2. Análisis de los resultados

Para identificar la mejor técnica se incluyeron en el M1 la Tabla 1 –nivel de proyecto- y la Tabla 2 –nivel de casos de uso- del Apéndice IV, y la totalidad de las variables de dichas tablas. Si se comparan los rangos de valores para la media, la desviación estándar y el error cuadrático medio de las técnicas redes neuronales y HMP de la Tabla 26, se observa que, al ser los rangos de valores menores para redes neuronales, esta técnica evidencia ser mejor que la HMP. Este resultado es razonable, puesto que HMP sólo trabaja con medias históricas, mientras que las redes neuronales utilizan algoritmos mucho más complejos, que les permiten procesar una mayor cantidad de información.

Tabla 26. Comparación entre las técnicas redes neuronales y HMP-Regresión lineal múltiple

Técnica	Z media	Z Desviación Estándar	Error cuadrático medio
Redes Neuronales	[0.655- 0.893]	[0.082- 0.221]	[67.748 – 158.836]
HMP y Regresión lineal múltiple	[0.351 – 0.755]	[0.072- 0.316]	[136.49 – 269.86]

En el modelo M2, se extrajo la variable UFP de M1. La variación en los resultados de M1 y M2 fue mínima, con una pequeña mejora del error en M2. Se obtuvo un resultado similar en M14, al introducir UFP y sacar AP de M11. También se vio en M16, que si se saca UFP de M15, dejando a nivel de proyecto sólo a EO y UFP, se pueden mejorar los resultados. Por lo tanto, se concluye que *la utilización de los UFP no produce una mejora significativa para el cálculo de la estimación de esfuerzo con la técnica redes neuronales.*

Al analizar los aportes individuales de las unidades de medida T, EO, P y AP a nivel de proyecto y de caso de uso (M3, M5, M7, M9), se comprueba que se obtuvo el mejor valor con AP, aunque ninguno logró mejorar los valores de M2. Además, al extraer las variables T, EO, P y AP a nivel de caso de uso en M4, M6, M8, M10, respectivamente, fue mínima la variación que se registró, si se la compara, con M3, M5, M7, M9, respectivamente. Por ejemplo, el error cuadrático medio de M3 es 74.657 y el de M4 es 74.72. Por lo tanto, se deduce que *es importante para la optimización de la estimación la información que brindan las métricas en su conjunto, y no la que brinda cada una de ellas en forma aislada.*

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

El mejor valor de estimación de la técnica redes neuronales se obtuvo en M11, con la combinación de las métricas T, EO y AP con los valores registrados a nivel de proyecto y de caso de uso. En M12, la extracción de estas variables a nivel de caso de uso del M11 no mejoró el modelo, y la diferencia negativa fue significativa. Por lo que se deduce que es importante el aporte del detalle a nivel de caso de uso para el conjunto de las métricas T, EO y AP. Al mismo tiempo, este resultado muestra que la métrica P no mejora la estimación del error, puesto que P es igual a la suma de T más AP.

En M13 se eliminó la tabla 2 del Apéndice IV del modelo M1, por lo cual se estimó usando sólo la Tabla a nivel de proyecto. En este modelo se obtuvieron los peores resultados, es decir, los errores más grandes, evidenciados por la media más alejada de 1, y los mayores valores de la desviación estándar y del error cuadrático medio de la Tabla 24. De ello se deduce que *el detalle a nivel de caso de uso mejora la estimación de esfuerzo.*

A modo de conclusión, se responden las preguntas de investigación planteadas anteriormente:

- a. La técnica de redes neuronales, aplicada a un modelo multi-relacional, redujo el error en la estimación de esfuerzo, si se lo compara con el obtenido mediante la técnica HMP-Regresión lineal múltiple, para las métricas T, EO y AP, y el conjunto de datos analizados, en un valor del 51% para el menor valor de los errores cuadráticos medios y en un 42%, para el mayor valor (cfr. Tabla 26).
- b. Fue significativa la incidencia de las métricas T, EO y AP en la mejora del error en la estimación de esfuerzo, usando la técnica de redes neuronales, aplicada a un modelo multi-relacional, en comparación con la incidencia de UFP, puesto que la presencia de UFP no mejora significativamente el error de la estimación.
- c. Se justifica el esfuerzo de registrar las horas trabajadas por casos de uso, en el conjunto de datos analizado, puesto que se logra reducir en un 58% el error cuadrático medio (cfr. M13 y M11 de la Tabla 24).

5.5. Descripción del contexto empírico

Podría haber algunas circunstancias que afecten la validez de los datos recopilados y, por lo tanto, que puedan modificar el alcance de las conclusiones. Por este motivo, para enmarcar la demostración empírica, se describen a continuación algunas de esas circunstancias y la forma en que fueron afrontadas.

I. En el proceso de desarrollo:

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

- a. *La posibilidad de que el proceso de desarrollo no hubiese estado definido.* Todos los proyectos estaban basados en un proceso de desarrollo definido, excepto aquellos desarrollados en el Departamento S&T. De hecho, el proceso de desarrollo usado para los proyectos académicos era una personalización del Proceso Unificado, y, como es esperable, la empresa CMM nivel 4 trabaja habitualmente sobre un proceso bien definido.
- b. *La posibilidad de que las tareas no hubiesen estado claramente definidas y / o terminadas.* Sólo fueron tenidas en cuenta las tareas de programación que habían sido completadas.
- c. *La posibilidad de que las horas trabajadas hubiesen sido calculadas de diferentes maneras.* Para encontrar una equivalencia en la forma de registro de las horas trabajadas en programación en los proyectos desarrollados en el contexto académico y en el Departamento S&T, y en la empresa CMM nivel 4, las horas trabajadas en tareas de Programación, Diseño, Diseño GUI y Análisis fueron calculadas como horas trabajadas en Programación. Esta equivalencia fue confirmada por un experto de la empresa CMM nivel 4, quien explicó que respondía a la forma con la cual trabaja la empresa: es muy difícil para ellos diferenciar las horas de Programación, Diseño, Diseño GUI y Análisis.

II. En el proceso de recopilación de datos:

- a. *La posibilidad de que la descripción de los casos de uso estuviese incompleta.* Casi todos los casos de uso fueron completamente escritos, excepto algunos casos de uso en P9. Por consiguiente, se estimaron las métricas propuestas para esos casos de uso.
- c. *La posibilidad de que las horas trabajadas hubiesen sido erróneamente registradas.* En los proyectos académicos, se solicitó a los estudiantes dividir las tareas de cada proyecto en término de casos de uso, lo que facilitó el registro de horas trabajadas. Por consiguiente, cada semana, las horas trabajadas reportadas por los estudiantes fueron registradas y, con regularidad, validadas. Las horas trabajadas en la empresa CMM nivel 4 fueron facilitadas por la misma, obteniéndolas de los registros de datos de la empresa. En el Departamento S&T, se usó una planilla con las descripciones de las tareas y el tiempo consumido en cada una. Sólo las horas trabajadas en P10 fueron calculadas de la información obtenida de la documentación del proyecto y las descripciones de las tareas realizadas por el programador que desarrolló el proyecto.
- d. *La posibilidad de que el conteo de las métricas propuestas fuese inexacto o realizado bajo la influencia del conocimiento previo del AE.*

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

La autora de este trabajo realizó personalmente el conteo dos veces y de una manera independiente de los valores del AE; no se obtuvo como fruto del análisis del AE, sino a partir de la descripción textual de los casos de uso.

- e. *La posibilidad de que la cantidad de datos fuese insuficiente.* De acuerdo con la cantidad y tipo de proyectos con los que se trabajó, fue posible obtener el alcance de las conclusiones que se lograron. Por lo tanto, la cantidad de proyectos es suficiente para sostener las conclusiones arribadas. Evidentemente, un mayor número y variedad de proyectos podría mejorar el alcance de las conclusiones, pero dada la dificultad para obtener el registro de horas trabajadas de proyectos de software en el ámbito industrial, se consideró necesario publicar estas conclusiones, para favorecer la futura recopilación de datos y su posterior análisis. Además, para obtener información suficiente como para modificar el alcance de las conclusiones, el esfuerzo de recopilación de datos podría haber dilatado en años la presentación de este trabajo, con la consiguiente pérdida de interés y actualidad.

5.6. Conclusiones de la demostración empírica

Se puede afirmar que el conjunto de 13 pequeños proyectos de negocios es variado, si se consideran aspectos tales como su origen –académico o de la industria del software-; las distintas tecnologías de última generación utilizadas para desarrollarlos; y los dominios de las aplicaciones desarrolladas –telefonía celular, proyectos que gestionan información, compras, hoteles, bandas de música, graduados, deuda de clientes, activos fijos, viajes, control de proyectos, ventas con tarjeta y seguimiento de expedientes-. Además, el rango de AE es [131-1042] horas trabajadas. Por lo que se concluye que, si bien es un conjunto de datos pequeños, es aceptablemente representativo de las aplicaciones de negocios.

Se utilizaron 4 técnicas distintas: HMP, regresión lineal simple y múltiple, y redes neuronales. La primera es simple, y se empleó para comparar cada una de las métricas propuestas en forma individual. La segunda y tercera son medianamente complejas –paramétricas-, lo que posibilita la comparación de las métricas en forma individual y conjunta, respectivamente. La cuarta -no-paramétrica-, que tiene un cierto grado de complejidad, es un modelo de varias variables. Por las características enunciadas, la tercera y la cuarta permitieron definir modelos que incluyen las 3 métricas propuestas: T, EO y AP. Es importante destacar que se han obtenido valores convergentes en los resultados alcanzados con las distintas técnicas.

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

En cuanto al comportamiento individual de cada métrica, se comprobó que, para el conjunto de datos estudiado, utilizando la técnica HMP, se pudo reducir el error en la estimación usando las métricas T para los valores de la Tabla 18 -46% y 22%- y P -59% y 41%-, si se comparan con UFP o UUCP, respectivamente. Además, P fue la métrica con la cual se obtuvieron los mejores resultados para las técnicas HMP y regresión lineal simple, siendo estos valores similares (cfr. error cuadrático medio Tabla 17 y Tabla 22).

La disminución del error en la estimación con EO se podría resumir diciendo que se verificó una disminución pequeña si se lo compara con UFP -7%- y un aumento del error -34%- si se lo compara con UUCP para la técnica HMP (cfr. Tabla 18, error cuadrático medio). En el caso de regresión lineal simple, al igual que UFP, no se observa una relación lineal.

En cuanto al comportamiento de las métricas propuestas en su conjunto, se llegó a resultados equiparables, tanto usando las técnicas de regresión lineal múltiple, como las redes neuronales, puesto que con ambas técnicas se pudo obtener el mejor resultado usando T, EO y AP. Además, entre el M13 -modelo que sólo tiene la Tabla de datos a nivel de proyecto- y el modelo de regresión múltiple, hay una pequeña diferencia, que probablemente se pueda atribuir a la presencia de UFP en M13 (cfr. Tablas 24 y 25).

Por cierto, sin olvidar la debilidad del modelo de regresión debido al no cumplimiento de la totalidad de los supuestos, no se debe despreciar la riqueza de la convergencia de resultados, que se resaltó en los párrafos anteriores.

El hecho de que P, en forma individual, posibilite una mayor disminución del error que T, es natural porque P es igual a T más AP. O sea, que tiene un mayor grado de información que T. Además, es lógico que en los modelos de variables múltiples, P sea reemplazada por AP, puesto que la diferencia entre T y P es, precisamente, AP.

Para las métricas propuestas T, EO y AP, y el conjunto de datos analizados, la técnica redes neuronales, aplicada a un modelo multi-relacional, redujo el error en la estimación de esfuerzo, si se comparan los resultados obtenidos con las técnicas HMP y regresión lineal múltiple, en un valor del 51% para el menor valor de los errores cuadráticos medios y en un 42%, para el de mayor valor (cfr. Tabla 26).

También, fue significativa la incidencia de las métricas propuestas T, EO y AP en la mejora del error de la estimación de esfuerzo, usando la técnica de redes neuronales, aplicada a un modelo multi-relacional, si se

DEMOSTRACION EMPIRICA DE LA DISMINUCION DEL ERROR DE LA ESTIMACION DE ESFUERZO

la compara con la incidencia de UFP, puesto que la presencia de UFP no mejoró significativamente el error de la estimación.

Finalmente, se puede afirmar que se justificó el esfuerzo de registrar las horas trabajadas por casos de uso, en el conjunto de datos analizado, puesto que se logró reducir el error en un valor del 58% del error cuadrático medio (cfr. M13 y M11 de la Tabla 24).

6. Diseño e implementación de una herramienta para el seguimiento y control de productos de software

En este capítulo se describe la génesis de una herramienta que usa las métricas de tamaño propuestas como base para el cálculo del avance de la construcción de un producto.

Se describe, en primer lugar, un meta-diseño, es decir, un diseño que no está orientado a un ambiente de implementación en particular. En la descripción de la idea y los requerimientos funcionales del meta-diseño se utilizaron la plantilla del informe de visión y la de casos de uso del Rational Unified Process. Dichos documentos fueron la base del diseño y programación de una herramienta (Sistema de Seguimiento y Control de Productos, SSCP), que fuera realizada en el contexto de un Trabajo final de la carrera de grado de la Facultad de Informática de la UNLP (Romero y Pons, 2008).

En este capítulo se presentan el informe de Visión, la descripción textual de los casos de uso, una breve descripción del sistema SSCP y, a modo de conclusión del capítulo, las lecciones aprendidas en la construcción de la herramienta.

6.1 Informe de Visión

El siguiente documento describe los requerimientos del producto a diseñar (MetaDesign), desde la perspectiva de los usuarios.

6.1.1. Posicionamiento

El primer aspecto a tener en cuenta en el posicionamiento del producto es la oportunidad de negocio. En la actualidad existe una gran variedad de herramientas que miden en forma automática métricas de producto, *pero suelen estar orientadas a determinados ambientes de desarrollo*. Además, la información sobre el avance de la construcción de un producto que necesitan los diferentes interesados (Gerentes, Líderes y Desarrolladores) *suele estar disponible en un conjunto, no siempre integrado, de herramientas*. Estas dos premisas son la base para identificar: la Sentencia que define el problema, que se muestra en la Tabla 27, y la Sentencia que define la posición del Producto, que se muestra en la Tabla 28.

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Tabla 27. Descripción de la Sentencia que define el problema

El problema de	Cálculo automático de métricas de producto y del control del avance de la construcción de un producto de software
afecta a	Gerentes, líderes de proyecto y desarrolladores
El impacto asociado es	No contar con la información necesaria para poder realizar las correcciones en el momento oportuno.
una adecuada solución sería	Una herramienta que sea multiplataforma, altamente compatible con diferentes ambientes, que se inserte en el ámbito de desarrollo habitual, sin la necesidad de incorporar tecnología nueva, que brinde una propuesta para el control automático de la construcción de un producto y un conjunto –no acotado- de métricas de diseño orientado a objetos

Tabla 28. Descripción de la sentencia que define la posición del Producto

Para	Desarrolladores, Empresas que desarrollen software, fábricas de software
Quienes	Necesiten implementar una herramienta que facilite el control del avance de la construcción de un producto de software y que además requieran información sobre el diseño de un producto orientado a objetos, y que tengan un ambiente integrado de modelización y de producción automática de código, basado en UML.
El (nombre del producto)	Meta Design
Que	Define un meta diseño, a ser implementado en diferentes plataformas
Nuestro producto	Brinda la posibilidad de un control permanente del avance de la construcción del producto, integrando las etapas de definición de requerimientos y construcción, por medio de una propuesta de un conjunto de métricas particular, pero no limitante. Además, logra un conjunto de características que, si bien existen en otros productos, no se encontró uno que cumpla con la totalidad de estas características. Por lo tanto, logra, con una sola herramienta, satisfacer las necesidades de información sobre el avance de la construcción de un producto del Gerente, del

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

	Líder de proyecto y de los desarrolladores.
No como	CostXPert 3.3 ¹² , MetriFlame v1.3 ¹³ , QMOOD++ ¹⁴ , SDMetrics ¹⁵ , Enterprise Architect ¹⁶

6.1.2 Descripción de Usuarios

En la Tabla 29 se muestra una descripción de los usuarios. El entorno del usuario es múltiple plataforma, y debe operar con herramientas desarrolladas en diferentes lenguajes. El usuario trabaja con herramientas de desarrollo de ambientes basados en UML y con generación de código y de archivos en XML.

Tabla 29. Descripción de los Usuarios

Nombre	Descripción
Gerente de Desarrollo de Producto	Controla el avance de la construcción de uno o más productos. Controla el trabajo de líderes y desarrolladores.
Líder de Proyecto	Controla el avance de la construcción de uno o más productos. Controla el trabajo de desarrolladores.
Desarrollador	Realiza la construcción de uno o más productos. Controla el avance de su trabajo.

6.1.3 Descripción global del producto

Se espera que el diseño de la herramienta sea utilizado para la implementación de herramientas fácilmente integrables con diferentes ambientes de desarrollos. El mismo es un meta diseño, esto quiere decir que debe ser extendido a los diferentes ambientes de desarrollo. Los datos registrados por esta herramienta sirven para controlar el avance de la construcción de un producto de software y pueden ser utilizados como base para la estimación de los costos de construcción.

Se identifican las siguientes suposiciones y dependencias con respecto a la herramienta:

¹² www.costxpert.com, herramienta evaluada en su versión 3.2 por Aurer-Graser-Biffi.

¹³ <http://virtual.vtt.fi/metriflame/>, herramienta evaluada por Aurer-Graser-Biffi

¹⁴

<http://www.ddj.com/showArticle.jhtml;jsessionid=DZOURWOWR3WBWQSNLRSKHSCJUNN2JVN?articleID=184410338>

¹⁵ <http://www.sdmetrics.com/CustomXMI.html#CustomXMI>

¹⁶ <http://www.sparxsystems.com.au/ea.htm>

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

a. Usuario

- El usuario ya ha realizado la definición de requerimientos con una descripción textual de los casos de uso. Además, sabe calcular la cantidad de T y EO por casos de uso.
- Se parte de la premisa de que las herramientas con las que tiene que operar usan UML, generan código y archivos en XML.
- El usuario en su ambiente de desarrollo trabaja habitualmente con casos de usos, diagramas de secuencia, diagramas de clase, y código generado en forma automática.
- Se presupone la sincronización y traceabilidad entre los casos de uso, diagrama de secuencia, diagrama de clases, generación automática de clases.

b. Diseño

- Se va a usar UML 2, XML, XMI, DI, MOF.

Las necesidades del sistema, junto con las prioridades definidas, se describen en la Tabla 30.

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Tabla 30. Necesidades del sistema

Necesidad	Prioridad	Características
Base de Datos Relacional	1	Se utilizará un Marco de Persistencia que asegure la independencia de la herramienta de la Base de Datos. Se busca que la persona que use la herramienta seleccione la Base de Datos y que pueda usar, además, el grupo de herramientas de consultas utiliza habitualmente.
Registro de avance de construcción del producto	1	El avance de construcción del producto va a ser medido con dos métricas EO y T.
Cálculo y registro de Métricas de producto	1	Registra y calcula las métricas definidas en el trabajo de Tesis de maestría "Métrica de diseño orientadas a objetos", de Gabriela Robiolo
La definición de las métricas está abierta	2	Siguiendo el marco definido en el trabajo de Tesis de maestría "Métrica de diseño orientadas a objetos", de Gabriela Robiolo, se podrán agregar nuevas métricas de producto OO.
Registro de las horas trabajadas por el desarrollador	2	El registro debe ser muy simple, de tal modo que no le resulte un esfuerzo extra al desarrollador. Al mismo tiempo, la herramienta debe facilitarle ese registro por medio de alarmas o registrando las horas que estuvo activa la herramienta.
Diseño de reportes de información a extraer de la base de Datos	3	Se sugerirán un conjunto de posibles reportes a obtener de la Base de datos, para los diferentes usuarios.

6.1.4. Otros Requerimientos del Producto

Los *estándares* a aplicar son:

- a. MOF 2.0/XMI Mapping Specification¹⁷, v2.1 (September 2005)
- b. Diagram Interchange¹⁸ 1.0 (April 2006)

¹⁷ www.omg.org/docs/formal/05-09-01.pdf

¹⁸ www.omg.org/docs/formal/06-04-04.pdf

c. Meta Object Facility (MOF)¹⁹ 2.0 Core proposal

d. UML 2.0²⁰

La interfase del producto con las herramientas de desarrollo va a estar definida por archivos XML, que siguen la norma XMI. El producto debe tener una *GUI totalmente intuitiva*, de modo tal que la capacitación del usuario sea la mínima imprescindible. *No hay que perder de vista que las métricas no son un fin en sí mismas sino que son una ayuda para el trabajo del desarrollador. Si le resultan complejas, no las va a usar, puesto que le dará prioridad al trabajo de desarrollo.*

Se trabajará en un *entorno de desarrollo orientado a objetos*.

6.2. Casos de uso del Meta-Diseño

A continuación se describen los casos de uso que definen la funcionalidad del meta-diseño.

6.2.1 Actualizar información desde los Diagramas de secuencia

Actor: Desarrollador

Se considera que un producto/proyecto es un entregable, una unidad que puede ser parte de otro producto o proyecto. Es una parte que podrá ser identificada con un nombre y del cual se obtendrá un código ejecutable. En este capítulo se lo llamará Producto.

La herramienta es el ambiente de desarrollo del desarrollador, quien genera los archivos XML y el código ejecutable.

Este caso de uso se ejecutará para actualizar la Base de datos, con respecto a un producto determinado, y para cada uno de sus casos de uso, con los siguientes datos: la cantidad de T que tiene incorporadas, las re-hechas, y las finalizadas, y para cada módulo, la cantidad de EO incorporados, los re-hechos, y los finalizados.

Los casos de uso tienen la siguiente información sobre T:

- a. Cantidad de T pactadas: son las T que han sido estipuladas en la definición contractual.
- b. Cantidad de T incorporadas: son las T que superan el número de lo pactado, y se obtienen como la diferencia entre las T calculadas a partir de los diagramas de secuencia y las T pactadas.
- c. Cantidad de T no finalizadas: son las T que corresponden a un caso de uso no finalizado.

¹⁹ www.omg.org/docs/ad/03-04-07.pdf

²⁰ www.omg.org/docs/formal/05-07-04.pdf

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

- d. Cantidad de T re-hechas: son las T que corresponden a un caso de uso que ha sido re-hecho.
- e. Cantidad de T finalizadas: son las T que corresponden a un caso de uso finalizado

Los módulos (conjunto de casos de uso) tienen la siguiente información sobre transacciones:

- a. Cantidad de EO pactados: son los EO que han sido estipulados en la definición contractual.
- b. Cantidad de EO incorporados: son los EO que superan el número de lo pactado, y se obtienen como la diferencia entre los EO calculados a partir de los diagramas de secuencia y los EO pactadas.
- c. Cantidad de EO no finalizados: un módulo se considera finalizado cuando todas las clases que interactúan en los diagramas de secuencias que conforman un módulo han sido finalizadas. Por lo tanto, los EO no finalizados son los diferentes EO del módulo que no tienen todas las clases finalizadas.
- d. Cantidad de EO re-hechos: son los EO que corresponden a un módulo que ha sido re-hecho. Al módulo se lo considera re-hecho si alguna de sus clases pasa al estado re-hecho.
- e. Cantidad de EO finalizados: un módulo se considera finalizado cuando todas las clases que interactúan en los diagramas de secuencias que conforman un módulo han sido finalizadas.

Flujo de Eventos

El sistema muestra los productos que ha dado de alta y los nombres de los diagramas de secuencia de cada uno de esos productos. También permite seleccionar los diagramas sobre los cuales se desea trabajar. De cada diagrama, se obtendrá la siguiente información:

- a. Las clases que intervienen en el diagrama de secuencia.
- b. Las T de cada diagrama de secuencia. El total de T del caso de uso se calcula contando los estímulos que el actor envía al sistema.
- c. Los EO de cada módulo. El total EO es igual a la suma de los *diferentes* EO del módulo. Esto significa que aunque el objeto de entidad esté repetido en el módulo, éste se cuenta una sola vez.

El producto no tiene casos de uso o se quiere dar de alta a algún caso de uso nuevo. El sistema tomará todos los archivos XML que corresponden a los diagramas de secuencia y los dará de alta, y les completará la información anteriormente descrita para cada caso de uso. Se aconseja

mantener el mismo nombre en la herramienta para los casos de uso y para los diagramas de secuencia.

El archivo XML del Diagrama de secuencia no ha sido generado. El sistema le muestra un mensaje de error.

Pre-condiciones: se ha generado un archivo XML del diagrama de secuencia.

Pos-condiciones: se actualizan en la Base de Datos los siguientes valores:

- a. Las clases que definen un caso de uso.
- b. La cantidad de T que tiene cada caso de uso, diferenciando las incorporadas, las no finalizadas, las re-hechas y las finalizadas.
- c. La cantidad de EO que tiene cada caso de uso, diferenciando los incorporados, los no finalizados, los re-hechos y los finalizados.

6.2.2 Actualizar información del Diagrama de Clases

Actor: Desarrollador

Este caso de uso se ejecutará para actualizar el estado de las clases del Producto en la base de datos.

Flujo de Eventos

El sistema muestra los productos que han sido dados de alta, permite seleccionar los productos y/o el diagrama que será utilizado para actualizar la información.

El sistema actualizará los estados de las clases e identificará las clases que son EO. El estado, por defecto, es no finalizado.

Aclaración: Cada vez que una clase -tanto en el diseño como en el código- sea modificada, deberá cambiar de estado. El cambio de estado es realizado en forma automática por la herramienta. El único cambio de estado que debe realizarse en forma manual es cuando se define como finalizado. En la Tabla 31 se describen los estados de una clase y para cada uno de ellos el estado siguiente.

Tabla 31. Estados de una clase

Estado	<i>Próximo estado</i>
Por defecto	No finalizado
No finalizado	Finalizado
Finalizado	Re-hecho
<i>Re-hecho</i>	<i>Finalizado</i>

El archivo XML del Diagrama de clases no ha sido generado. El sistema le muestra un mensaje de error.

Pre-condiciones: se ha generado un archivo XML del diagrama de clases y se han dado de alta los casos de uso con sus respectivas clases.

Pos-condiciones: se actualiza el estado de las clases del Producto en la Base de Datos y se identifica aquellas clases que son objetos de identidad.

6.2.3. Calcular métricas de Producto

Actor: Desarrollador

Este caso de uso se ejecutará para actualizar las métricas de Producto en la base de datos. El archivo sobre las que calculará es un archivo XML que se obtiene a partir del diagrama de clases de cada producto.

Flujo de Eventos

El sistema muestra aquellos productos que tiene datos de alta y las posibles métricas que puede calcular. Permite seleccionar el o los productos y las métricas que va a calcular. Se calcula cada métrica de cada producto que ha sido seleccionado y el resultado se guarda en la base de datos.

El sistema tendrá la opción de consultar las métricas calculadas.

El Diagrama de clases no ha sido generado. El sistema muestra un mensaje de error.

Pre-condiciones: se ha generado un archivo XML del diagrama de clases.

Pos-condiciones: se actualiza la Base de Datos con valores de cada una de las métricas seleccionadas.

6.2.4. Ingresar horas trabajadas

Actor: Desarrollador

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Cada desarrollador ingresa las horas trabajadas en forma periódica. Cada período puede ser de un día, o cada vez que cierre y abra la herramienta. Es importante facilitarle al desarrollador la carga de horas trabajadas.

Flujo de Eventos

El sistema muestra los productos que tiene dados de alta por el desarrollador, la última carga de horas trabajadas realizada para cada producto y le permite realizar para una fecha dada, una nueva carga de horas trabajadas en horas y minutos, por producto o por caso de uso por producto.

El sistema permite activar la carga de horas trabajadas por caso de uso. Por defecto, se oculta esta opción.

Pre-condiciones: Se ha dado de alta el Producto.

Pos-condiciones: Se actualiza la Base de Datos con los valores de las horas trabajadas para cada Producto, por cada desarrollador y por cada caso de uso.

6.2.5 Modificar y Consultar horas trabajadas

Actor: Desarrollador - Líder - Gerente

Sólo el Desarrollador puede modificar sus horas trabajadas.

Cada uno de estos actores tiene una visión más amplia de la consulta que el anterior. Un gerente tiene líderes y desarrolladores a cargo. Un Líder tiene desarrolladores a cargo. Un desarrollador tiene productos a cargo. Como tienen el mismo formato, a cada uno se le permite seleccionar todos los productos a los que tiene acceso.

Los datos son tomados de la base de Datos.

Flujo de Eventos

Selecciona si quiere ver el detalle de horas trabajadas por producto o por desarrollador. El sistema muestra los productos y los desarrolladores que tienen dados de alta, el actor selecciona las entidades de interés y se muestra el resultado de la consulta.

Si se han cargado las horas trabajadas por caso de uso, se podrá ver el detalle de las horas trabajadas por caso de uso.

El desarrollador puede modificar los valores de horas trabajadas.

6.2.6 Consultar avance de construcción de Producto

Actor: Desarrollador - Líder - Gerente

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Cada uno de estos actores tiene una visión más amplia de la consulta que el anterior. Como tienen el mismo formato, a cada uno se le permite seleccionar todos los productos a los que tiene acceso.

Los datos son tomados de la base de Datos.

Flujo de Eventos

El sistema muestra los productos que tiene datos de alta. Selecciona los productos a consultar.

a. Medido en T. En la Tabla 32 se describe el contenido de la consulta y en las fórmulas 26, 27 y 28 las respectivas fórmulas de cálculo.

Porcentaje de avance = T finalizadas / (T pactadas + T incorporadas) (26)
por Caso de uso

Porcentaje de avance = Total T finalizadas / (Total T pactadas + Total T incorporadas) (27)
del producto

Productividad = Total T finalizadas / Total Horas trabajadas (28)

Si se registran las horas trabajadas por caso de uso (lo que es opcional), es posible calcular la productividad por caso de uso.

Tabla 32. Descripción de la consulta de avance de construcción de un producto basado en T

Producto	Caso de uso	Cantidad de T pactadas	Cantidad de T incorporadas	Cantidad de T no finalizadas	Cantidad de T re-hechas	Cantidad de T finalizadas	% de avance
Total Producto							
% de Avance del Producto							

b. Medido en EO

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

En la Tabla 33 se describe el contenido de la consulta. Los cálculos se realizan en forma similar al punto anterior.

Tabla 33. Descripción de la consulta de avance de construcción de un producto basado en T

Producto	Módulo	Cantidad de EO pactados	Cantidad de EO incorporados	Cantidad de EO no finalizados	Cantidad de EO re- hechos	Cantidad de EO finalizados	%de avance
Total Producto							
% de Avance del Producto							

6.2.7. Consultar Métricas de Producto

Actor: Desarrollador - Líder - Gerente

Cada uno de estos actores tiene una visión más amplia de la consulta que el anterior. Como tienen el mismo formato, a cada uno se le permite seleccionar todos los productos a los que tiene acceso.

Los datos son tomados de la base de Datos.

Flujo de Eventos

- a. El sistema muestra los productos que tiene datos de alta. Selecciona los productos a consultar.
- b. Se muestran, por producto, los ítems detallados en el cuadro. Puede verse el detalle de cada métrica calculada.
- c. Se muestra en la Tabla 34 la salida propuesta para un producto determinado.

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Tabla 34. Descripción de la consulta de métricas de producto

Característica	Métrica	Valor Métrica
Tamaño	Cantidad de clases desarrolladas	
	Cantidad de Interfases desarrolladas	
	Porcentaje de cantidad de Interfases desarrolladas sobre cantidad de clases	
Reutilización	Cantidad de Clases externas especializadas	
	Cantidad de interfases externas extendidas	
	Cantidad de colaboradores externos	
	Cantidad de clases que implementan interfases externas	
	Porcentaje de reutilización sobre el total de clases desarrolladas	
Herencia	Cantidad de jerarquías de clases desarrolladas	
	Rango de niveles de especialización por jerarquía de clases desarrolladas	
	Porcentaje de jerarquías sobre clases desarrolladas	
	Rango de porcentaje de métodos reemplazados en una jerarquía	
	Cantidad de clases raíz no abstractas	
	Cantidad de clases raíz no abstractas que implementan Interfases	
	Cantidad de jerarquías extendidas de clases externas	
	Rango de niveles agregados a jerarquías donde la raíz es externa	
	Rango de porcentaje de métodos reemplazados en jerarquías donde la raíz es externa	
	Cantidad de jerarquías de interfases desarrolladas	
	Rango de niveles de especialización por jerarquías de interfases desarrolladas	
	Rango de métodos de interfase	
Porcentaje de jerarquías de interfases sobre cantidad de		

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

	interfases	
Colaboración – Agregación	Rango de colaboradores por clase	
	Mediana de colaboradores por clase	
	Porcentaje de clases que no tienen colaboradores.	
Responsabilidades públicas	Mediana de métodos de interfase por clase	
	Rango de métodos de interfase por clase	
	Total de responsabilidades públicas	

6.2.8 Ingresar el Tamaño Pactado de un Producto y las identificaciones de los módulos

Actor: Desarrollador / Administrador

Flujo de Eventos

El sistema muestra los productos que tiene cargados.

El actor selecciona los productos y se muestran los casos de uso con las T/EO pactadas. Permite modificar el valor de las T/EO pactadas y agregar a cada caso de uso el módulo que le corresponde.

Pre-condiciones: Existen los productos con sus casos de uso en la Base de Datos.

Pos-condiciones: Se modifica el valor de las de las T/EO pactadas.

6.2.9 Editar Usuario

Actor: Administrador

Flujo de Eventos

El sistema muestra la lista de usuarios del sistema, los perfiles, con sus permisos relativos. Además permite dar de alta un perfil, si es necesario, o modificar los permisos.

El sistema permite dar de alta, modificar o remover un usuario.

Si quiere crear un nuevo usuario y el usuario ya existe, el sistema le solicitará que ingrese otro nombre.

6.2.10. Editar Métricas de Producto

Actor: Administrador

El sistema implementará un sub-conjunto del conjunto sugerido de métricas en (Robiolo, 2004), pero no restringirá la implementación del resto.

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

El sistema muestra la lista de métricas de producto ya ingresadas al sistema.

El sistema permite dar de alta una métrica nueva. La fórmula de cálculo de las métricas se escribirá en SQL, para facilitar la extensión y modificación de las mismas.

También se podrán dar de baja, modificar o remover las métricas. Se describe en la Tabla 35 el contenido de la pantalla de edición.

Tabla 35. Pantalla Editar Métricas de Producto

Nombre de la Métrica	Descripción de la forma de cálculo	Forma de cálculo

6.2.11. Editar Producto

Actor: Desarrollador - Administrador

Flujo de Eventos

El sistema muestra la lista de productos del sistema. El sistema permite dar de alta, baja, modificar o remover un producto.

Si el producto ya existe, se solicitará que ingrese otro nombre.

Si el producto tiene datos asociados a él (Clases, Casos de uso, Horas trabajadas, métricas) no puede ser dado de baja.

6.2.12 Modificar estado de las clases

Actor: Desarrollador

El desarrollador asignará el estado finalizado a clases que están finalizadas. Un caso de uso se considera finalizado cuando todas las clases que lo forman están finalizadas. Un módulo se considera finalizado cuando todas las clases que lo forman están finalizadas. Cuando se genere el archivo XML del diagrama de clases, éste contendrá la información sobre el estado de las clases.

Se deben individualizar las clases que son EO.

Flujo de Eventos

En el diagrama de clases y/o en otra interfase donde tenga acceso a las clases, se marcará el estado finalizado.

6.3. Descripción de una herramienta para el seguimiento y control de productos (SSCP)

El trabajo de esta tesina de grado (Romero y Pons, 2008) consistió en desarrollar una herramienta que permite asistir al ingeniero de software en los siguientes aspectos:

- a. Realizar una evaluación rápida de un producto de software de cualquier tamaño, pero con particular interés en los productos medianos y grandes.
- b. Controlar múltiples proyectos de software a medida que van evolucionando con el tiempo.
- c. Definir métricas personalizadas para evaluar la calidad de un producto dado.
- d. Generar datos para una etapa de estimación.
- e. Generar información valiosa para utilizar en proyectos futuros

De esta manera, se distinguen dos aspectos en la herramienta:

- a. Control de avance del proyecto, basado en el tamaño de los casos de uso, medido en T. Esto posibilita conocer información cuantitativa de proyectos, de tal forma que es posible:
 - conocer el estado actual de progreso
 - tener la historia del avance del proyecto
 - obtener métricas de tamaño y métricas de productividad
- b. Caracterización del diseño orientado a objetos de un producto, por medio de la medición de un conjunto no acotado de métricas de diseño orientado a objetos.

Se logró un producto con las siguientes características:

- Capacidad de obtención de información desde modelos UML
- Separación entre la extracción y el procesamiento de los datos
- Generación de información relevante en los distintos niveles
- Integración con el ambiente de desarrollo habitual
- Seguimiento de proyectos en el tiempo
- Monitoreo de múltiples proyectos
- Resultados usables
- Capacidad de definir métricas personalizadas

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

Se muestra en la Figura 13 un diagrama de casos de uso de la herramienta desarrollada y en la Figura 14, una vista de la pantalla que resume la información para el seguimiento y control del proyecto.

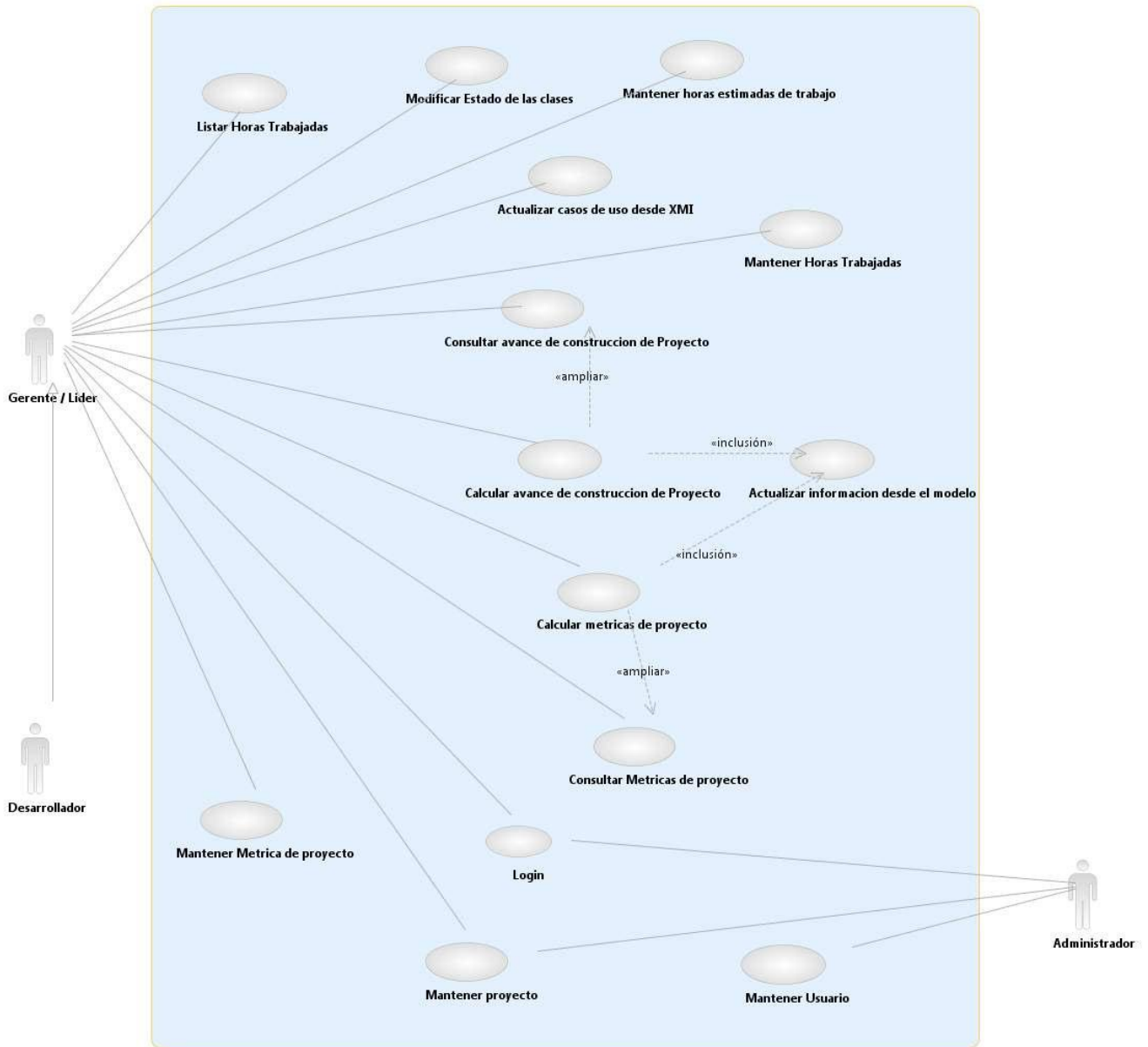


Figura 13. Diagrama de casos de uso de la herramienta SSCP

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

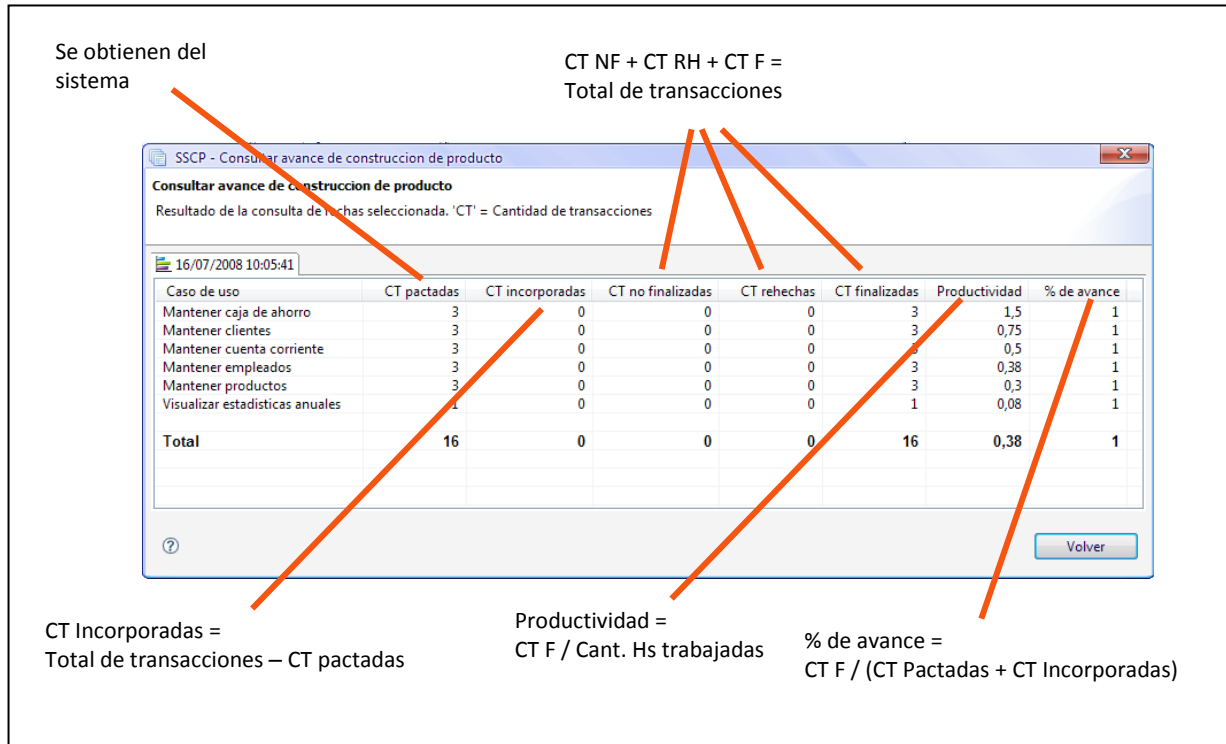


Figura 14 – Vista de la información para seguimiento y control del producto de la herramienta SSCP

En la Figura 15 se muestra la Arquitectura de la herramienta. Se implementó la herramienta como una extensión del Rational Software Architect²¹ -versión 7-, en el lenguaje Java, con SWT²² y jFace, para hacer que la herramienta sea portable.

²¹ Rational Software Modeler, Herramienta de modelado y diseño de aplicaciones, <http://www.ibm.com/software/awdtools/modeler/swmodeler>

²² Standard Widget Toolkit, <http://www.eclipse.org/swt/>

DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA PARA EL SEGUIMIENTO Y CONTROL DE PRODUCTOS DE SOFTWARE

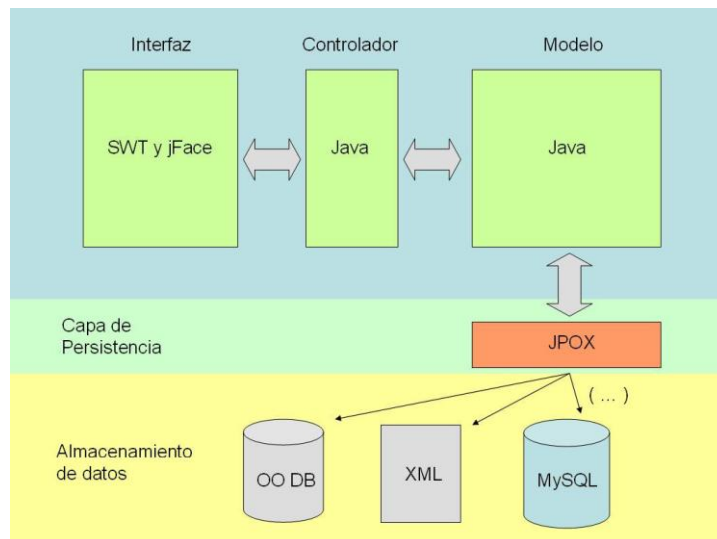


Figura 15. Arquitectura de la Herramienta SSCP

Para la persistencia de la información se utilizó el motor de bases de datos relacional MySQL²³, dada su madurez, difusión y aceptación. De esta manera, los resultados generados por el sistema eventualmente deben poder ser utilizados por otro sistema. Al mismo tiempo, para flexibilizar el uso de una determinada Base de Datos, se decidió utilizar para la persistencia de la información el framework de JPOX (Java Persistent Objects)²⁴, que implementa las interfaces de persistencia JDO y JPA, con la ventaja que se puede realizar consultas en SQL, JDOQL y JPQL. También, se eligió la utilización del estándar XMI para la extracción de los datos.

Se logró medir, en forma automatizada, las T de la siguiente forma:

- Inicialmente, se cuentan manualmente -de la descripción textual de los casos de uso- la cantidad de T. Estas T son ingresadas a la herramienta SSCP como T "pactadas".
- El RSA es usado en forma normal, a excepción de que se imponen dos restricciones: todo diagrama de secuencia va a albergar una sola T, y se debe agregar a cada clase, a modo de estereotipo, -facilidad del RSA- un estado de la clase. Los estereotipos se definen por medio de los perfiles. En este caso, se definió un perfil con un estereotipo que extiende a las clases, definiendo 3 posibilidades: "No Finalizada", "Re-hecha" y "Finalizada". Este perfil fue exportado a un archivo

²³ MySQL, Motor de Bases de datos relacional, <http://www.mysql.com>

²⁴ Java Persistent Objects, mapeador Objeto-Relacional que implementa la interfaz JDO), <http://www.jpox.org>

- “EstadoClase.profile.uml” y debe ser aplicado al proyecto en el cual se quiere tener estas características de extensión.
- c. El RSA permite exportar los diagramas de secuencia y de clases en archivos XML. En estos archivos XML, la herramienta SSCP identifica las T de cada caso de uso y el conjunto de clases que constituye el diagrama.
 - d. El estado de las clases implica el estado de las T:
 - Al menos 1 clase en estado “Re-hecha”, entonces toda la transacción se considera Re-hecha
 - Al menos 1 clase en estado “No-Finalizada”, entonces toda la transacción se considera “No-Finalizada”
 - Todas las clases “Finalizadas”, entonces la transacción se considera finalizada

6.4 Conclusiones del capítulo

A modo de conclusión, se analizan las lecciones aprendidas con la construcción de la herramienta SSCP.

- a. Se logró el objetivo principal que es conocer el avance del proyecto.
- b. Fue posible realizar una implementación de meta-diseño. No se implementó para los OE y el cambio automático de estados (cfr. 6.2.2), por falta de recursos.
- c. Fue oportuna la elección del estándar XMI para la extracción de información de las herramientas de modelización, dado que permiten la extracción de datos de diferentes ambientes de diseño y desarrollo.
- d. La extensión realizada al RSA resultó amigable y totalmente integrada al contexto de desarrollo. Se espera que esto facilite su uso.
- e. La arquitectura de capas también permite integrar el SSCP con la base de datos usada por los desarrolladores.
- f. Las restricciones incorporadas para contar las T -todo diagrama de secuencia va a albergar una sola T, y se agregará a cada clase, a modo de estereotipo, un estado de la clase- son fáciles de implementar , y modifican poco la forma de trabajo del desarrollador. Al mismo tiempo, ésto se ha resuelto de tal modo que sea natural a la UML, usando los recursos de RSA. No se han introducido conceptos nuevos, por ejemplo, el concepto de estereotipo está en el UML.
- g. Falta contar las T, EO y P en código, para comprobar la completa
traceabilidad de las métricas.

7. Conclusiones

La idea directriz de este trabajo ha sido encontrar una unidad de medida similar al “metro cuadrado”, para la industria del software. El interés por este tipo de unidad de medida radica en el hecho de que, en la industria de la construcción, la misma se utiliza como base para realizar estimaciones de costos, dado que mide el tamaño de las construcciones. Para desarrollar esta idea, se optó por trabajar en un contexto de UML, donde los casos de uso son el modelo que se utiliza para la definición de los requerimientos. Los casos de uso tenían una ventaja que se consideró prioritaria: son escritos en una etapa temprana del ciclo de vida del desarrollo de software.

Esta idea inicial se materializó en un conjunto de unidades de medida, debido a que la experimentación demostró que con una sola unidad de medida, no era posible abarcar la realidad compleja del software. Por lo tanto, se seleccionaron tres atributos esenciales y al mismo tiempo, básicos: funcionalidad, datos y complejidad.

Con el objetivo de identificar funciones, se seleccionaron a las T, porque, al ser la respuesta de un sistema a un estímulo dado, se pueden identificar con cierto grado de objetividad a partir del estímulo del actor, logrando de esta forma la identificación de una funcionalidad concreta. Con el objetivo de identificar datos, se seleccionaron a los EO porque contienen la información persistente de los sistemas. Con el objetivo de identificar complejidad, se seleccionaron a los P, dado que se encontró un modo simple de calcular la complejidad ciclomática, que puede ser adaptado para ser aplicado en la descripción textual de los casos de uso.

Por lo tanto, las tres unidades de medida que se definieron fueron: T, EO y P. La cuantificación, tanto del caso de uso como de la aplicación, se mide y expresa ya sea en T, en EO o en P. Son medidas que expresan características diferentes de una aplicación: funcionalidad, datos y complejidad.

Se definieron dichas métricas, y se describió su modo de cálculo. Se aplicaron a un conjunto de proyectos del ámbito académico y de la industria del software. Corroborando que es posible su conteo, a partir de la descripción textual de los casos de uso y usarlas en métodos de estimación temprana de esfuerzo, con la finalidad de reducir el error de la estimación.

7.1 Contribuciones principales

No sólo se buscaban nuevas unidades de medida, sino también, que las unidades de medidas satisficieran un conjunto de características que se consideraban claves para su aceptación y uso:

CONCLUSIONES

- a. *Simple*. Cada unidad de medida captura solamente un aspecto de una aplicación: funcionalidad, datos o complejidad. Indudablemente dichos aspectos deben estar separados, para facilitar el entendimiento intuitivo de lo que expresan. Además, se ha alcanzado la simplificación del método de conteo porque:
- cada métrica tiene sólo un elemento, lo que evita contar más de una vez una misma cosa
 - los pesos de complejidad usados en UFP Y UUCP no se usaron, para evitar reducir el cálculo realizado en una escala absoluta a una escala ordinal.
 - Los Factores de Complejidad Técnica y Ambiental, usados en algunas de las métricas de tamaño funcional, no ~~son~~ fueron tenidos en cuenta.
- b. *Temprana*. Las tres métricas se pueden usar en un momento temprano del ciclo de vida del desarrollo de software. La descripción textual de los casos de uso, base del conteo de los elementos de cada unidad de medida, se escribe en una etapa temprana del desarrollo del producto.
- c. *Precisa*. Las métricas reducen el error de la estimación de esfuerzo, si se lo compara con los errores producto de los cálculos basados en las unidades de medida UFP y UUCP. En las conclusiones de la demostración empírica se destaca que usando las métricas propuestas fue posible disminuir el error de la estimación de esfuerzo para un conjunto de pequeños proyectos de negocios, si se compara con las métricas UFP y UUCP. La selección de un estándar internacional y de otra métrica basada en casos de uso que goza de un importante grado de aceptación para realizar la comparación, resalta la importancia del aporte de este trabajo. Para el caso de HMP, se validaron los resultados con la aplicación del coeficiente de correlación de Spearman. Además, es valioso haber llegado a resultados equiparables usando diversas técnicas: HMP, regresión lineal simple y múltiple y redes neuronales. *Sin embargo, cabe destacar que no es posible extender las conclusiones a todos los proyectos que tengan similares características, a pesar de que se considera a ésta una adecuada selección de proyectos, puesto que la muestra no es aleatoria, ni suficientemente grande, ni variada.*
- d. *Uniforme*. Estas unidades de medida resolvieron las limitaciones de los casos de uso, en cuanto a la variabilidad de su tamaño. La definición de cada una de las métricas propuestas permite representar diferentes tamaños y complejidad de casos de uso, resolviendo la variabilidad de los casos de uso, *sin imponer una forma particular de ser escritos*, lo cual afectaría su adopción. Esto se ha logrado debido a

CONCLUSIONES

que, si bien los casos de uso son propios de UML, las unidades de medida propuestas T, EO y P son inherentes a UML, puesto que por su naturaleza están unidas a UML. Esta es la razón por la cual no ha sido necesario introducir nuevos conceptos; se tomaron elementos que ya existían en el contexto de UML. Cabe destacar que al seleccionar los nombres de las métricas, también se quiso resaltar esta característica. Sin embargo no son algo exclusivo de UML, puesto que podrían ser identificadas en otros contextos.

- e. *Intuitiva*. La simplicidad de cada una de las métricas, tanto desde el punto de vista de los atributos esenciales que capturan (funcionalidad, complejidad y datos), como de los conceptos transacciones, objetos de entidad y caminos, facilita la comprensión intuitiva por parte de todas las personas involucradas en el desarrollo de software.
- f. *Factible de automatización*. Se ha desarrollado una herramienta para el cálculo automatizado de las transacciones, a partir de los diagramas de secuencia del RSA. Esto ha permitido comprobar, que es posible mejorar el control y seguimiento de los proyectos, para los diferentes perfiles –gerente de desarrollo, líder de proyecto, programador-, en un ambiente integrado de desarrollo.

No por simple, el aporte del presente trabajo deja de ser valioso. Es más, según el sentir de algunos autores, representados por Fenton y Pfleeger (1997): “*La unidad de medida a ser usada ha de ser fácil de comprender, simple y consistente con la teoría de medición*”.

7.2. Trabajos futuros

Numerosas líneas de trabajo futuro quedan abiertas a partir de esta tesis. Algunas de ellas son:

- c. *Ampliar el alcance de las conclusiones de este trabajo. Con esta finalidad, se identifican cuatro trabajos posibles:*
 - *Comprobar la eficacia de las unidades de medida en ambientes diferentes*. Sería necesario replicar el trabajo experimental realizado con otros proyectos de software medianos, grandes y de diferente dominios, como por ejemplo, video juegos, aplicaciones que integren voz, imagen, o señales continuas.
 - *Usar las métricas con otros métodos de estimación de esfuerzo*. Uno de los objetivos planteados en este trabajo fue disminuir el error de la estimación. Faltaría comprobar que mejoran el error si son aplicadas o complementarias a métodos de estimación más difundidos, como ser: analogías y juicio de expertos.
 - *Utilizar técnicas de Data Mining*. En este trabajo se utilizó para el cálculo de la estimación de esfuerzo solamente la técnica Redes

CONCLUSIONES

Neuronales. Podrían usarse otras técnicas, como ser: red bayesiana, Classification and regresión trees, case base reasoning.

- d. *Comprobar experimentalmente que el conteo de T, EO y P insume menos tiempo que el conteo de UFP y UCP.* Como el objetivo de este trabajo es buscar una métrica más simple, es necesario realizar un experimento o un caso de estudio, de tal forma que se pruebe objetivamente si la medición con las nuevas métricas, basadas en la descripción textual de los casos de uso, insume un tiempo menor.
- e. *Comprobar experimentalmente que el método de conteo de T, EO y P es más objetivo que el conteo de UFP y UCP.* Existen mediciones del grado de subjetividad e inconsistencias con FP. Por ejemplo, Kemerer investigó las inconsistencias en FP y reportó que las diferencias en el conteo de FP de un mismo sistema rondan un promedio del 12.2%. Por la simplicidad e intuición de las métricas propuestas, se espera que los valores de inconsistencias sean menores para un mismo sistema.
- f. *Comprobar experimentalmente que el uso de documentos complementarios reduce el error en la estimación de esfuerzo.* Es posible utilizar documentos complementarios para realizar el cómputo de las métricas. Los documentos complementarios suelen ser usados en aplicaciones complejas, donde es necesario volcar información que no ha sido incluida en los casos de uso. Se espera que al usar documentos complementarios, ya que introducen un mayor nivel de detalle, se reduzca el error en la estimación.
- g. *Establecerlas como Estándares de medición.* Analizar cuáles son sus debilidades para ser consideradas como estándares. Comparar T con el Standard ISO/IEC 14143-1:1998 y proponer mejoras para que las métricas propuestas sean candidatas a ser promovidas como estándares.
- h. *Comprobar si es factible medir automáticamente las métricas en código.* Este trabajo presenta una herramienta que mide las transacciones a partir de los diagramas de secuencia. Para comprobar la traceabilidad de las métricas desde la descripción textual de requerimientos a código, es necesario implementar una herramienta que las calcule en forma automática desde el código.

Referencias bibliográficas

- Anda, B., Angelvik, E. and Ribu, K. 2002. Improving Estimation Practices by Applying Use Case Models. *Lecture Notes In Computer Science*, Vol. 2559, 383-397.
- Anda, B.C.D., Benestad, H.C. and Hove, S.E. 2005. A multiple-Case study of effort estimation based on use case point. In *Fourth International Symposium on Empirical Software Engineering (Australia, November 17-18, 2005) ISESE'2005*, IEEE Computer Society, 407-416.
- Anda, B., Dreiem, H., Sjøberg, D. and Jørgensen, M. 2001. Estimating Software Development Effort Based on Use Cases-Experiences from Industry. *Lecture Notes In Computer Science*, Vol. 2185, 487 - 502.
- Bernárdez B., Durán A. and Genero M. 2004. Empirical Evaluation and Review of a Metrics-Based Approach for Use Case Verification. *Journal of Research and Practice in Information Technology*, Vol. 36, No. 4, 247-258.
- Briand L.C., Morasca S., Basili V.R. 1996. Property-Based Software Engineering Measurement. *IEEE Transactions on Software Engineering*, Vol. 22, 68 - 86.
- Braz M, Vergilio S. 2006. Effort Estimation Based on Use Cases. *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 221-228.
- Carroll E.R. 2005. Estimating software based on use case points. *Conference on Object Oriented Programming Systems Languages and Applications, Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (San Diego, CA, USA)*, 257 - 265.
- Ceddia J. and Dick M. 2004. Automating the estimation of project size from software design tools using modified function points. *Proceedings of the sixth conference on Australasian computing education - Volume 30. ACM International Conference Proceeding Series; Vol. 57, 33 - 39*
- Charette, R. 2005. Why software fails. *IEEE Spectrum*. September.
- Chen, A., Hihn, J. and Lum, K. 2006. Selecting Best Practices for Effort Estimation. *IEEE Transactions on Software Engineering Vol 32 , Issue 11, 883-895*
- Diev, S. 2006. Use cases modeling and software estimation: applying use case points. *CAN Software Engineering Notes*, Vol. 31, N. 6, 1-4.
- Di Martino, S., Ferrucci, F. and Mendes E. 2007. *First International Symposium on Empirical Software Engineering and Measurement ESEM07*, 324-333.
- Fenton, N.E. and Pfleeger, S.L. 1997. *Software Metrics*. PWS Publishing Company.
- Ferens, D.V. and Gurner, R.B. 1992. An evaluation of three function point models for estimation of software effort *Aerospace and Electronics Conference, 1992. NAECON 1992., Proceedings of the IEEE 1992 National vol.2 635-642*
- Fetcke, T., Abran, A., and Tho-Hau Nguyen. 1997. Mapping the OO-Jacobson approach into function point analysis. *Technology of Object-Oriented Languages and Systems, Tools 23*, 192 - 202.
- Gibbs, W. 1994. Software's Chronic Crisis. *TRENDS IN COMPUTING*, Copyright Scientific American, September, p. 86.
- Hasting, T.E. and Sajeev, A.S.M. 2001. A Vector-Based Approach to Software Size Measurement and Effort Estimation. *IEEE Transactions on Software Engineering*, Vol.27.
- Harput, V., Kaindl, H. and Kramer, S. 2005. Extending Function Points Analysis to Object-Oriented Requirements Specifications. *11th IEEE International Software Metrics Symposium (Como, Italy, September 19-22)*.
- Hispano Portuguesa SPSS. Guía para el análisis de datos SPSS. DOI = <http://www2.uca.es/serv/ai/formacion/spss/Pantalla/verguia.pdf>

REFERENCIAS BIBLIOGRAFICAS

- Issa A., Odeh M. and Coward D. 2006. Software Cost Estimation using Use-Case Models: a Critical Evaluation. *Information and Communication Technologies*, Vol No: 2, 2766- 2771.
- Issa A., Odeh M. and Coward D. 2005. Using Use Case Models to Generate Object Points. *Acta Press, Software Engineering From Proceeding*, 455.
- ISO/IEC 14143-1:1998. Functional size measurement. DOI = www.iso.org.
- ISO/IEC 19761:2003 COSMIC-FFP - A Functional Size Measurement Method. DOI = www.iso.org
- ISO/IEC 20926:2003 IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual. DOI = www.iso.org
- ISO/IEC 20968:2002 Mk II Function Points Analysis - Counting Practices Manual. DOI = www.iso.org
- Jacobson, I., Christerson, M., Jonsson, P. and Övergaard, G. 1992. *Object-Oriented Software Engineering*, Addison Wesley.
- Jorgensen, M. 2004. A review of studies on expert estimation of software development effort. *Journal on System and Software*, Vol. 70, No. 1-2, 37-60.
- Jorgensen, M. y Shepperd. 2007. A systematic review of software development cost estimation studies, *IEEE Transactions on Software Engineering*, Vol. 33, No. 1. p. 3-53, January.
- Karner, G. 1993. *Metrics for Objectory*. Diploma thesis, University of Linköping.
- Kim S., Lively W. and Simmons D. 2006. An Effort Estimation by UML Points in Early Stage of Software Development. *Software Engineering Research and Practice*, 415-421 DOI = <http://ww1.ucmss.com/books/LFS/CSREA2006/SER5194.pdf>
- Kitchenham, B. 1997. Counterpoint: the problem with Function Points. *IEEE Software*, Vol. 14, No. 2.
- Kitchenham, B. and Känsälä K. 1993. Inter-Item Correlations among Function points. *International Conference on Software Engineering Proceedings of the 15th international conference on Software Engineering (Baltimore, Maryland, United States)*, 477 – 480.
- Kitchenham, B.A., Pickard, L.M., MacDonell, S.G. and Shepperd, M.J. 2001. What accuracy statistics really measure. *IEE Proc.-Softw*, Vol. 148, N 3.
- Kusumoto, S., Matukawa F., Inoue K., Hanabusa S. and Maegawa Y. 2004. Estimating effort by Use Case Points: method, tool and case study. *Proceedings of the 10th International Symposium on Software Metrics*.
- Lavazza L.A., del Bianco V. and Garavaglia C. 2008. Model-based functional size measurement. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (Kaiserslautern, Germany)*, 100-109.
- Levesque G., Bevo V. and Cao, D. 2008. Estimating Software size with UML Models. In *Proceedings of the 2008 C³S²E conference, ACM International Conference Proceeding Series*, Vol. 290, 81-87.
- Lokan C. 1999. An empirical study of the corelations between Function Points elements. *Proceedings of the 6th International Symposium on Software Metrics*.
- McCabe, T. 1976. A Complexity measure, *IEEE Transactions on software Engineering*, Vol. SE-2, NO. 4.
- Matson, J.E., Barrett, B.E. and Mellichamp, J.M.1994. Software Development Cost Estimation Using Function Points *IEEE Transactions on Software Engineering*, Volume 20 , Issue 4 (April 1994), 275 - 287
- Mendes, E. 2007. A comparison of Technique for Web Effort Estimation. *First International Symposium on Empirical Software Engineering and Measurement ESEM07*, 334-343.

REFERENCIAS BIBLIOGRAFICAS

- Mendes, E., Mosley, N.; Counsell, S. 2003. Early Web size measures and effort prediction for Web costimation. Software Metrics Symposium, Proceedings. Ninth International, Volume , Issue , 3-5 Sept., 18 – 39.
- Mendes, E., Watson, A., Triggs, C., Mosley, N. and Counsell, S. 2003. Comparative Study of Cost Estimation Models for Web Hypermedia Applications Empirical Software Engineering, Volume 8 , Issue 2, p. 163 – 196
- Mendes, E., Watson, A., Triggs, C., Mosley, N. and Counsell, S. 2002. A Comparison of Development effort estimation techniques for Web Hypermedia Applications. Proceedings of the Eight Symposium on Software Metrics (METRICS 02).
- Mohagheghi P, Anda B. and Conradi R. 2005. Effort estimation of use cases for incremental Large-scale Software development. Proceedings of the 27th international conference on Software engineering, 303 – 311.
- Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S.S., Gallis, H., Lien, A.C. and . Hove, S.E. 2004. A Survey on Software Estimation in the Norwegian Industry, InProceedings of the 10th International Symposium onSoftware Metrics, Chicago, IL, USA, 14-16 September 2004, IEEE Computer Society Press, Chicago, USA, p.208–219.
- Montgomery, D, Peck, E.A., and Vining, G.G. 2004. Introducción al análisis de regresión Lineal, Compañía Editorial Continental.
- Ohsugi, N., Monden, A., Kikuchi, N., Braker, MD., Tsunoda, M., Kakimoto, T. and Matsumoto, K. 2007. Is This Cost Estimate Reliable – The Relationship between Homogeneity of Analogues and Estimation Reliability- First International Symposium on empirical Software Engineering and Measurement ESEM07, 384-392.
- Olsina, L and Martín, M. 2004. Ontology for Software Metrics and Indicators. Journal of Web Engineering, Vol.2 Nro. 4 (2004). Rinton Press, 262-281.
- Robiolo, G. 2004. Métricas de Diseño orientado a Objetos. Una aplicación de Java. Tesis de Magister UNLP.
- Robiolo, G. y Orosco, R. 2006. A Preliminary Study on the Early Estimation of Effort based on Use Cases. ASSE 2006 Argentine Symposium on Software Engineering, Mendoza, Argentina, September 6 – 7.
- Robiolo, G, Orozco, R. 2007. An alternative method employing uses cases for early effort estimation. Software Engineering Workshop SEW 2007.
- Robiolo, G. A simple approach that improves early effort estimation based on use cases. Proceedings of the Third International Doctoral Symposium on Empirical Software Engineering, IDoESE, 2008, Kaiserslautern.
- Robiolo, G., Orosco, R. 2008. Employing use cases to early estimate effort with simpler metrics. Innovations in Systems and Software Engineering, NASA Journal. Springer, Volume 4, Number 1, April, 31-43(13).
- Romero, A, Pons, C. 2008. Una herramienta para medición, control y seguimiento de proyectos. Trabajo de grado de la carrera Ingeniería en Informática, Universidad Nacional de La Plata.
- Ruhe, M., Jeffery, R., Wiczorek, I. 2003. Using Web Objects for estimating software development effort for Web applications. In Proceedings of the Ninth International Software Metrics Symposium (Sydney, Australia, 3-5 September 2003), 30.
- Romero, A, Pons, C. Una herramienta para medición, control y seguimiento de proyectos”. Trabajo de grado de la carrera Ingeniería en Informática, Universidad Nacional de La Plata.
- Smith, J. 1999. The estimation of effort based on use cases. IBM Rational Software . White Paper.
- Symons, Charles R.1988. Funtion Point: Difficulties and Improvements. IEEE Transactions in Software Emgineering

REFERENCIAS BIBLIOGRAFICAS

- Stensrud, E., Foss T., Kitchenham B., Myrtveit I. 2002. An Empirical Validation of the Relationship Between the Magnitude of Relative Error and Project Size. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS.02)
- Sagardía, P. y Ale, J. 2008. Estimaciones en Ingeniería de Software utilizando Data Mining Multirelacional. Trabajo de Grado de la carrera Ingeniería en Informática de la Universidad Asutral.
- Uemura, T., Kusumoto, S. and Inoue, K. 1999. Function points measurement tool for UML design specification. Proceeding. of the Sixth International Software Metrics Symposium (Boca Raton, Florida, US).
- Vinsen, K., Jamieson D. and Callender, G. 2004. Use Case Estimation - The Devil is in the Detail. Proceedings of the Requirements Engineering Conference. 12th IEEE International RE'04, Vol. 00, 10-15.
- Watson, A.H. and Mac Cabe, T.J.. 1996. Structured Testing: Atesting Methodology using the Complejidad Ciclomática Metric, Computer Systems Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-0001 DOI=<http://www.mccabe.com/pdf/nist235r.pdf>

Glosario de siglas y términos

Actor: conjunto coherente de roles que los usuarios de los casos de uso desarrollan cuando interactúan con los casos de uso.

Archivo externo: elemento de FP. Es un archivo referenciado por la aplicación, pero mantenido por otro sistema.

Archivo Interno: elemento de FP. Es un archivo lógico de la aplicación.

Árboles de clasificación y regresión: en Data Mining y Machine Learning es un modelo predictivo, donde las hojas representan clasificaciones y las ramas representan las conjunciones de rasgos que conducen a aquellas clasificaciones. Los árboles de clasificación tienen una salida discreta y los de regresión una salida continua.

Atributos internos: son aquellos que pueden ser medidos en término del producto, examinándolo en forma independiente de su comportamiento.

Atributos externos: son aquellos que pueden ser medidos solamente en su relación con su entorno. En este caso se tiene en cuenta su comportamiento.

Bayesian Network: Una red bayesiana es un modelo probabilístico multivariado que relaciona un conjunto de variables aleatorias mediante un grafo dirigido, que indica influencia causal.

CART: Classifications and regression trees, ver Árboles de clasificación y regresión

Case based reasoning: Razonamiento basado en casos es un tipo de sistema experto, que consiste en el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores.

Casos de uso: descripción de un conjunto de secuencias de acciones que un sistema realiza, y que produce un resultado de valor observable por un actor.

Consulta externa: elemento de FP. Es una transacción donde, a partir de una solicitud de datos, se obtiene una respuesta del sistema.

Complejidad: atributo que caracteriza a un entregable, calculado como la cantidad de relaciones que existen entre sus elementos. El aumento o disminución de complejidad está unido a la mayor o menor cantidad de relaciones entre los elementos.

Complejidad Ciclomática: es una métrica de software que suministra una medición cuantitativa de la complejidad lógica de un programa. Es una de las métricas de complejidad de software más ampliamente

aceptada, ya que ha sido concebida para ser independiente del lenguaje.

DI: intercambio de diagramas ver XMI.

Entrada externa: elemento de FP. Transacción que ingresa datos al sistema.

Entregable: Es una pieza de información que es usada o producida por un proceso de desarrollo de software. En inglés es "artifacts".

Estereotipo: es una extensión del vocabulario de UML, que permite crear nuevos tipos de bloques constructores que son derivados de los ya existentes, pero al mismo tiempo, son específicos de un problema particular.

Estimación de expertos: es un trabajo de estimación conducido por una persona reconocida como un experto. Una parte significativa del proceso está basado en un razonamiento no explícito y no recuperable (intuitivo).

Error Cuadrático Medio: es igual a la raíz cuadrada, de la media de los cuadrados de las diferencias, entre los valores reales y estimados.

FUSP: Fuzzy Use Case Size Points. Métrica de software. Es una modificación del Use Case Size Point. Considera conceptos de la teoría Fussy Set para crear una clasificación gradual y mejorar el manejo de la incertidumbre.

Factor de complejidad técnica: factor aplicado a los puntos función no ajustados para obtener los valores de puntos función ajustados.

FIV: Factor de Inflación de la Varianza. Es el inverso del nivel de tolerancia. Recibe este nombre porque son usados en el cálculo de la varianza de los coeficientes de regresión. Cuanto mayor es el FIV, mayor es la varianza del coeficiente de regresión.

Framework (marco): Micro-arquitectura que provee una grilla (template) incompleta para sistemas de un dominio específico.

Función Transaccional: utilizada en el cálculo de FP. Representa la funcionalidad provista al usuario para procesar los datos de una aplicación.

JFace: proporciona una serie de elementos muy frecuentes a la hora de desarrollar interfaces gráficas de usuario, tales como cuadros de diálogo. Mejora la productividad de los programadores

Puntos función: métrica de software que calcula el tamaño funcional de una aplicación. Su uso es independiente del lenguaje de programación en el cual la aplicación esta implementada.

Puntos casos de uso: métrica de software que calcula el tamaño funcional de una aplicación. Es una adaptación de puntos función a UML.

Magnitud del error relativo: se define como el valor absoluto del cociente entre la diferencia entre el esfuerzo real y el esfuerzo estimado, y el esfuerzo real

MOF: The Meta-Object Facility (MOF) es un estándar de Object Management Group (OMG) para la ingeniería orientada a modelos.

Outliers: valor atípico en estadística, un outlier es una observación que es numéricamente distante del resto de los datos.

Productividad: cociente entre el Tamaño y el Esfuerzo o cociente entre la Complejidad y el Esfuerzo.

Tamaño: atributo que caracteriza la dimensión de un entregable. Se calcula sumando la cantidad de elementos que tiene un entregable. El aumento o disminución de tamaño está unido a la mayor o menor cantidad de elementos.

Tipo Elemento de dato: utilizado para calcular la complejidad de los elementos de FP. Es un dato no sub-divisible.

Tipo Elemento Registro: utilizado para calcular la complejidad de los elementos de FP. Es un sub-grupo de elementos de datos reconocible por el usuario dentro de IF o EF.

Tipo de Referencia a archivo: utilizado para calcular la complejidad de los elementos de FP. Es una lectura o mantenimiento de un archivo interno o externo.

Transacción: es una interacción completa entre el actor y el sistema. En este trabajo, se identifica a la transacción en un caso de uso, o diagrama de secuencia, desde el estímulo del actor hasta que el actor recibe la respuesta a ese estímulo.

Rational Unified Process: proceso unificado de desarrollo de software desarrollado por Rational.

Regression Trees: ver Árboles de clasificación y regresión.

Regresión lineal simple y múltiple: La regresión lineal es usada para modelar el valor de la variable dependiente basada en su relación lineal con una o más variables independientes. La regresión lineal múltiple es una extensión de la regresión lineal simple. Busca la mejor estimación de parámetros de tal modo que se minimice el error por medio del método de mínimos cuadrados. Este método es comúnmente usado como técnica de estimación de costo.

Residuos: Los residuos son las diferencias entre los valores observados y los esperados

Salida externa: elemento de FP. Es una transacción que entrega una salida de datos del sistema.

SQL: es un lenguaje de programación de base de datos diseñado para la recuperación y la dirección de datos en sistemas de dirección de base de datos relacionada (RDBMS).

Stakeholders (interesados o involucrados): son las personas que pueden afectar o son afectados por un sistema o aplicación en particular.

Stepwise regression: la regresión paso a paso incluye modelos de regresión en los cuales la opción de variables predictivas es realizada según un procedimiento automático.

Traceabilidad: es la capacidad de remontar (identificar y medir) todas las etapas que condujeron a un punto particular en un proceso que consiste en una cadena de acontecimientos interrelacionados.

Objetos de Entidad: son los objetos persistentes que pertenecen al vocabulario del dominio del sistema.

UML: es un lenguaje de modelización del software que ha sido unificado.

Use Case rouge: es un método de estimación desarrollado por (Issa et al., 2006). El término rouge significa limpiamente correcto pero no exacto o detallado; aproximado.

Use case pattern catalogue: es un método de estimación desarrollado por (Issa et al., 2006). Se usa para estimar la reutilización del software.

USP: Use Case Size Points, métrica de software, que considera la estructura interna del caso de uso, mejorando la captura de la funcionalidad de un sistema.

Vector Prediction Size Measure: métrica de software que incorpora la funcionalidad y complejidad en un modo balanceado y ortogonal.

GLOSARIO DE SIGLAS Y TERMINOS

XML: Extensible Markup Language. Se traduce como un lenguaje de marcas ampliable. Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

XMI: XMI o XML Metadata Interchange, que se traduce como XML de Intercambio de Metadatos, es una especificación para el Intercambio de Diagramas.

Z: es una forma de cálculo del error que se define como el cociente entre el valor estimado y el valor real.

Apéndices

Apéndice I. MRE para diferentes métodos de estimación

Fila	Referencia	Métricas	Método de estimación	MRE	Descripción de los datos
1	(Mendes, 2007)	Variables from Tukutuku Data Base	Bayesian Network Stepwise Regression Case-based Reasoning (different criteria) CART	0.343 0.948 (1.381, 1.347, 2.03) 6.904	150 projects Tukutuku Data Base Web effort
2	(Chen, et al. 2006.)	Lines of code	COCOMO COSEKMO	(0.45,0.58,1.88) (Min,Median,Max) (0.45,1.57,6.49) (Min,Median,Max)	19 modeling results
3	(Mendes, et al., 2003)	Web size measures	Early Web size measures, Stepwise Regression and Case-based Reasoning	[0.56 - 2.97] MMRE	36 Web Projects
4	(Ohsugi, et al. 2007)	14 variables, incluye FP	Analogy-based cost estimation method	(0.747, 0.607, 1.515) MMRE	104 projects 3 levels of Homogeneity (High, Medium, Low)
5	(Matson, et al.,1994.)	FP	Regresión lineal (transformación logarítmica) Regresión lineal múltiple	Model A: 0.87 MMRE Model B: 0.71 MMRE	104 company data
6	(Braz, and Vergilio, 2006)	UCP FP USP (Use Case Size Points) FUSP(Fuzzy Use Case Size Points)	Productivity	[0.391- 0.776] [0.034 – 0.608] [0.17 – 0.679] [0.114 – 0.676]	4 modules of a system, each one 1 month work
7	(Anda, et al., 2005)	Use Case Points Expert estimation	Productivity	[0.00 - 0.54] [0.21 – 0.76]	1 project, 4 different development processes
8	(Ferens, and Gurner, 1992)	FP	Lineal regression	Mean: 0.46 Standard Deviation: 0.53	36 projects
9	(Moløkken-Østvold, et al., 2004)	Data projects	84% Expert Estimation 16% Combination of expert estimation and model estimation	Mean 0.41 (Brebias similar to MRE) Median 0.21(Brebias)	Set of 44 projects from Norwegian Industry

APENDICES

10	(Issa, et al., 2006)	Use cases	Use Case rough	[0. 45– 0.53] MMRE	Wide spectrum of software projects
11	(Ruhe et al. ,2003)	FP Web objects	Lineal regression Lineal regression Expert estimation	0.33 MMRE 0.24 MMRE 0.37 MMRE	12 web application from Australian small software company
12	(Mendes, Watson et al. 2003)	8 size and complexity variable	Cased Based Reasoning, Stepwise Regression, Regression Trees	[0.13 – 0.29] MMRE	37 web hypermedia application developed by postgraduate students
13	(Anda, Angelvik, et al. 2002)	Use case Points Expert estimation	Productivity	[0.04 – 0.30] [0.18 -0.25]	3 projects
14	(Di Martino, et al. 2007)	25 application variables	Case-based Reasoning Stepwise Regression	[0.14 – 0.21] [0.12 –0.23]	15 web applications
15	(Hasting, T.E. and Sajeev, A.S.M. 2001)	Vector Prediction Size Measure	Vector Prediction Model	0.20	8 industry projects from different organizations, application domains, application types
16	(Mendes et al. 2002)	8 size and complexity	Stepwise regression Case based reasoning	[0.02 – 0.23] [0.10 – 0.32]	37 web hypermedia application developed by postgraduate students

Apéndice II. Definición matemática de Tamaño y Complejidad
(Briand et al. 1996)

Representación de Sistemas y Módulos.

Un sistema S será representado como un par $\langle E, R \rangle$, donde E representa el conjunto de elementos de S, y R es una relación binaria sobre la E ($R \subseteq E \times E$) que representa las relaciones entre los elementos de S. Considerando un sistema la $S = \langle E, R \rangle$, un sistema $m = \langle E_m, R_m \rangle$ es un módulo de S, si y sólo si, $E_m \subseteq E$, la $R_m \subseteq E_m \times E_m$, y $R_m \subseteq R$.

a. Tamaño

Intuitivamente, el tamaño es reconocido como un concepto de medida importante. Según el planteo aquí presentado, el tamaño no puede ser negativo (propiedad Tamaño 1), y se espera que sea nulo cuando un sistema no contiene ningún elemento (el Tamaño 2). Cuando los módulos no tienen elementos en común, se espera que el tamaño sea aditivo (Tamaño 3).

El tamaño de un sistema S es una función $Tamaño(S)$, que es caracterizado por las propiedades siguientes:

TAMAÑO 1: *No negatividad.*

El Tamaño de un sistema $S = \langle E, R \rangle$ es no-negativo si $Tamaño(S) \geq 0$.

TAMAÑO 2: *Valor nulo.* El Tamaño de un sistema $S = \langle E, R \rangle$ es nulo si E es vacío. Entonces, $E = \emptyset$, entonces $Tamaño(S) = 0$

TAMAÑO 3: *Aditividad de Módulos.* El Tamaño de un sistema $S = \langle E, R \rangle$ es igual a la suma de los Tamaños de dos de sus módulos $m_1 = \langle E_{m1}, R_{m1} \rangle$ y $m_2 = \langle E_{m2}, R_{m2} \rangle$, si cualquier elemento de S es un elemento de m_1 o m_2 ,

$(m_1 \subseteq S \text{ y } m_2 \subseteq S \text{ y } E = E_{m1} \cup E_{m2} \text{ y } E_{m1} \cap E_{m2} = \emptyset)$ implica que:

$$Tamaño(S) = Tamaño(m_1) + Tamaño(m_2)$$

Por lo tanto, el tamaño de un sistema $S = \langle E, R \rangle$ puede ser calculado sumando los módulos disjuntos $m_e = \langle \{e\}, R_e \rangle$, cuyo conjunto de elementos está compuesto por un elemento diferente de E. Cumpliéndose que para cada m_e es $R_e = \emptyset$ o $R_e = \{ \langle e, e \rangle \}$.

$$Tamaño(S) = \sum_{e \in E} Tamaño(m_e)$$

Las propiedades enunciadas anteriormente se mantienen cuando se realiza una transformación en una escala ratio (por ejemplo, $f(x) = a x$). Por lo tanto, no hay contradicción entre este concepto de tamaño y la definición de medidas de tamaño en una escala ratio.

APENDICES

b. Complejidad

Desde este punto de vista, la complejidad es una propiedad de un sistema, que depende de las relaciones entre sus elementos.

La complejidad de un sistema S es una función Complejidad(S) es caracterizada por las siguientes propiedades:

COMPLEJIDAD 1: *No negatividad*. La complejidad de un sistema $S = \langle E, R \rangle$ es no negativa, Complejidad(S) ≥ 0

COMPLEJIDAD 2: *Valor nulo*. La complejidad de un sistema $S = \langle E, R \rangle$ es nulo si R es vacío, $R = \emptyset$ implica que Complejidad(S) = 0

COMPLEJIDAD 3: *Simetría*. La complejidad de un sistema $S = \langle E, R \rangle$ no depende de la convención seleccionada para representar la relaciones entre sus elementos

($S = \langle E, R \rangle$ y $S^{-1} = \langle E, R^{-1} \rangle$) implica que Complejidad(S) = Complejidad(S^{-1})

COMPLEJIDAD 4: *Modulos Monotonos*. La complejidad de un sistema $S = \langle E, R \rangle$ es no menor que la suma de las complejidades de dos cualquiera de sus módulos sin relación en común,

($S = \langle E, R \rangle$ y $m_1 = \langle E_{m1}, R_{m1} \rangle$ y $m_2 = \langle E_{m2}, R_{m2} \rangle$ y $m_1 \cup m_2 \subseteq S$ y $R_{m1} \cap R_{m2} = \emptyset$)

implica que Complejidad(S) \geq Complejidad(m_1) + Complejidad(m_2).

COMPLEJIDAD 5: *Aditividad de módulos disjuntos*. La complejidad de un sistema $S = \langle E, R \rangle$ que tiene dos módulos disjuntos m_1 y m_2 , es igual a la suma de las complejidades de los dos módulos.

($S = \langle E, R \rangle$ y $S = m_1 \cup m_2$ y $m_1 \cap m_2 = \emptyset$) implica que

Complejidad(S) = Complejidad(m_1) + Complejidad(m_2).

Las propiedades mencionadas anteriormente permanecen cuando se aplica la admisible transformación de la escala ratio. Por lo cual, no hay contradicción entre el concepto de complejidad anteriormente definido y la medición de complejidad en una escala ratio.

Apéndice III. Análisis de los supuestos de los modelos de la regresión lineal

En este Apéndice se desarrolla el análisis de los supuestos²⁵, donde la variable dependiente es Esfuerzo en los siguientes modelos de regresión lineal: simple, variables independientes T, P, UUCP; múltiple, variables dependientes T, EO, AP.

a. Normalidad.

A continuación Se muestran, para cada uno de los modelos, dos gráficos que informan sobre el grado en que los residuos estandarizados²⁶ se aproximan a una distribución normal. El histograma de los residuos tipificados con una curva normal supuesta, que se construye tomando una media 0 y una desviación típica de 1.

En las figuras 1 y 2 se ve claramente que no es sostenible el supuesto de que los residuos tienen una distribución normal, debido a que los histogramas tienen una distribución no semejante a una curva normal y en los gráficos de probabilidad normal, los puntos no se encuentran sobre la diagonal del gráfico.

b. Independencia

El estadístico de Durbin-Watson proporciona información sobre el grado de independencia que existe entre los residuos. Este estadístico oscila entre 0 y 4, y toma el valor 2 cuando los residuos son independientes. Los valores menores que 2 indican autocorrelación positiva y los mayores que 2, autocorrelación negativa. Se puede asumir independencia entre los residuos cuando este estadístico toma los valores entre 1.5 y 2.5.

La Tabla 1 muestra los valores de Durbin-Watson para cada uno de los modelos. Se puede observar en todos los modelos, a excepción del modelo donde la variable independiente es T, que se verifica el supuesto de independencia entre los residuos.

Tabla 1. Valores de Durbin-Watson para cada uno de los modelos

Variables Independientes	T	P	UCCP	T-EO-AP
Durbin-Watson	0.87	1.07	1.61	1.96

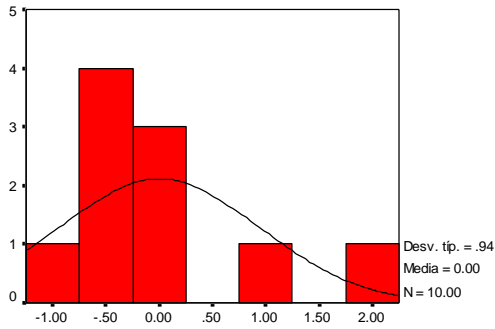
²⁵ <http://www2.uca.es/serv/ai/formacion/spss/Inicio.pdf>

²⁶ residuos divididos por su error estandarizado, tienen una media 0 y una desviación estándar de 1.

Variable Dependiente T

Histograma

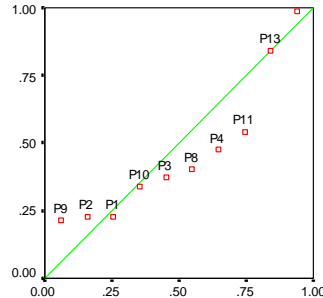
Variable dependiente: AE



Regresión Residuo tipificado

Gráfico P-P normal de regresión Res

Variable dependiente: AE

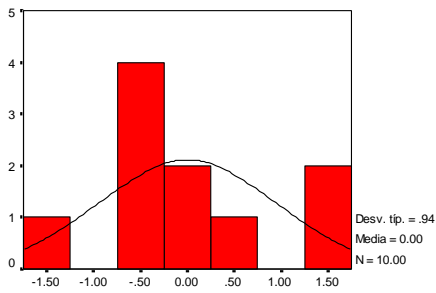


Prob acum observada

Variable Dependiente P

Histograma

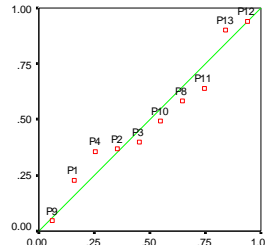
Variable dependiente: AE



Regresión Residuo tipificado

Gráfico P-P normal de regresión Res

Variable dependiente: AE

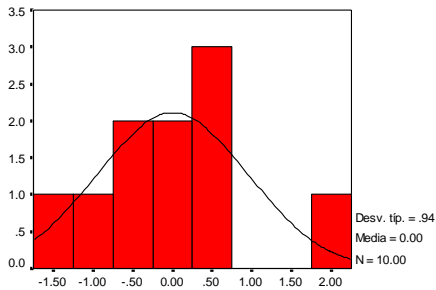


Prob acum observada

Variable Dependiente UUCP

Histograma

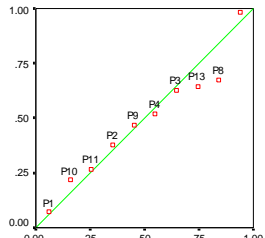
Variable dependiente: AE



Regresión Residuo tipificado

Gráfico P-P normal de regresión Res

Variable dependiente: AE



Prob acum observada

Figura 1. Histograma y Gráfico de probabilidad normal de los modelo de regresión simple

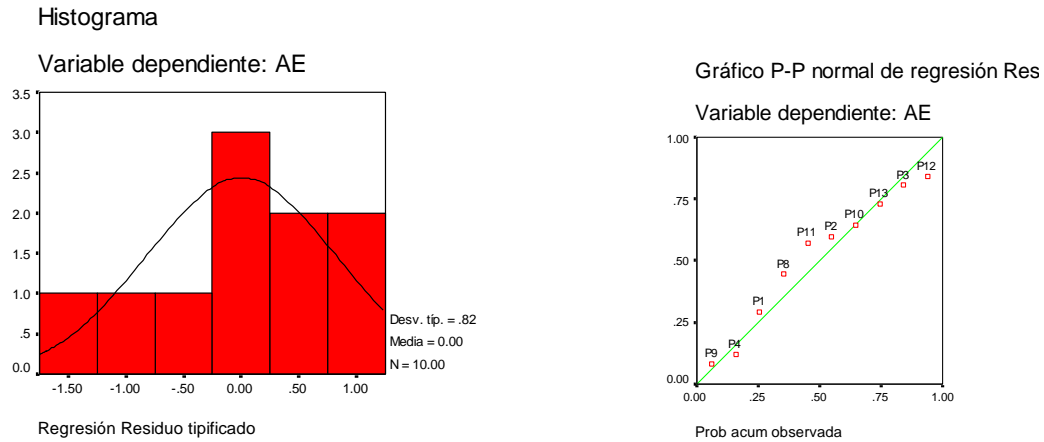


Figura 2. Histograma y Gráfico de probabilidad normal del modelo de regresión múltiple donde las variables dependientes son T, EO y AP.

c. Homocedasticidad.

Se utilizan para el análisis de la homocedasticidad los diagramas de dispersión de las variables ZPRED y ZRESID. El supuesto de igualdad de varianzas implica que la variación de los residuos debe ser uniforme en todo el rango de valores pronosticado, esto es, que el tamaño de los residuos es independiente del tamaño de los pronósticos²⁷, de lo que se desprende que el diagrama de dispersión no debe mostrar ninguna pauta de asociación entre los pronósticos y los residuos.

ZPRED corresponde a las estimaciones estandarizadas²⁸, que son los pronósticos divididos por su desviación estándar, o sea, transformados en valores Z (con media 0 y desviación estándar 1). ZRESID son residuos tipificados. El tamaño de cada residuo tipificado indica el número de desviaciones estándares que se aleja de su media, de modo que si están normalmente distribuidos, el 95% se encontrará dentro del rango (-1.96 - +1.96), lo que permitirá identificar valores de los residuos más grandes que lo requerido.

En la figura 5 se muestran los gráficos de dispersión de las variables ZPRED y ZRESID para los modelos de regresión lineal de variables dependientes T, P, UUCP y T-EO-AP.

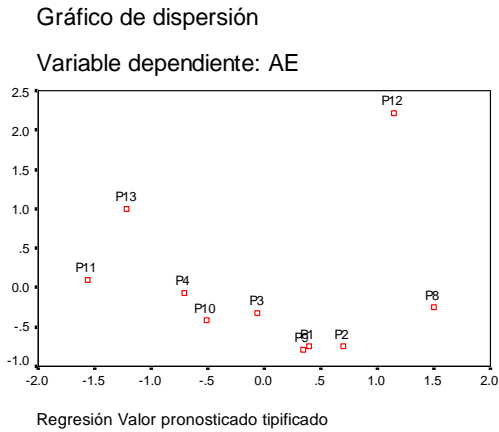
En dichos gráficos no se observa ninguna asociación entre las variables ZPRED y ZRESID, por lo que se concluye que no hay evidencia para rechazar el supuesto de igualdad de varianzas.

²⁷ Valores estimados

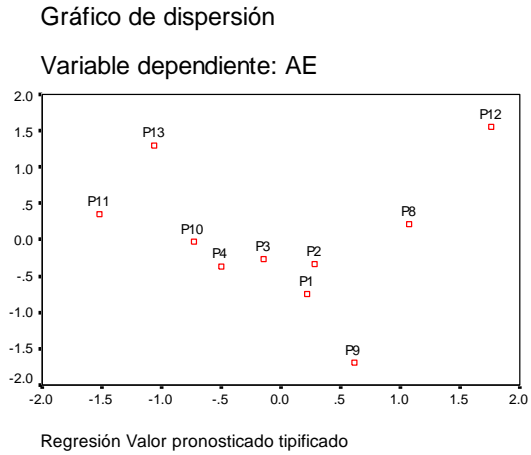
²⁸ Se calcula restando a cada estimación la media de las estimaciones y dividiendo la diferencia, por la desviación estándar de las estimaciones.

APENDICES

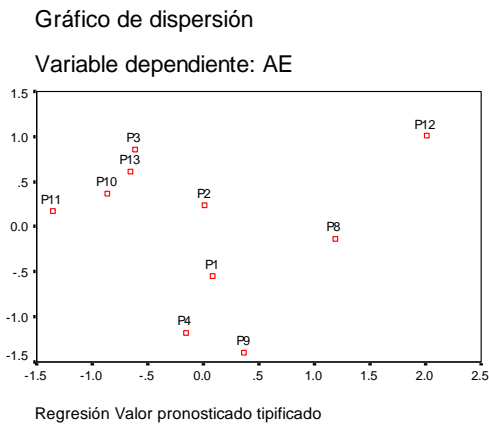
a. T



b. P



c. UUCP



d. T-EO-AP

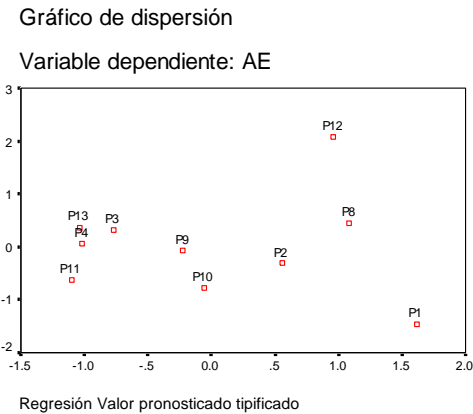


Fig. 3. Gráficos de dispersión de las variables ZPRED y ZRESID para los modelos de regresión lineal de variables dependientes T, P, UUCP y T-EO-AP

d. Linealidad.

Se busca que la relación entre las variables independientes y la dependiente sea lineal. La figura 4 muestra los gráficos de dispersión de los modelos de regresión lineal simple, donde se observa que en todos existe este tipo de relación. La figura 5 muestra los diagramas de regresión parcial. Estos diagramas son similares a los de dispersión, pero no están basados en los valores originales de las dos variables representadas, sino en los residuos obtenidos al efectuar un análisis con el resto de las variables independientes. La recta que mejor se ajuste a la nube de puntos es la definida por los correspondientes coeficientes de regresión. En estos gráficos se puede observar el tamaño y signo de los coeficientes de regresión, los cuales indican, en todos los casos, la existencia de una relación lineal.

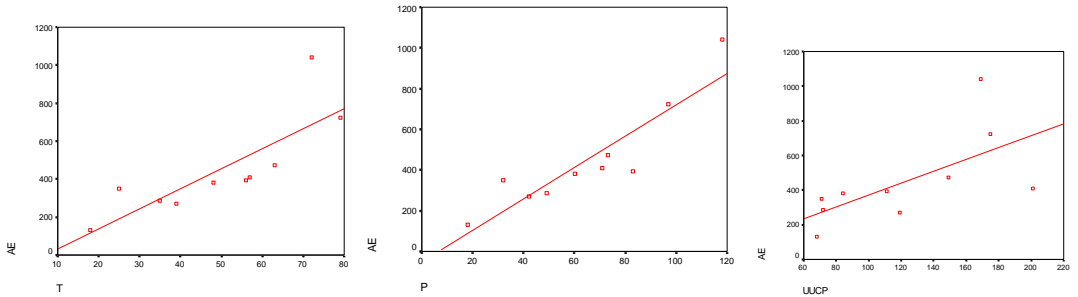


Figura 4. Gráficos de dispersión de los modelos de regresión lineal simple

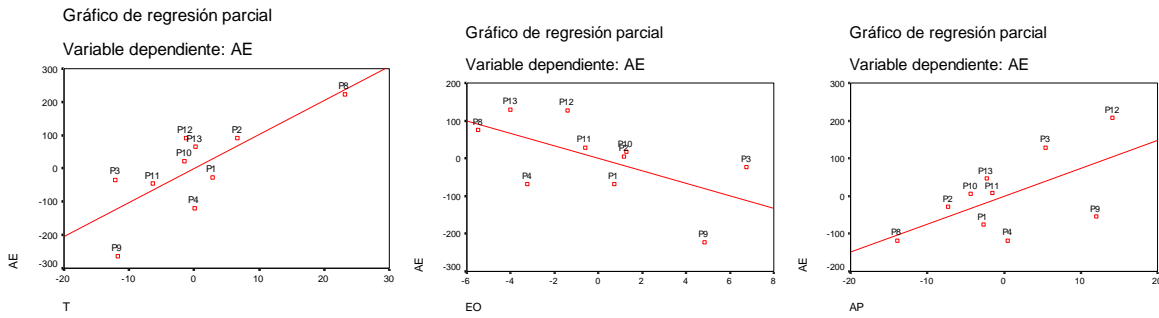


Figura 5. Gráficos de regresión parcial

e. No-colinealidad.

Como indicios de que no existe colinealidad en el modelo de regresión múltiple, se puede citar que:

- Los valores de tolerancia²⁹ no son muy pequeños (0.257, 0.461, 0.415)
- El rango de valores VIF³⁰ es aceptable [2.171 – 3.899], puesto que cuanto mayor es el VIF, mayor es la varianza del correspondiente coeficiente de regresión.
- Los índices de condición³¹ no superan el valor 15; el mayor valor es 12.93

Por lo tanto, se concluye lo siguiente:

- Ninguno de los modelos cumple el supuesto de Normalidad.

²⁹ Se obtiene restando 1 al coeficiente de determinación R²

³⁰ Factores de Inflación de la Varianza, son los inversos de los niveles de tolerancia

³¹ Son la raíz cuadrada del cociente entre el autovalor más grande y cada uno del resto de los autovalores

APENDICES

- T no cumple el supuesto de Independencia
- No hay evidencia, en ninguno de los casos, para rechazar el supuesto de homocedasticidad ni el de linealidad, ni el de no-colinealidad.

Apéndice IV. Tabla de datos modelo multi-relacional

Tabla 1. Datos de las variables a nivel de proyecto

Proyecto	Esfuerzo Real (AE)	T	EO	P	UFP
P1	410.00	57	18	71	144
P2	473.50	63	21	73	266
P3	382.40	48	22	60	171
P4	285.00	35	8	49	142
P5	328.00	21	5	34	89
P6	198.00	23	11	35	75
P7	442.02	33	10	50	57
P8	722.65	79	17	97	210
P9	392.00	56	19	83	311
P10	272.00	39	16	42	155
P11	131.00	18	9	18	117

APENDICES

Tabla 2. Datos de las variables a nivel de caso de uso

Proyecto	Caso de Uso	AE	T	EO	AP	P	WEB	Tipo	Clasif
P1	Login	22	1	1	1	2	1	0	0
P1	Logout	1	1	1	0	1	1	0	0
P1	Listar choferes	9.25	1	1	0	1	1	0	0
P1	Alta de Chofer	14.75	3	1	1	4	1	0	0
P1	Modificación de chofer	8.75	2	1	1	3	1	0	0
P1	Baja de chofer	9.75	2	1	1	3	1	0	0
P1	Detalle de chofer	6.75	1	1	0	1	1	0	0
P1	Listar tripulaciones	11.25	1	2	0	1	1	0	0
P1	Alta de tripulación	10.25	1	2	1	2	1	0	0
P1	Modificación de tripulación	8.25	2	2	0	2	1	0	0
P1	Baja de tripulación	6.25	2	1	0	2	1	0	0
P1	Detalle de tripulación	5.25	1	2	0	1	1	0	0
P1	Listar micros	4.25	1	1	0	1	1	0	0
P1	Alta de micros	10.75	1	1	0	1	1	0	0
P1	Modificación de micros	6.75	2	1	0	2	1	0	0
P1	Baja de micros	2.75	2	1	1	3	1	0	0
P1	Detalle de micros	3.75	1	1	0	1	1	0	0
P1	Listar terminales	5.25	1	1	0	1	1	0	0
P1	Alta de terminal	5.75	1	1	0	1	1	0	0
P1	Modificación de terminal	5.75	2	1	0	2	1	0	0
P1	Baja de terminal	4.75	2	1	1	3	1	0	0
P1	Detalle de terminal	6.75	1	1	0	1	1	0	0
P1	Lista de recorridos	8.5	1	2	0	1	1	0	0
P1	Alta de recorrido	17.5	1	2	0	1	1	0	0
P1	Modificación de recorrido	8	2	2	0	2	1	0	0
P1	Baja de recorrido	4	2	1	1	3	1	0	0
P1	Detalle de recorrido	4	1	2	0	1	1	0	0
P1	Listar viajes	17	1	4	0	1	1	0	0
P1	Alta de viaje	15	2	4	1	3	1	0	0
P1	Modificación de viaje	19	2	4	1	3	1	0	0
P1	Baja de viaje	12	2	1	1	3	1	0	0
P1	Detalle de viaje	14	1	4	0	1	1	0	0
P1	Login	32	1	1	1	2	1	0	0
P1	Editar perfil de la tripulación	19	1	1	0	1	1	0	0
P1	Listado de viaje para una fecha	20	2	3	0	2	1	0	0
P1	Ver el contenido del viaje	4	1	2	0	1	1	0	0
P1	Iniciar Viaje	15.5	1	1	0	1	1	0	0

APENDICES

P1	Realizar parada	13	2	4	1	3	1	0	0
P1	Finalizar viaje	17.5	2	1	1	3	1	0	0
P2	Dar de alta categoría (A_categoría)	56	3	2	1	4	1	0	0
P2	Dar de baja categoría (ABM_categoría)	20	3	2	1	4	1	0	0
P2	Modificar categoría (M_Categoría)	13	3	2	1	4	1	0	0
P2	Dar de alta propiedad (A_categoría)	14.5	3	2	0	3	1	0	0
P2	Asignar Propiedad a categoría (A_categoría)	3	2	2	0	2	1	0	0
P2	Desasignar Propiedad a categoría (A_categoría)	3	2	2	0	2	1	0	0
P2	Dar de Alta Línea de producto (A_linea_de_producto)	27.5	2	1	0	2	1	0	0
P2	Dar de Baja línea de producto (ABM_linea_de_producto)	4.5	2	1	0	2	1	0	0
P2	Modificar Línea de producto (M_linea_de_producto)	7.5	2	1	0	2	1	0	0
P2	Consultar línea de producto (C_linea_de_producto)	5	1	1	0	1	1	0	0
P2	Ingresar producto nuevo al catálogo(A_Producto)	88	3	4	1	4	1	0	0
P2	Eliminar producto del catálogo(ABM_Producto_2)	6	3	4	0	3	1	0	0
P2	Activar producto (ABM_Producto_2)	0	3	4	0	3	1	0	0
P2	Desactivar producto (C_productos_desactivados)	3	3	4	0	3	1	0	0
P2	Alta de pedido (A_pedido_2)	71	3	6	1	4	1	0	0
P2	Consultar datos del Pedido (C_pedido)	4	2	6	1	3	1	0	0
P2	Preaprobar Pedido (preaprobar_pedidos_realizados)	15	3	1	1	4	1	0	0
P2	Ver pedidos realizados (lista_de_pedidos_realizados)	23	1	1	0	1	1	0	0
P2	Armar paquete (A_paquete)	43.5	3	2	1	4	1	0	0
P2	Consultar paquete (C_paquete)	1.5	2	2	0	2	1	0	0
P2	Modificar cotización (M_cotizacion)	8.5	2	2	2	4	1	0	0
P2	Consultar cotización (C_cotización)	4	2	2	0	2	1	0	0
P2	Ver cotizaciones realizadas (C_cotizaciones_realizadas)	15.5	2	2	0	2	1	0	0
P2	Ver motivos de rechazo (C_motivo_rechazo)	4	1	2	0	1	1	0	0
P2	Dar de alta usuario	9.25	2	1	0	2	1	0	0
P2	Dar de baja usuario	7.75	2	1	0	2	1	0	0
P2	Modificar usuario	11.75	2	1	0	2	1	0	0
P2	Consultar usuario	3.75	1	1	0	1	1	0	0
P3	WBS - Programación ABM de Usuarios	56.42	5	1	3	8	1	0	0

APENDICES

P3	WBS - Programación ABM de Servicios	22.18	4	1	1	5	1	0	0
P3	WBS - Programación ABM de Habitación	44.35	7	2	1	8	1	0	0
P3	WBS - Programación Reservar	107.18	15	5	4	19	1	0	0
P3	WBS - Programación Check In	106.69	6	5	2	8	1	0	0
P3	WBS - Programación Carga de consumo	16.02	9	3	1	10	1	0	0
P3	WBS - Programación Check out	29.57	2	5	0	2	1	0	0
P4	Desarrollar 'Ingreso al Sistema'	42.00	2	1	2	4	1	0	0
P4	Desarrollar 'Gestión de Proyectos'	64.00	17	3	2	19	1	0	0
P4	Desarrollar 'Administración de Usuarios'	44.00	4	1	2	6	1	0	0
P4	Desarrollar 'Administración de Recursos'	46.00	4	1	3	7	1	0	0
P4	Desarrollar 'Administración de Empresas'	41.00	4	1	2	6	1	0	0
P4	Desarrollar 'Administración de Empleados'	48.00	4	1	3	7	1	0	0
P10	1. Ver mensaje en el foro	10	1	1	0	1	1	1	0
P10	2. Ver mensajes no leídos	10	1	1	0	1	1	1	0
P10	3. Agregar mensaje al foro	20	3	1	2	5	1	1	0
P10	4. Nuevo tema	25	4	1	0	4	1	1	0
P10	5. Selección de contenidos de interés (custom portal)	20	2	1	1	3	1	1	0
P10	6. ABM de contenidos (custom portal)	25	4	1	0	4	1	1	0
P10	8. Actualización de la información de un contenido	10	6	2	0	6	1	1	0
P10	9. Visualización del calendario	15	1	1	0	1	1	1	0
P10	10. Ver lista de postgrados	8	2	1	0	2	1	1	0
P10	11. Ver nómina de graduados	8	2	4	0	2	1	1	0
P10	12. Ver datos de graduado	5	1	1	0	1	1	1	0
P10	13. Enviar mensaje privado	16	2	3	0	2	1	1	0
P10	14. Ver mensajes privados recibidos	15	1	1	0	1	1	1	0
P10	15. Ver resumen mensajes privados recibidos	15	1	1	0	1	1	1	0
P10	16. Leer mensaje privado	10	3	1	0	3	1	1	0
P10	17. Ver sucesos destacados	10	1	1	0	1	1	1	0
P10	18. Alta de sucesos destacados	20	1	1	0	1	1	1	0
P10	19. Ver discurso de acto final	10	2	1	0	2	1	1	0
P10	20. Alta de discurso de acto final	20	1	1	0	1	1	1	0
P11	Identificación	5	2	1	0	2	0	1	0

APENDICES

P11	Registrar evento	10	2	2	0	2	0	1	0
P11	Buscar deudor	8	2	1	0	2	0	1	0
P11	Registrar nuevo deudor	8	1	1	0	1	0	1	0
P11	Registrar nueva deuda	12	1	2	0	1	0	1	0
P11	Registrar alerta en cronograma	13	1	2	0	1	0	1	0
P11	Registrar seguimiento / cambio de estado	11	2	2	0	2	0	1	0
P11	Registrar cancelación de deuda	13	2	1	0	2	0	1	0
P11	Listado de deudores por estado	10	1	1	0	1	0	1	0
P11	Resumen de saldos	10	1	2	0	1	0	1	0
P11	Evolución de la deuda	13	1	2	0	1	0	1	0
P11	Consulta de cronograma de alertas	9	1	2	0	1	0	1	0
P11	Consulta de detalle de deudor	9	1	4	0	1	0	1	0
P8	1. Ingresar al Sistema	10	1	2	1	2	1	1	0
P8	2. Salir del Sistema	10	1	1	0	1	1	1	0
P8	3. Alta de Aviso Entrante	35	7	4	0	7	1	1	0
P8	4. Enviar Aviso Entrante	20	3	1	2	5	1	1	0
P8	5. Visualizar Aviso	25	2	4	0	2	1	1	0
P8	6. Modificar Aviso Entrante	25	2	4	1	3	1	1	0
P8	7. Eliminar Aviso Entrante	15	2	1	1	3	1	1	0
P8	8. Buscar Avisos (Consulta)	40	2	3	0	2	1	1	0
P8	9. Revisar Cajón	25	3	3	0	3	1	1	0
P8	10. Finalizar Tratamiento de Aviso Entrante	25	3	1	1	4	1	1	0
P8	11. Cerrar Aviso Entrante	25	3	1	1	4	1	1	0
P8	12. Ver casilla de Expedientes Cerrados	20	3	1	0	3	1	1	0
P8	13. Imprimir Aviso	35	2	1	0	2	1	1	0
P8	14. Alta de Expediente	45	6	6	2	8	1	1	0
P8	15. Alta de Expediente Interno	40	6	6	1	7	1	1	0
P8	16. Validar expediente	25	2	1	1	3	1	1	0
P8	17. Buscar Expedientes (Consulta)	40	2	2	0	2	1	1	0
P8	18. Visualizar Expediente	28	2	6	0	2	1	1	0
P8	19. Delegar Expediente en el alta	30	3	2	0	3	1	1	0
P8	20. Delegar Expediente en la finalización	20	2	2	0	2	1	1	0
P8	21. Tratar Expediente	40	6	4	2	8	1	1	0
P8	22. Firmar Expediente	30	2	2	0	2	1	1	0
P8	23. Cerrar Expediente	20	3	2	1	4	1	1	0

APENDICES

P8	24.	Crear Minuta	35	6	2	0	6	1	1	0
P8	25.	Cambiar Password	15	2	1	2	4	1	1	0
P8	26.	Alta de Entidad	25	1	1	1	2	1	1	0
P8	27.	Modificar Entidad	19.65	2	1	1	3	1	1	0
P9	1.	Ingreso al sistema	3	1	2	0	1	1	1	0
P9	2.	Cambio de contraseña	2	2	1	0	2	1	1	0
P9	3.	Administración de usuarios	13	7	1	0	7	1	1	0
P9	4.	Administración de permisos	20	7	2	0	7	1	1	0
P9	5.	Administración de empresas	13.5	9	1	7	16	1	1	0
P9	6.	Administración de ubicaciones	27	4	3	5	9	1	1	0
P9	7.	Administración de categorías	3	7	1	12	19	1	1	0
P9	8.	Alta de bienes desde PC	32	3	9	1	4	1	1	0
P9	9.	Alta de bienes con uso de Pocket PC	36.75	2	4	0	2	1	1	0
P9	10.	Búsqueda y Actualización de Bienes	109	3	9	2	5	1	1	0
P9	11.	Baja de bienes con uso de Pocket PC	36.75	2	1	0	2	1	1	0
P9	12.	Movimiento/traslado de bienes con uso de Pocket PC	36.75	2	3	0	2	1	1	0
P9	13.	Control de inventario con uso de Pocket PC	36.75	2	1	0	2	1	1	0
P9	14.	Planificación de Control de Inventarios	5	2	1	0	2	1	1	0
P9	15.	Cambio de ubicación de los bienes	17.5	3	3	0	3	1	1	0
P5		1-Administrar entidad	20	3	1	0	3	1	0	0
P5		16-Alta de un nuevo usuario	87	2	1	1	3	1	0	0
P5		15-Alta de una banda	17	1	2	2	3	1	0	0
P5		12-Creación de espacio de banda	45	1	1	2	3	1	0	0
P5		23-Creación de espacio del bar	22.67	2	1	2	4	1	0	0
P5		13-Invitar / echar a un músico	31	2	2	2	4	1	0	0
P5		14-Acceder a espacio de banda	20	2	1	1	3	1	0	0
P5		18-Consulta de locales	24	2	1	1	3	1	0	0
P5		5-Crear promociones	13.33	1	3	1	2	1	0	0
P5		6-Definir política del local	12	1	1	0	1	1	0	0
P5		17-Búsqueda de bandas	13	2	1	1	3	1	0	0
P5		3-Consultar historial de una banda	23	2	3	0	2	1	0	0
P6		1. Carga de crédito	57.06	2	2	3	5	2	0	0

APENDICES

P6	2. Pago del Servicio	28.56	2	5	2	4	2	0	0
P6	4. Consulta de Últimos movimientos	26.56	3	3	1	4	2	0	0
P6	15. Reporte de Consumos	15.56	3	2	1	4	2	0	0
P6	6. Consulta de dinero adeudado	15.56	3	3	2	5	2	0	0
P6	3. Consulta de saldo	15.56	2	2	2	4	2	0	0
P6	8. Alta y Baja Clientes	21.06	3	1	0	3	2	0	0
P6	9. ABM Productos y Servicios	18.06	5	1	1	6	2	0	0
P7	Compra de entrada	368.02	13	5	2	15	2	0	0
P7	ABM Complejo	14	4	1	3	7	2	0	0
P7	ABM Salas	14	4	1	3	7	2	0	0
P7	ABM Películas	14	4	1	3	7	2	0	0
P7	ABM Funciones	14	4	1	3	7	2	0	0
P7	ABM Usuarios	18	4	1	3	7	2	0	0