

**Un Procedimiento de Medición de Tamaño Funcional para  
Especificaciones de Requisitos**



PRESENTADO POR: *Nelly Condori Fernández*

*Departamento de Sistemas Informáticos y Computación*

UNIVERSIDAD POLITÉCNICA DE VALENCIA

DIRIGIDO POR:

*Oscar Pastor López*

*Silvia Abrahão*

*Valencia, Mayo 2007*

Esta tesis ha sido presentada en el Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia, cumpliendo los requisitos para el grado de Doctor en Filosofía en Informática.

© Nelly Condori-Fernández Arequipa, Perú

Reservado todos los derechos Impreso en Valencia, España

*Dedico este trabajo a mis padres Celestina y Rosendo, y a mis hermanos Carmen, Yenny y Nelson por su comprensión y apoyo incondicional a lo largo de mi carrera profesional.*



# ***Agradecimientos***

La presente tesis no hubiera sido posible si no hubiera contado con el apoyo recibido a lo largo de esta carrera iniciada aproximadamente hace cuatro años.

En primer lugar a mi director de tesis, Oscar Pastor, por todo el apoyo recibido durante el desarrollo del presente trabajo. Valoro mucho sus consejos e inagotable vitalidad, gracias por esa inquebrantable confianza que me ha permitido siempre avanzar aún en situaciones adversas. A mi codirectora de tesis Silvia Abrahão, gracias por introducirme a nuevos conocimientos de la ingeniería del software empírica, uno de los pilares de esta tesis.

En segundo lugar, mis más sinceros agradecimientos a CARE Technologies S.A, quien financió de manera ininterrumpida el presente trabajo de investigación hasta su culminación.

En tercer lugar, a Alain Abran, con quien tuve la oportunidad de hacer una enriquecedora estancia de investigación durante tres meses en la Universidad de Québec, muchas gracias por los valiosos comentarios y el tiempo brindado, incluso fines de semana. Así mismo, agradezco también a Sioui, Samir, Ayasha, Sofia, Stéphane y Jorge, por hacer que mi estancia en Montreal sea mucho más agradable.

A todos los miembros del grupo OO-Method, en especial a mis compañeros de laboratorio: Sergio, Ignacio, Luis, David, José Luis, y Nathalie por hacerme sentir como en casa. A mis ex-compañeros que a

pesar de emprender otros rumbos, hoy en día seguimos compartiendo una linda amistad: Enric, Isabel Villanueva, José Carlos, David Melo y Marlyn.

A todos mis amigos con quienes he compartido muchas comidas en la “La Vella”: Gonzalo, Javi, Jennifer, Carlos, Manolo, Rafa, Alejandro, Cristóbal, Abel y Artur. En especial a Nour, Maria Eugenia e Isabel (la chama) por estar siempre conmigo en los buenos y malos momentos, muchas gracias por su valiosa amistad.

A mis amigos Leopoldo, Clemente y Rogelio por sus consejos y ánimos constantes, aunque muchas veces haya sido a la distancia.

A Nieves Brisaboa, quién por medio de la Red Iberoamericana RITOS2 pude conocer a Oscar, mi director, gracias por su confianza y apoyo moral.

A mi familia por su comprensión, paciencia, y motivación, gracias queridos padres y hermanos.

Finalmente, a Duncan por su apoyo incondicional, e infinita comprensión, gracias por mostrar siempre interés en muchos de los tópicos de mi tesis. Cariño tu apoyo fue crucial para su culminación.

# ***Abstract***

Nowadays, software size is used in the management and control of software production, and is employed as one of the essential parameters for benchmarking productivity and quality of software projects and delivered products. Although the importance of early size assessment is evident, it is currently only achievable in later phases of the software lifecycle (analysis, design, and implementation).

Software size can be quantified using several approaches such as code lines and Functional Size Measurement (FSM) methods. An FSM method measures the size of software by quantifying the Functional User Requirements. The most widely used FSM method is Function Point Analysis (FPA). This method was developed to measure Management Information Systems (MIS) and assumes the use of traditional software development methodologies such as structured analysis and design.

Although IFPUG FPA has achieved increasing popularity in the industry, its lack of applicability to all software types and the rapid evolution of development paradigms have produced many variations of this method.

In order to address these weaknesses, COSMIC-FFP has emerged as the second generation of FSM methods. This method is an international standard approved by ISO (ISO/IEC 19761). However, the generality of COSMIC-FFP requires instantiation through a more specific and systematic procedure in conjunction with a software development method.

A number of proposals that use the COSMIC-FFP method to measure the functional size of future software applications from high-level specifications already exist. However, these proposals lack rigorous and systematic definition; for instance, a common problem is the lack of a clear definition of the various concepts that contribute to software size, particularly at the requirements modelling level. Another weakness is that validation is not covered by these proposals, despite the fact that the ISO/IEC 14143-3 standard provides a framework for verifying whether an FSM method complies with specific performance properties (e.g. repeatability, reproducibility, accuracy, etc.).

This thesis confronts these problems directly by analyzing and adapting COSMIC-FFP in order to measure the functional size of a software application from its corresponding requirements specification in the OO-Method context. OO-Method is an automatic software production method, based on a model-transformation strategy.

Therefore, the objective of this thesis is twofold: 1) to define a procedure for the automatic measurement of the functional size at the earliest stage of the software production process using COSMIC-FFP; and 2) to evaluate the quality of this measurement procedure by looking at its design, application, and the results obtained. To achieve these objectives, a measurement procedure called RmFFP has been systematically defined. To do this we used the process model for software measurement suggested by Abran and Jacquet.

The thesis dealt with the following: 1) the definition of a set of mapping rules that allows the significant primitives of the Requirements Model to



be identified; 2) the definition of a set of measurement rules for obtaining the functional size of OO systems generated using the OO-Method approach; 3) the application of both rules sets to specific real case studies; 4) design validation by means of conformity evaluation with ISO/IEC 19761, theoretical validation using a formal framework, and an analysis carried out from a metrology perspective; 5) evaluation of the results in terms of accuracy and reproducibility; 6) finally, the empirical evaluation of the perceptions of users employing RmFFP to measure early-stage functional size. We have corroborated that RmFFP is perceived as easy to use and useful and that, furthermore, there is an intention to use it in the future.



# ***Resumen***

Hoy en día el tamaño del software es utilizado en la gestión y control de producción como uno de los parámetros esenciales de los modelos de estimación que contribuyen a la calidad de los proyectos de software y productos entregables. Aunque la importancia de la medición temprana del tamaño es evidente, actualmente esta medición es solamente alcanzada en fases tardías del ciclo de vida del software (análisis, diseño e implementación).

El tamaño de software puede ser cuantificado usando diferentes técnicas, como las líneas de código y los métodos de medición de tamaño funcional. Un método de medición de tamaño funcional mide el tamaño del software cuantificando los requisitos funcionales. El método Análisis de Puntos de Función (FPA) es el método mayormente utilizado. Este método fue desarrollado para medir Sistemas de Información de Gestión desarrollados con métodos tradicionales. Aunque IFPUG FPA ha ido alcanzado mayor popularidad en la industria, este método carece de aplicabilidad a todo tipo de software y a nuevos paradigmas de desarrollo.

Para direccionar estas debilidades, COSMIC-FFP ha surgido como un método de segunda generación y ha sido aprobado como un estándar internacional (ISO/IEC 19761). Existen varias propuestas que usan COSMIC-FFP para medir el tamaño funcional de aplicaciones software. Sin embargo, estas propuestas carecen de una definición rigurosa y sistemática; por ejemplo, un problema común es la falta de claridad en la

identificación de varios conceptos que contribuyen al tamaño de software, particularmente a nivel de modelado de requisitos. Otra debilidad es que la validación y verificación no son tratadas adecuadamente por estas propuestas, a pesar de que el estándar ISO/IEC 14143-3 provee un marco general para verificar si un método de medición de tamaño funcional es exacto, reproducible, repetible, etc.

Esta tesis afronta estos problemas analizando y adaptando COSMIC-FFP para medir el tamaño funcional de futuras aplicaciones de software a partir de su correspondiente especificación de requisitos en el contexto de OO-Method. OO-Method es un método de producción de software basado en una estrategia de transformación de modelos.

Por lo tanto, el objetivo de esta tesis es doble: 1) Definir un procedimiento para la medición automática del tamaño funcional en una fase temprana del proceso de producción de software usando COSMIC-FFP; y 2) evaluar la calidad de este procedimiento en cuanto a su diseño, aplicación y resultados obtenidos. Para lograr estos objetivos, un procedimiento de medición de tamaño funcional llamado RmFFP ha sido sistemáticamente definido. Para hacer esto, se utilizó un modelo general de proceso de medición del software, sugerido por Abran y Jacquet.

La presente tesis incluye: 1) la definición de un conjunto de reglas de representación que permite identificar las primitivas significativas del Modelo de Requisitos; 2) La definición de un conjunto de reglas de medición para obtener el tamaño funcional de sistemas OO generados usando OO-Method; 3) la aplicación de estas reglas a casos de estudio reales; 4) la validación del diseño por medio de la evaluación de

conformidad con la ISO/IEC 19761, la validación teórica usando un marco formal y un análisis llevado a cabo desde una perspectiva de la metrología; 5) la evaluación de los resultados en términos de la exactitud y reproducibilidad; 6) finalmente, la evaluación empírica de las percepciones de los usuarios utilizando RmFFP para medir el tamaño funcional a partir de una fase temprana. Se corroboró que RmFFP es fácil de usar y útil, y además, existe la intención de ser usado en el futuro.



# ***Resum***

Hui en dia el tamany del programari s'utilitza en la gestió i control de la producció com un dels paràmetres essencials dels models d'estimació que contribueixen a la qualitat dels projectes de programari i els corresponents productes finals. Encara que la importància de la mesura del tamany en etapes primerenques és evident, actualment aquest mesurament és només aconseguit en fases tardanes del cicle de vida de desenvolupament de programari (anàlisi, disseny i implementació).

El tamany del programari pot ser quantificat mitjançant diferents tècniques, com el nombre de línies de codi i els mètodes de Mesurament de Tamany Funcional. Un mètode de Mesurament de Tamany Funcional avalua el tamany del programari quantificant els requisits funcionals. El mètode Anàlisi de Punts de Funció (FPA) és el mètode majorment utilitzat. Aquest mètode va ser ideat per a mesurar Sistemes d'Informació de Gestió desenvolupats amb mètodes tradicionals. Encara que IFPUG FPA ha anat aconseguint una creixent popularitat en la indústria, aquest mètode no és d'aplicació a tot tipus de programari i a nous paradigmes de desenvolupament de programari.

Per tal d'abordar aquestes debilitats, COSMIC-FFP ha sorgit com un mètode de segona generació i ha sigut aprovat com a estàndard internacional (ISO/IEC 19761). Existeixen ja diverses propostes que utilitzen COSMIC-FFP per tal de mesurar el tamany funcional d'aplicacions de programari. No obstant això, aquestes propostes no tenen

una definició rigorosa i sistemàtica; per exemple, un problema comú és la falta de claredat en la definició de diversos conceptes que contribueixen al tamany del programari, particularment a nivell de modelatge de requisits. Una altra debilitat és que la validació i verificació no són tractades adequadament per aquestes propostes, a pesar que l'estàndard ISO/IEC 14143-3 proveeix d'un marc de treball general per a verificar si un mètode de Mesurament de Tamany Funcional és exacte, reproduïble, repetible, etc.

Aquesta tesi afronta aquests problemes, analitzant i adaptant COSMIC-FFP per tal de mesurar el tamany funcional de futures aplicacions de programari a partir de la seua corresponent especificació de requisits en el context d'OO-Method. OO-Method és un mètode de producció de programari basat en una estratègia de transformació de models.

Per tant, l'objectiu d'aquesta tesi és doble: 1) Definir un procediment per a la mesura automàtica del tamany funcional en una fase més primerenca del procés de producció de programari utilitzant COSMIC-FFP; i 2) avaluar la qualitat d'aquest procediment quant al seu disseny, aplicació i resultats obtinguts. Per tal d'aconseguir aquests objectius un procediment de mesurament de tamany funcional anomenat RmFFP ha sigut sistemàticament definit. Per a dur a terme aquesta definició s'ha utilitzat el model general de procés de mesurament de programari suggerit per Abran i Jacquet.

La present tesi inclou: 1) la definició d'un conjunt de regles de correspondència que permet identificar les primitives significatives del Model de Requisites; 2) La definició d'un conjunt de regles de mesurament



per a obtenir el tamany funcional de sistemes OO generats usant OO-Method; 3) l'aplicació d'ambdós conjunts de regles a casos d'estudi reals; 4) la validació del disseny per mitjà de l'avaluació de conformitat amb la ISO/IEC 19761, la validació teòrica emprant un marc formal i una anàlisi duta a terme des d'una perspectiva de la metrologia; 5) l'avaluació dels resultats en termes de l'exactitud i reproductibilitat; 6) finalment, l'avaluació empírica de les percepcions dels usuaris utilitzant RmFFP per a mesurar el tamany funcional a partir d'una fase primerenca. Es va corroborar que RmFFP és fàcil d'utilitzar i beneficiós, i a més, es pretén que siga emprat en el futur.

# Índice general

1. Introducción .....	- 9 -
1.1. Motivación .....	- 9 -
1.2. Descripción del problema.....	- 10 -
1.3. Objetivos .....	- 10 -
1.4. Contexto de investigación.....	- 11 -
1.5. Estructura de la tesis .....	- 13 -
2. Estado del Arte.....	- 15 -
2.1. Metrología y medición del software.....	- 16 -
2.1.1. Estándar ISO/IEC 14143 .....	- 20 -
2.1.1.1. Parte 1: Definición de conceptos .....	- 20 -
2.1.1.2. Parte 2: Evaluación de Conformidad.....	- 20 -
2.1.1.3. Parte 3: Verificación de un método de medición..	- 21 -
2.1.1.4. Parte 4: Modelo de referencia.....	- 21 -
2.1.1.5. Parte 5: Dominios funcionales.....	- 22 -
2.1.1.6. Parte 6: Guía de uso de la ISO/IEC 14143 y de los métodos estándares.....	- 22 -
2.1.2. Estándar ISO/IEC 15939 .....	- 22 -
2.1.3. Estándar ISO 5725 .....	- 23 -
2.1.4. Modelo de Proceso de Medición.....	- 24 -
2.2. Métodos de medición de tamaño funcional.....	- 27 -
2.2.1. ISO/IEC 19761: COSMIC-FFP (2003) .....	- 29 -
2.2.2. ISO/IEC 20926: IFPUG 4.1 (2003) .....	- 29 -
2.2.3. ISO/IEC 20968: Mk II FPA (2002) .....	- 30 -

2.2.4. ISO/IEC 24570: NESMA FPA (2004) .....	- 31 -
2.3. <i>Procedimientos de Medición de tamaño funcional</i> .....	- 31 -
2.3.1. Primera generación .....	- 34 -
2.3.1.1. Propuesta de Fetcke (1998) .....	- 34 -
2.3.1.2. Propuesta de Tavares (1999) .....	- 35 -
2.3.1.3. Propuesta de Bertolami (2003) .....	- 36 -
2.3.1.4. Propuesta de Lehne (1997) .....	- 37 -
2.3.1.5. Propuesta de Uemura et al. (1999).....	- 38 -
2.3.1.6. Propuesta de Abrahão et al. (2004).....	- 39 -
2.3.2. Segunda generación .....	- 41 -
2.3.2.1. Propuesta de Bevo (1999).....	- 41 -
2.3.2.2. Propuesta de Jenner (2001).....	- 42 -
2.3.2.3. Propuesta de Azzouz (2003).....	- 43 -
2.3.2.4. Propuesta de Habela et al. (2005).....	- 44 -
2.3.2.5. Propuesta de Poels (2003) .....	- 45 -
2.3.2.6. Propuesta de Diab et al. (2004).....	- 47 -
2.3.2.7. Propuesta de Nagano (2003 ).....	- 48 -
2.4. <i>Discusión</i> .....	- 49 -
3. Modelo de requisitos de OO-Method.....	- 56 -
3.1. <i>Introducción</i> .....	- 56 -
3.2. <i>Ingeniería de Requisitos de OO-Method</i> .....	59
3.2.1. Misión del sistema .....	59
3.2.2. Árbol de Refinamiento de Funciones .....	60
3.2.3. Modelo de casos de uso .....	61
3.2.4. Diagramas de secuencia.....	64
3.3. <i>Herramienta de soporte</i> .....	70
3.4. <i>Conclusiones</i> .....	72

4. RmFFP: Un procedimiento de medición.....	73
4.1. <i>Introducción</i> .....	73
4.2. <i>Diseño de RmFFP</i> .....	74
4.2.1. Definición de objetivos:.....	74
4.2.2. Caracterización del concepto a medir .....	74
4.2.3. Diseño del metamodelo .....	75
4.2.3.1. Reglas de representación .....	77
4.2.3.2. Duplicación de movimiento de datos.....	86
4.2.4. Definición de reglas de asignación numérica .....	90
4.3. <i>Conclusiones</i> .....	91
5. Proceso de la aplicación de RmFFP .....	95
5.1. <i>Introducción</i> .....	95
5.2. <i>El proceso de la aplicación de RmFFP</i> .....	97
5.2.1. Recopilación de la documentación del software.....	98
5.2.2. Construcción del Modelo del Software.....	100
5.2.2.1. Identificar el Propósito de la Medición.....	102
5.2.2.2. Identificar el punto de vista de medición.....	102
5.2.2.3. Identificar el alcance de la medición .....	102
5.2.2.4. Identificar Usuarios .....	102
5.2.2.5. Identificar la Frontera del Software .....	103
5.2.2.6. Identificar Procesos Funcionales .....	103
5.2.2.7. Identificar Grupos de Datos.....	103
5.2.2.8. Identificar atributos de datos .....	103
5.2.2.9. Identificar movimientos de datos.....	103
5.2.3. Aplicación de las reglas de asignación numérica.....	104
5.2.3.1. Eliminar movimientos de datos duplicados.....	104
5.2.3.2. Aplicar función de medición .....	104
5.2.3.3. Aplicar funciones de agregación.....	104

5.3. Aplicando RmFFP en un caso de estudio .....	105
5.4. Conclusiones .....	114
6. Validación del diseño .....	117
6.1. Alcance de la validación del diseño.....	118
6.2. Evaluación de conformidad.....	120
6.2.1. Proceso de evaluación.....	121
6.2.2. Resultado de la evaluación de conformidad .....	122
6.3. Análisis metrológico del diseño .....	124
6.3.1. Un modelo sobre conceptos de metrología .....	124
6.3.2. Análisis metrológico del diseño de RmFFP.....	128
6.3.2.1. Cantidades y unidades .....	130
6.3.2.2. Fundamento de la medición.....	132
6.3.2.3. Estándares de medición .....	133
6.4. Validación teórica.....	133
6.4.1. Trabajos relacionados .....	134
6.4.1.1. Propuesta de Zuse (1998) .....	134
6.4.1.2. Propuesta de Fetcke (2001) .....	136
6.4.1.3. Propuesta de Abrahão (2004) .....	137
6.4.1.4. Propuesta de Poels (2003) .....	137
6.4.1.5. Discusión .....	138
6.4.2. Modelo de Proceso DISTANCE.....	140
6.4.3. Validando teóricamente RmFFP.....	142
6.4.3.1. Encontrar una abstracción de medición .....	144
6.4.3.2. Modelar distancias entre abstracciones de medición.....	145
6.4.3.3. Encontrar una abstracción de referencia .....	146
6.4.3.4. Cuantificar las distancias entre abstracciones de medición.....	147
6.4.3.5. Definir la medida de software.....	147

6.5. Conclusiones.....	148
7. Evaluación de la aplicación.....	151
7.1. Introducción.....	151
7.2. Análisis metrológico de la aplicación de RmFFP.....	153
7.2.1. Proceso de medición.....	153
7.2.2. Estándares de medición.....	156
7.3. Evaluación de la exactitud de RmFFP.....	156
7.3.1. Contexto y estrategia.....	156
7.3.2. Medición con COSMIC-FFP.....	158
7.3.3. Medición con RmFFP.....	159
7.3.3.1. Especificación funcional del sistema SCV.....	160
7.3.3.2. Aplicación de las reglas de representación.....	161
7.3.3.3. Aplicación de las reglas de asignación numérica.....	- 165 -
7.3.4. Evaluación de la exactitud de RmFFP.....	- 165 -
7.4. Precisión y productividad: Un estudio empírico.....	- 167 -
7.4.1. Definición.....	- 167 -
7.4.2. Diseño experimental.....	- 168 -
7.4.2.1. Selección de sujetos.....	- 168 -
7.4.2.2. Objeto experimental.....	- 168 -
7.4.2.3. Selección de variables.....	- 168 -
7.4.2.4. Formulación de hipótesis.....	- 169 -
7.4.2.5. Instrumentos.....	- 170 -
7.4.2.6. Tareas experimentales.....	- 170 -
7.4.3. Ejecución.....	- 171 -
7.4.4. Análisis e Interpretación.....	- 171 -
7.4.4.1. Análisis de la Reproducibilidad de RmFFP.....	- 172 -
7.4.4.2. Análisis de la Productividad con RmFFP.....	- 174 -
7.4.5. Evaluación de la validación.....	- 176 -

7.4.5.1. Validez interna.....	- 177 -
7.4.5.2. Validez externa.....	- 177 -
7.5. Conclusiones.....	- 178 -
8. Adopción en la práctica.....	180
8.1. Descripción general.....	- 182 -
8.2. Adopción de RmFFP en la práctica.....	- 183 -
8.2.1. Definición.....	- 184 -
8.2.2. Diseño.....	- 184 -
8.2.2.1. Selección de sujetos.....	- 184 -
8.2.2.2. Objeto experimental.....	- 185 -
8.2.2.3. Selección de variables.....	- 185 -
8.2.2.4. Formulación de hipótesis.....	- 186 -
8.2.2.5. Instrumento experimental:.....	- 188 -
8.2.2.6. Tareas experimentales.....	- 189 -
8.2.3. Ejecución.....	- 190 -
8.2.4. Análisis e Interpretación.....	- 190 -
8.2.4.1. Evaluación de la facilidad de uso percibida.....	- 191 -
8.2.4.2. Evaluación de la utilidad percibida.....	- 193 -
8.2.4.3. Evaluación de la intención de uso.....	- 194 -
8.2.4.4. Evaluando las relaciones del MAM.....	- 196 -
8.2.5. Evaluación de la validez.....	- 199 -
8.2.5.1. Validez de la conclusión.....	- 199 -
8.2.5.2. Validez interna.....	- 200 -
8.2.5.3. Validez de la construcción.....	- 200 -
8.3. Replicación.....	- 203 -
8.3.1. Evaluación de la validez.....	- 205 -
8.3.2. Análisis e Interpretación de resultados.....	- 208 -
8.4. Discusión.....	- 209 -

9. Conclusiones y trabajos futuros .....	212
9.1. Conclusiones .....	212
9.2. Publicaciones relacionadas .....	214
9.2.1. Revistas Internacionales .....	214
9.2.2. Revistas Iberoamericanas y Nacionales .....	214
9.2.3. Conferencias Internacionales .....	215
9.2.4. Workshops Internacionales e Iberoamericanos .....	216
9.2.5. Conferencias Nacionales .....	217
9.3. Trabajos futuros .....	217
Referencias .....	220
Lista de Figuras .....	231
Listado de Tablas .....	233
Lista de Abreviaturas .....	235
Anexos y Apéndices .....	239





<b>Capítulo</b>	
<i>1</i>	<i>Introducción</i>

*“Door meten tot weten”.*

*- Kammerlingh*

### *1.1. Motivación*

Actualmente, el tamaño del software es un parámetro esencial para la utilización de los diferentes modelos de estimación (productividad, costo, etc.) que apoyan a la administración de proyectos de software, a lo largo de todo su proceso de desarrollo.

Por otro lado, el modelo CMM (Capability Maturity Model) es uno de los modelos con mayor aceptación para entender y mejorar la eficacia del proceso de desarrollo del software. Este modelo se organiza básicamente en cinco niveles de madurez, donde cada nivel representa una etapa evolutiva de madurez organizacional [122]. Como el nivel 2 se preocupa por la buena administración de proyectos, la medición de tamaño funcional desempeña un rol importante en la satisfacción de este nivel de madurez. Consecuentemente, la capacidad de cuantificar con exactitud el tamaño funcional resulta ser crucial, más aún si la medición es llevada a cabo en etapas tempranas, donde su utilidad es mucho mayor que en etapas tardías.

## **1.2. Descripción del problema**

En los recientes años, se ha estado trabajando sobre un método de producción de software, llamado OO-Method [54]. Este método se basa en la transformación de modelos que permiten obtener un producto final de software a partir de un esquema conceptual orientado a objetos. Como un enriquecimiento a OO-Method, se definió un procedimiento de medición de tamaño funcional que afronte principalmente dos problemas:

- La medición poco fiable del tamaño funcional cuando ésta es llevada a cabo en fases tempranas del desarrollo del software, ocasionando, de este modo, un impacto desfavorable en la buena gestión de los proyectos de software.
- La difícil automatización de los métodos estándares de medición de tamaño funcional, debido a la generalidad de sus manuales de medición, aumenta el riesgo de cometer errores a lo largo de un proceso de medición manual o parcialmente automatizado.

Por consiguiente, la presente tesis pretende responder a la siguiente interrogante:

¿Es posible medir de manera fiable y automática el tamaño funcional de una aplicación software a partir de la especificación de requisitos obtenida con OO-Method?

## **1.3. Objetivos**

El objetivo general de la presente tesis es *proveer un procedimiento de medición que facilite medir de manera exacta y precisa el tamaño funcional de una aplicación producida automáticamente con OO-Method a partir de una especificación de requisitos.*

Se entiende por procedimiento de medición un conjunto de operaciones específicas que son ejecutadas en una medición particular de acuerdo a un método dado [1]. Por consiguiente, el presente procedimiento de medición a definir será de acuerdo al método estándar ISO/IEC 19761 (COSMIC-FFP) [22].

El logro de este objetivo dependerá de la satisfacción de los siguientes objetivos específicos:

- Definir un conjunto de reglas que permitan modelar y cuantificar el tamaño funcional de una especificación de requisitos.
- Establecer un proceso que guíe la aplicación del procedimiento de medición propuesto.
- Validar el diseño del procedimiento de medición.
- Evaluar la calidad de los resultados obtenidos al aplicar el procedimiento de medición en casos de estudio.
- Evaluar la adopción del procedimiento propuesto en la práctica en base a las percepciones de utilidad y facilidad de uso.

#### ***1.4. Contexto de investigación***

El presente trabajo de investigación es de carácter interdisciplinario ya que involucra tanto a la Ingeniería de Requisitos como a la Medición del Software, áreas de investigación que han sido de mi interés desde finales de 2000, mientras estudiaba en la universidad Nacional de San Agustín, Arequipa-Perú.

Un trabajo publicado en el 5to workshop de ingeniería de requisitos y ambientes de software [123] me permitió seguir un largo y enriquecedor camino junto al grupo de investigación OO-Method de la Universidad

Politécnica de Valencia, desde diciembre de 2002. Sin embargo, el trabajo desarrollado en esta tesis, se inició a finales de 2003, el mismo que fue soportado por una beca financiada por la empresa CARE Technologies S.A. Además, la participación en contratos y proyectos I+D+I hicieron también posible la culminación de la presente tesis. A continuación estos proyectos y contratos I+D+I son listados:

- Ingeniería de Ambientes Web (TAISSI, Tecnología Software Avanzada para la Ingeniería del Software de la Sociedad de la Información), Proyecto CICYT, con ref. TIC2001-3530-C02-01 , 2002-2004 (Responsable: Oscar Pastor)
- Development of E-services or the New Digital Society (DESTINO), Ministerio de Ciencia y Educación, ref. TIN 2004-03534, 2005-2007 (Responsable: Vicente Pelechano)
- Atenea, Arquitectura Middleware y Herramientas, Ministerio de Industria, Turismo y Comercio, FIT 340503, 2006-2008. (Responsable: Vicente Pelechano)
- Estimación del Tamaño Funcional de Sistemas de Información Orientados a Objeto, 2003 -2004, Convenio I+D con CARE Technologies S.A. (Responsable Silvia Abrahão)
- ISASI (Ingeniería de Software Avanzada para los Sistemas de Información), 2004-2007, Convenio I+D con CARE Technologies (Responsable: Oscar Pastor)

Este conglomerado de convenios de I+D con empresas y proyectos de investigación han generado un interesante contexto de investigación para el desarrollo de esta Tesis, en el que aspectos más orientados a la teoría y

aspectos más orientados a la práctica se han conjugado de manera armoniosa y productiva.

### **1.5. Estructura de la tesis**

La estructura de la presente tesis es organizada en los siguientes capítulos, los cuales son descritos a continuación:

El **capítulo 2** presenta el estado del arte relacionado con medición del software. Un análisis desde la perspectiva de la metrología es llevado a cabo, así como una revisión de los estándares y trabajos relacionados, los cuales son discutidos detalladamente en base a un conjunto de criterios previamente definidos.

El **capítulo 3** describe la fase de Ingeniería de requisitos del proceso de desarrollo de OO-Method, un método que se basa en los principios del paradigma MDA (Model Driven Architecture).

El **capítulo 4** presenta el diseño de un procedimiento de medición de tamaño funcional, llamado RmFFP, el cual es conforme con el método estándar COSMIC-FFP. En primer lugar, un conjunto de reglas fueron definidas con el fin de construir un modelo de software, el cual es definido como el resultado de abstraer las primitivas del modelo de requisitos de acuerdo al metamodelo de COSMIC-FFP. En segundo lugar, con el fin de verificar que el modelo de software ha sido abstraído correctamente, un conjunto de reglas de eliminación de duplicidad de movimientos de datos fue también definido. Finalmente, se describe un conjunto de reglas de medición que permite cuantificar el modelo de software construido.

El **capítulo 5** describe el conjunto de pasos a seguir para aplicar el procedimiento de medición RmFFP; así como su ilustración en un caso de estudio real.

El **capítulo 6** presenta la validación del diseño del procedimiento de medición, la cual comprende de: 1) la evaluación de su conformidad con el método estándar 19761, 2) el análisis basado en el vocabulario de términos básicos de metrología y 3) la validación teórica llevada a cabo en base a un marco formal, llamado DISTANCE.

El **capítulo 7** presenta la evaluación de la aplicación de RmFFP, la cual comprende de: 1) el análisis del proceso de medición 2) verificación de la exactitud y precisión de los resultados obtenidos y 3) evaluación de la productividad que tienen los usuarios al aplicar RmFFP.

El **capítulo 8** complementa la evaluación llevada a cabo en el capítulo anterior, mediante un estudio experimental que permite evaluar las percepciones de los usuarios cuando usan el procedimiento de medición RmFFP y predecir la intención a ser usado en un futuro.

El **capítulo 9** resume las conclusiones y contribuciones del trabajo presentado.

*“Cuando creo que lo tengo todo entre mis manos, miro al frente y me doy cuenta que sólo tomé una porción de arena en una playa de dimensiones infinitas”.*

*- Maria Arce*

La unidad de longitud fue uno de los primeros descubrimientos hechos por el hombre, siendo el mismo cuerpo humano la base que facilitó su creación; tal como es el caso de la pulgada, la mano, el pie, y la yarda. Sin embargo, estas unidades de medida tenían la desventaja de ser variadas entre una población y otra. Frente a esta carencia de uniformidad, que se constituyó en la principal fuente de error y fraude en las transacciones comerciales, surgió la necesidad por estandarizar las medidas, siendo el metro una de las primeras medidas definidas por la asamblea nacional francesa en 1795. Durante la última mitad del siglo diecinueve hubo un incremento en la adopción del *sistema métrico* en diferentes países, y esto originó una carencia de uniformidad en la fabricación de copias para su difusión. Para superar esta dificultad, la Oficina Internacional de Pesos y Medidas (BIPM) fue fundada, el 20 de Mayo de 1875, fecha conocida como el día mundial de la *metrología* [2].

Con el transcurrir de los años, el alcance del BIPM fue extendida a nuevas áreas de tecnología con la definición de nuevos estándares de medición,



tales como los estándares eléctricos (1937), fotométricos (1937), de radiación por ionización (1960), estándares de medición en tiempo (1988) y estándares de medición en la química (2000). De este modo, el Sistema Internacional de Unidades ha estado en continuo proceso de renovación para asegurar una mayor exactitud y precisión de las medidas, y su extensión a otras áreas de tecnología. Tal es el caso de las tecnologías de la información, donde las mediciones llevadas a cabo actualmente son incipientes. A continuación se describe el problema de la medición del software enfocado desde el punto de vista de la metrología.

## ***2.1. Metrología y medición del software***

El término Metrología es definido por la Organización Internacional de Metrología Legal (OIML) [3] en el Vocabulario Internacional de Términos en Metrología (VIM) [1], como la *ciencia de la medición que abarca aspectos teóricos y experimentales en cualquier nivel de incertidumbre y cualquier campo de la ciencia o la tecnología.*

Los campos de aplicación de la metrología son ilimitados, ya que las nuevas áreas de estudio incitan algunas veces a la creación de nuevas unidades de medida e instrumentos de medición. Por consiguiente, la metrología es una rama de la ciencia que está en continua evolución.

El Instituto Nacional de Estándares y Tecnología de los EE.UU (NIST) tiene muchos años de experiencia en medir cantidades físicas, pero en lo que respecta a medición de la “tecnología de la información digital” es relativamente nueva. El NIST identificó un conjunto de cambios, oportunidades y prioridades que surgen al intentar aplicar los conceptos de metrología a la Tecnología de la Información (TI) [4]. Ellos identificaron dos aspectos que distinguen a la “metrología de la tecnología de la información” de la “metrología física”:

Primero.- las cantidades de la TI no son medibles usando un sistema de medición físico tal como es el Sistema Internacional de Unidades. De las siete unidades base solamente el “segundo” aparece como esencial para la TI.

Segundo.- los métodos existentes para calcular expresiones de incertidumbre en la metrología física no pueden ser aplicados fácilmente en la TI. La incertidumbre puede ser mucho más difícil de definir que de cuantificar. Los métodos estadísticos sobre repetibilidad y exactitud en la metrología física no se aplican con claridad en las mediciones llevadas a cabo en las TI.

Muchos años de trabajo han sido dedicados en la metrología física para refinar conceptos como la “trazabilidad”, es decir que el resultado de una medida, no importa dónde se haga, pueda ser relacionado con algún estándar nacional o internacional y que esta relación esté documentada [2]. Actualmente, a pesar de la existencia de un gran número de “métricas” propuestas en la literatura de la ingeniería del software, la capacidad de determinar el nivel de fiabilidad en los resultados de medición del software teniendo en cuenta el concepto de trazabilidad es aún incipiente.

Diferentes clasificaciones de métricas de software fueron propuestas con el fin de mejorar la selección adecuada de métricas para un determinado propósito; entre las taxonomías encontradas tenemos las de Peng y Wallace (1993) [6], Brito y Carapuca (1994) [7], Woodings (1995) [8] y Mendes et al. (2005) [9].

Sin embargo, todas estas métricas aún no han tenido una plena aceptación por los ingenieros del software. Abran et al. afirman lo siguiente: *para que la ingeniería del software sea conocida como una disciplina de ingeniería legítima y una profesión reconocida, un consenso sobre una base del conocimiento es imperativo* [10]. Además ella *necesita ser soportada por medidas, métodos de medición*

*y modelos cuantitativos y descriptivos bien probados* [5]. Conscientes de este problema, una ontología de medición de software ha sido introducida por Garcia et al. [127], con el fin de armonizar los diferentes estándares y modelos de medición existentes en la literatura de la ingeniería del software.

El documento ISO que constituye el consenso internacional sobre la terminología de metrología es el “Vocabulario Internacional de términos básicos y generales en Metrología” (VIM) [1], el cual es ampliamente conocido en las ciencias físicas, químicas y sociales. Sin embargo, este documento es casi desconocido en la comunidad de “métricas de software”, ya que muchos de los conceptos fundamentales de la ciencia de la medición no están siendo utilizados para la definición de métricas e instrumentos de medición. Abran identificó algunas ambigüedades de alto nivel en el dominio de la medición del software y propuso un conjunto de recomendaciones para lograr una evolución de las métricas del software hacia la metrología del software [5]. Gray discutió la aplicabilidad de la Metrología a la TI [12], y concluyó que la medición del software carece de las bases fundamentales en las que se puedan aplicar los principios de la metrología; por ejemplo las líneas de código y los puntos de función tienen problemas con los principios de unidad, escala y medición de incertidumbre.

Una propuesta inicial de Abran y Sellami introduce un modelo de conceptos de medición del Vocabulario ISO de metrología, con el fin de facilitar su presentación y comprensibilidad a la comunidad de ingeniería del software[11]. Posteriormente, un análisis del método estándar de medición COSMIC-FFP [22],[24] y del informe técnico de la ISO 9126-4 [15] es llevado a cabo usando este modelo de conceptos de metrología en [13] y [14] respectivamente. El diseño del método COSMIC-FFP considera la mayoría de conceptos de metrología, por ejemplo, las unidades y cantidades en la definición de reglas de medición y caracterización del concepto a ser medido.

Sin embargo, en lo que respecta al diseño de métricas de calidad en uso que comprende el estándar ISO 9126-4 [15], varias insuficiencias fueron detectadas, tales como:

- Las unidades de cantidades base (calculadas directamente), a excepción del tiempo, y cantidades derivadas (calculadas por alguna ecuación) carecen de símbolo.
- Las unidades de algunas cantidades derivadas son ambiguas ya que dependen sobre otras cantidades con unidades desconocidas;

Recientemente la Organización de Estándares Internacionales, ISO, ha considerado algunos conceptos relacionados a metrología en la introducción de algunos estándares a la comunidad de ingeniería del software. Concretamente, podemos enumerar los siguientes:

ISO/IEC 14143-1: Information technology -Software measurement-Functional size measurement-Part 1: Definition of concepts [16].

ISO/IEC TR 14143-3: Software Engineering – Functional Size Measurement- Part 3: Verification of Functional Size Measurement Methods [18].

ISO/IEC 15939: Information Technology-Software Measurement Process [23].

ISO/IEC 19761: Software Engineering-COSMIC-FFP-A functional Size Measurement Method [22].

A continuación una descripción de los dos primeros estándares (ISO/IEC 14143 e ISO/IEC 15939) y de la ISO 5725 es presentada. Sin embargo, la ISO/IEC 19761 es descrita en la sección 2.2.

### **2.1.1. Estándar ISO/IEC 14143**

El grupo de trabajo N° 12 de la ISO/IEC JTC1/SC7 definió el presente estándar para resolver las inconsistencias generadas por la diversidad de métodos de medición de tamaño funcional, los cuales fueron creados con una falta de consenso en conceptos de medición. El estándar ISO/IEC 14143 provee un marco general en el cual un método de medición de tamaño funcional puede ser definido, probado y verificado bajo un conjunto de propiedades y requisitos mínimos. Este documento ISO está formado por seis partes, de los cuales las partes 3, 4 y 5 de la ISO 14143 han sido publicados como informes técnicos. Es decir, estas tres partes están aun en estudio técnico para que en un futuro puedan ser considerados como estándares internacionales.

#### **2.1.1.1. Parte 1: Definición de conceptos**

La primera parte del estándar identifica las características fundamentales de los métodos de medición de tamaño funcional y define un conjunto de requisitos genéricos para que un método sea considerado como un método de medición de tamaño funcional. El rol de esta parte es promover la interpretación consistente de los principios de medición de tamaño funcional [16].

#### **2.1.1.2. Parte 2: Evaluación de Conformidad**

La segunda parte del estándar establece un marco para evaluar la conformidad de un método de medición de tamaño con los requisitos genéricos definidos en la primera parte de este estándar [17]. La ISO 14143-2 apunta a asegurar que los resultados obtenidos del proceso de evaluación de conformidad sean objetivos, imparciales, consistentes, repetibles, completos y auditables. La evaluación de conformidad es realizada por referencia cruzada

ente cada componente del método candidato y las disposiciones correspondientes de la ISO 14143-1 [16].

Esta parte incluye tres anexos adicionales, el primero provee lineamientos informativos para determinar la competencia del equipo de evaluación de conformidad. El segundo provee un ejemplo de “checklist” para asistir en la evaluación de conformidad de un método candidato. Finalmente, el tercer anexo provee una plantilla para informar los resultados de evaluación de conformidad.

#### ***2.1.1.3. Parte 3: Verificación de un método de medición***

La tercera parte es un informe técnico que provee el proceso y criterios a seguir para que un método de medición de tamaño funcional pueda ser verificado en base a las siguientes propiedades: repetibilidad y reproducibilidad, exactitud, convertibilidad, discriminación del umbral, aplicabilidad a dominios funcionales. La ISO/IEC TR 14143-3 [18] fue desarrollada para asistir en la selección de un método apropiado a sus necesidades verificando que propiedades de rendimiento de un método de medición de tamaño funcional son verdaderas.

#### ***2.1.1.4. Parte 4: Modelo de referencia***

La cuarta parte de la ISO 14143 [19] es un informe técnico que describe un conjunto de Requisitos de Usuario de Referencia (RUR) correspondientes a los dominios de aplicación de negocios y tiempo real, los cuales son presentados en sus Anexos A y B respectivamente. Además un conjunto de características son enumeradas para que un método de medición de tamaño funcional sea considerado como un método de referencia. Estos métodos de referencia en conjunción con los RUR dan lugar a resultados de referencia. En base a estos resultados de referencia, un determinado método

de medición puede ser evaluado comparando sus resultados de tamaño funcional que han sido obtenidos para la misma colección conocida de RUR.

#### ***2.1.1.5. Parte 5: Dominios funcionales***

El presente informe técnico permite evaluar las características de los requisitos funcionales de usuario y categorizarlos en uno o más dominios funcionales. De este modo, los desarrolladores y usuarios de un método de medición de tamaño funcional determinarán con claridad los dominios funcionales donde dicho método puede ser utilizado efectivamente [20]. Ejemplos de evaluación de requisitos funcionales de usuario para determinar el dominio funcional son presentados en una sección de anexos.

#### ***2.1.1.6. Parte 6: Guía de uso de la ISO/IEC 14143 y de los métodos estándares***

La ISO/IEC 14143-6 [21] presenta un resumen de cada una de las partes de la ISO 14143, así como los actuales métodos estándares de medición de tamaño funcional (ISO/IEC 19761 [22], ISO/IEC 20926 [25], ISO/IEC 20968 [26] y la ISO/IEC 24570 [27]). La selección adecuada de un método estándar es guiada por un proceso que también es descrito en esta parte. Además, una guía de cómo usar el tamaño funcional es también proveída.

#### ***2.1.2. Estándar ISO/IEC 15939***

El estándar ISO/IEC 15939 [23] identifica las actividades y tareas que son necesarias para identificar, seleccionar, aplicar y mejorar la medición del software dentro de un proyecto general o estructura de medición organizacional. Este estándar también provee definiciones para términos de medición comúnmente utilizados dentro de la industria del software.

Una actividad se compone de una o más tareas que contribuyen al logro del proceso de medición del software. Una tarea es un segmento bien

definido de trabajo. Este estándar no especifica los detalles de cómo llevar a una tarea incluida en alguna actividad. El proceso de medición de software consta de cuatro actividades que son realizadas de manera iterativa y cíclica. Dichas actividades son: establecer y mantener un comité de medición, planear el proceso de medición, realizar el proceso de medición, y evaluar la medición. La segunda y tercera actividad son consideradas como el corazón del proceso de medición.

### **2.1.3. Estándar ISO 5725**

El estándar ISO 5725, a diferencia de los anteriores estándares descritos, ha sido desarrollado por la comunidad de Metrología con el fin de evaluar la exactitud de los métodos de medición, especialmente en aquellos métodos cuyas medidas pertenecen a una escala continua y dan un solo valor como resultado. La ISO 5725 consta de seis partes, las cuales son enumeradas a continuación:

Parte 1 [28]: Principios generales y definiciones (1994), cuyo propósito es describir de manera general los principios generales para ser entendidos cuando la exactitud de un método de medición es evaluado.

Parte 2 [29]: Método básico para la determinación de la repetibilidad y reproducibilidad de un método de medición estándar (1994). Provee una guía sobre el diseño de pruebas y análisis de resultados para la estimación cuantitativa de la repetibilidad y reproducibilidad de los métodos de medición.

Parte 3 [30]: Medidas intermedias de la precisión de un método de medición estándar (1994). Su objetivo es especificar cuatro medidas intermedias que permiten gestionar los cambios que pueden ocurrir en las condiciones de observación (tiempo, calibración, operador y equipo).

Parte 4 [31]: Métodos básicos para la determinación de la veracidad (trueness) de un método estándar (1994). Provee métodos básicos para



estimar el sesgo (bias) cuando un método de medición es aplicado. La ISO 5725-4 puede ser sólo aplicada si un valor de referencia puede ser establecido como valor verdadero convencional (por ejemplo, estándares de medición, material de referencia, método de medición de referencia).

Parte 5 [32]: Métodos alternativos para la determinación de la precisión de un método estándar (1998). Provee una descripción detallada de alternativas al método básico para la determinación de la repetibilidad y reproducibilidad. La precisión es la proximidad existente entre los resultados de prueba independientemente obtenidos bajo determinadas condiciones. Esta parte del estándar complementa a la ISO 5725-2 [29], la cual puede ser de mucha utilidad para algunas situaciones particulares

Parte 6 [33]: Uso de los valores de exactitud en la práctica (1994). Provee algunas indicaciones sobre como los valores de exactitud pueden ser usados en situaciones prácticas. (por ejemplo, una manera de verificar la aceptabilidad de los resultados probados bajo condiciones de repetibilidad y reproducibilidad).

#### **2.1.4. Modelo de proceso de medición**

Jacquet y Abran definen un modelo de proceso de medición con la finalidad de complementar el estándar ISO/IEC 14143 [16] mediante un conjunto de pasos que guíen las fases de diseño, aplicación, análisis y utilización de un método de medición propuesto [34]. Estos pasos son los siguientes:

1. Diseño del Método de Medición que inicia con la definición de objetivos, paso importante para saber qué es lo que se desea medir, cuál es el punto de vista de medición y cuál es la intención de uso. El siguiente paso es la caracterización del concepto a ser medido, el cual incluye la identificación y

definición del atributo a medir. El tercer paso es el diseño o selección del metamodelo del software, el cual consta de un conjunto de características seleccionadas para describir el software a medir y un conjunto de relaciones. Finalmente, la definición de las reglas de asignación numérica que permiten la cuantificación del software a medir.

2. Aplicación del Método de Medición que inicia con la documentación del software requerido para ser medida. Esta documentación permite la construcción del modelo del software en base al metamodelo definido. Asimismo las reglas de asignación numérica definidas son inmediatamente aplicadas al modelo del software derivado del anterior paso.
3. Análisis de los Resultados del Método de Medición, el cual incluye la presentación de los resultados y la revisión de los mismos.
4. Utilización de los Resultados en diferentes modelos, tales como los modelos de calidad, productividad, estimación, etc.

Teniendo en cuenta las tres primeras fases del Modelo de Proceso de Medición (diseño, aplicación y análisis), los estándares de medición descritos anteriormente podrían ser clasificados en:

- *Estándares de soporte para el diseño de métodos de medición:* la ISO/IEC 14143-1 se constituye en el estándar básico a tener en cuenta para el diseño de nuevos métodos de medición de tamaño funcional. Además el informe técnico ISO/IEC TR 14143-5 puede también

ser tomado en cuenta para determinar los dominios funcionales en el que un nuevo método de medición sería utilizado.

- *Estándares de soporte para la aplicación y análisis de resultados de medición:* El estándar ISO/IEC 15939 da soporte tanto a la aplicación como al análisis de resultados de medición, al proveer un proceso sistemático desde su planeación hasta su evaluación. En cambio, el informe técnico ISO/IEC TR 14143-5 da soporte sólo a la aplicación de la medición porque ayuda a la identificación correcta de los dominios funcionales para una adecuada selección del método de medición a aplicar.
- *Estándares de soporte para la verificación del diseño de métodos de medición:* La ISO/IEC 14143-2 se constituye en un estándar a ser considerado como un soporte para la verificación del diseño del método de medición en términos de su conformidad con la ISO/IEC 14143-1.
- *Estándares de soporte para la verificación de la aplicación de los métodos de medición:* Los informes técnicos ISO/IEC TR 14143-3 y la ISO/IEC TR 14143-4 dan soporte a la verificación de los resultados para la evaluación de un determinado método de medición de tamaño funcional. Asimismo, la ISO 5725 es un estándar que permite evaluar la exactitud de los métodos de medición en general, el cual ha sido aplicado en diferentes disciplinas.

La Tabla 1 resume la clasificación de los estándares anteriormente descritos.

**Tabla 1. Estándares de medición**

<b>Diseño del método de medición</b>	<b>Verificación del diseño</b>
ISO/IEC 14143-1 ISO/IEC TR 14143-5	ISO/IEC 14143-2
Aplicación y análisis de resultados	Verificación de la Aplicación
ISO/IEC 15939 ISO/IEC TR 14143-5	ISO/IEC TR 14143-3 ISO/IEC TR 14143-4 ISO 5725

A continuación, un conjunto de métodos de medición de tamaño funcional conformes con el estándar ISO 14143 son descritos.

## ***2.2. Métodos de medición de tamaño funcional***

Numerosos métodos de medición de tamaño del software han sido propuestos en el pasado, los cuales pueden ser clasificados teniendo en cuenta dos enfoques.

1. Enfoque técnico: formado por aquellos métodos de medición cuyas medidas son derivadas de características técnicas del software, por ejemplo el número de líneas de código [35]. Este enfoque tiene la desventaja de usar medidas tardías y poco significativas para el nivel administrativo de la organización debido a su alta dependencia con la tecnología de desarrollo (lenguaje de programación, plataforma, etc.). En la literatura son también conocidas como mediciones ‘a posteriori’ [35] o mediciones sintácticas.
2. Enfoque funcional: formado por aquellos métodos de medición cuyas medidas son derivadas a partir de la funcionalidad requerida por los usuarios. Este enfoque tiene

la ventaja de permitir una medición temprana del tamaño del producto final, de ser independientes a la tecnología de desarrollo y por ende fáciles de ser entendidas por el nivel administrativo. En la literatura son también conocidas como mediciones ‘a priori’ [36] o mediciones semánticas.

El primer método basado en un enfoque funcional, fue el Análisis de Puntos de Función (APF) desarrollado por Albrecht [37]. Este método ha ido obteniendo una amplia popularidad en la industria del software, pero la falta de aplicabilidad para todos los tipos de software y la rápida evolución de los paradigmas de desarrollo de software ha ido ocasionando algunas variaciones de este método de medición, entre las que destacamos: el método “Puntos Característica” (Feature Points) propuesto por Capers Jones [38], cuyo propósito fue extender el método APF para poder ser aplicados a algoritmos científicos. Whitmire desarrolló el método “3D Puntos de Función” para poder medir aplicaciones de negocios y sistemas de tiempo real [39]. “Full Function Point” fue desarrollado por St-Pirre et al. [40] con el fin de ampliar el dominio funcional del tradicional método APF. La Asociación de Métricas de Software de Holanda publicó una variante del método APF para simplificar algunas de las reglas de medición [41].

A pesar de existir un número de variantes de APF, sólo cuatro métodos de medición de tamaño funcional fueron confirmados como estándares internacionales: COSMIC-FFP (ISO/IEC 19761)[22], IFPUG 4.1 (ISO/IEC 20926)[25], Mk II FPA (ISO/IEC 20968)[26] y NESMA FPA (ISO/IEC 24570)[27]. A continuación cada uno de estos métodos estándares es introducido brevemente.

### 2.2.1. ISO/IEC 19761: COSMIC-FFP (2003)

En 1997, St-Pierre et al. [40] propusieron un método que permitiera medir el tamaño funcional de las aplicaciones de tiempo real. Este método fue llamado Full Function Points (FFP). En 1999, el Consorcio Internacional de Medición de Software Común, liderados por Alain Abran y Charles Symons, publican la versión 2.0 del Manual de Medición COSMIC-FFP [44], como el primer método de medición de tamaño funcional de *segunda generación*. Esta generación se caracterizó por la extensión del alcance del punto de vista de medición. Es decir, COSMIC-FFP, a diferencia de los métodos tradicionales, no sólo mide la funcionalidad desde el punto de vista del usuario final (persona que interactúa con el sistema), sino que, al ampliar el concepto de usuario, la medición del sistema es realizada desde diferentes puntos de vista. De este modo, toda la funcionalidad es cubierta, la cual era muchas veces parcialmente invisible por los métodos tradicionales. En Mayo de 2001, algunos cambios son presentados en la versión 2.1 para mejorar las definiciones de ciertos conceptos básicos de COSMIC-FFP [45]. En Enero de 2003, COSMIC publicó la versión 2.2 del manual de medición [24], la cual fue reconocida como un método estándar a través de la norma ISO/IEC 19761 [22].

El modelo general de software COSMIC-FFP distingue cuatro tipos de movimientos de datos: entrada, salida, lectura y escritura. Un movimiento de dato transfiere uno o más atributos de datos que pertenecen a un único grupo de dato. Así mismo, COSMIC-FFP establece como principio de medición que el tamaño funcional del software es directamente proporcional al número de movimientos de datos.

### 2.2.2. ISO/IEC 20926: IFPUG 4.1 (2003)

IFPUF 4.1 está basado sobre el método originalmente propuesto por Albrecht a mediados de 1970, el cual fue por primera vez publicado en 1979

[37]. En 1984, Albrecht refinó el método, pero desde 1986, año en que el Grupo Internacional de Usuarios de Puntos de Función (IFPUG) fue formado, varias versiones han sido publicadas por IFPUG. Hasta que en el 2003, el Manual IFPUG 4.1 [43] fue reconocido como un estándar internacional por medio de la ISO/IEC 20926 [25], el cual provee una clara y detallada documentación del método de puntos de función. Cabe mencionar que este reconocimiento de IFPUG 4.1 como estándar no incluye a la parte de ajuste de puntos de función, ya que este ajuste no reúne los requisitos necesarios de la ISO/IEC 14143-1. El método IFPUG 4.1 asume un modelo de software que consta de dos partes: Primero, el *Modelo de Datos* que representa a los archivos lógicos internos y externos. Segundo, El *Modelo de Transacciones* que representa las operaciones que el usuario desea hacer con los datos. Estos pueden ser de entrada, salida y consulta.

### 2.2.3. ISO/IEC 20968: Mk II FPA (2002)

Mk II FPA surge en 1988 gracias a Charles Symons como una nueva variante del método precursor Análisis de Puntos de Función. Este método asume un modelo de software en el cual la funcionalidad es expresada en términos de *Transacciones Lógicas*, donde cada transacción consta de tres tipos de componentes: Entradas, Salidas y Referencias de Entidades (referencia a su manipulación de datos almacenados). El tamaño de los componentes tipo Entrada y Salida es proporcional al número de Tipos de Elementos de Datos (DETs) del componente, mientras que el tamaño del componente Referencias a Entidades es proporcional al número de Tipos de Entidades que son referenciadas en el procesamiento. Luego el tamaño de una transacción lógica es la suma de los tamaños de los tres tipos de componentes que fueron previamente multiplicados por sus correspondientes pesos. MkII FPA ha sido desarrollado a través de tres versiones, siendo MkII FPA 1.3.1 [42] la última versión que ha sido conforme al estándar ISO 14143-1 [16] y por ende

reconocida como el primer método estándar a finales del 2002. La ISO/IEC 20968 [26] especifica el conjunto de definiciones, convenciones y actividades del método de medición de tamaño funcional MK II FPA. El método no considera reglas para medir complejos algoritmos ni requisitos de tiempo real. MkII FPA es ampliamente utilizado en el Reino Unido y es mantenido por la Asociación de Métricas de este país, llamado UKSMA.

#### **2.2.4. ISO/IEC 24570: NESMA FPA (2004)**

NESMA publicó la primera versión de su Manual Definición y Guías de Conteo para la aplicación de Análisis de Puntos de Función (CPM 1.0) en 1990. Este manual asumió los principios del método IFPUG FPA 3.1, pero dio pautas más concretas para que pudieran ser aplicadas a modelos lógicos, es decir antes de que una aplicación fuera implementada. La versión 2.0 de NESMA fue lanzada en 1996 [41] y se constituyó en un complemento al manual de IFPUG FPA. En 2002, NESMA FPA 2.0 inicia un proceso de conformidad al estándar ISO 14143-1, resultando ser conforme y considerado como un método de medición de tamaño funcional. Finalmente, en 2004 NESMA FPA es reconocida como un estándar internacional a través de la ISO/IEC 24570 [27].

La Figura 1 resume la evolución de estos cuatro métodos estándares.

### **2.3. Procedimientos de medición de tamaño funcional**

Un método de medición se distingue básicamente de un procedimiento de medición por su genericidad. Sin embargo, en la mayoría de los trabajos revisados, ambos términos son utilizados indistintamente, a pesar de que sus definiciones son claramente explicadas en el VIM [1].



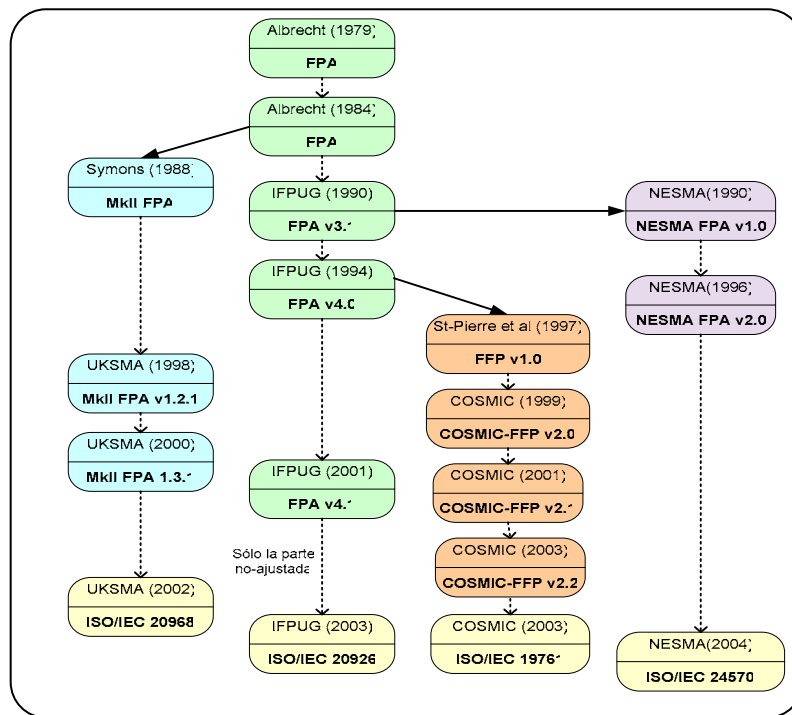


Figura 1. Evolución de los métodos estándares

En la presente sección, un análisis sobre los procedimientos de medición encontrados en la literatura ha sido llevado a cabo en base a un conjunto de criterios, los cuales fueron adecuados de Lothar [46].

- **Método de medición:** comprende del nombre y versión del Método utilizado para la definición del procedimiento de medición.
- **Contexto de desarrollo:** Es el contexto en el que el método de medición seleccionado es instanciado. Comprende del nombre del contexto de desarrollo de software (método, proceso).
- **Dominios funcionales:** Comprende los diferentes dominios funcionales en los que el Procedimiento de Medición puede ser aplicado.

- **Artefacto requerido:** Se precisa el objeto o conjunto de objetos utilizados como entrada para el proceso de medición. Ejemplo: Diagrama de Casos de Uso, Diagrama de Secuencias, etc.
- **Reglas de representación:** Se resume el conjunto de reglas definidas para la operacionalización del procedimiento de medición.
- **Instrumento software de medición:** Se verifica si el procedimiento de medición es automatizado. En caso de serlo, se precisa el nombre, versión y plataforma en la que es soportada.
- **Verificación del procedimiento:** Descripción del procedimiento utilizado para verificar el procedimiento definido. Puede ser proyectos piloto, experimentos, etc.

A continuación 2 grupos de procedimientos de medición son presentados. El primer grupo está conformado por procedimientos de medición que se basan en métodos estándares tradicionales de tamaño funcional (IFPUG FPA, Mark II FPA y NESMA FPA), mientras que el segundo grupo está conformado por procedimientos de medición que se basan en el método estándar COSMIC-FFP, llamado también método de segunda generación, tal como se explicó en la sub-sección 2.4.

### 2.3.1. *Primera generación*

Entre los trabajos encontrados tenemos a Fetcke et al. [47], Tavares et al. [48], Bertolami et al. [49], Lehne [56], Uemura et al. [55] y Abrahão [53]. Cada una de estas propuestas es descrita a continuación.

#### 2.3.1.1. *Propuesta de Fetcke (1998)*

Fetcke [47] precisa un conjunto de reglas basadas sobre el método de medición IFPUG FPA [50] para el método de desarrollo OOSE [51]. Resumiendo las características más destacadas de esta propuesta son:

***Método de medición:*** IFPUG FPA v 4.0 [50].

***Contexto de desarrollo:*** Método OOSE.

***Dominios funcionales:*** Sistemas de Información de Gestión.

***Artefacto requerido:*** Modelo de Casos de Uso, Modelo de Objetos

***Reglas de representación:*** El diagrama de casos de uso es requerido para identificar la Frontera del sistema, el cual es comprendido por el conjunto de casos de uso que interactúan con los actores de dicho diagrama (usuarios y aplicaciones externas). Los Objetos tipo Entidad del Modelo de Objetos son identificados como ILFs (Internal Logic File). Además, si los casos de uso hacen uso implícito de archivos lógicos que no son representados en el Modelo de Objetos (Ejemplo: Mensajes de error, texto de ayuda, etc.) estos archivos son también considerados como ILFs. En cuanto a las relaciones de agregación y herencia se tiene que todo objeto que es parte de otro objeto es representado como un RET (Record Element Type). Asimismo, todo Objeto Abstracto es también representado como un RET. Los atributos de los objetos tipo entidad son identificados como DET (Data Element Type). Cada objeto mantenido o leído por un caso de uso es considerado como un FTR (File Type Referenced). En cuanto al concepto de Transacciones, los casos de uso son

candidatos para identificar uno o más transacciones. Finalmente, Fetcke no ajusta el tamaño funcional obtenido a pesar de usar el método IFPUG 4.0.

***Instrumento software de medición:*** No

***Verificación del procedimiento:*** Es aplicado en tres proyectos de la industria y comparado con otras aproximaciones.

### ***2.3.1.2. Propuesta de Tavares (1999)***

Tavares [48] realiza un trabajo muy similar a la propuesta de Fetcke [47], define un conjunto de reglas de representación entre los conceptos del método de medición de tamaño funcional IFPUG FPA y las primitivas del Lenguaje de Modelado Unificado. Las características generales de esta propuesta son:

***Método de medición:*** IFPUG FPA v 4.1 [43]

***Contexto de desarrollo:*** Lenguaje de Modelado Unificado (UML).

***Dominios funcionales:*** Sistemas de Información de Gestión.

***Artefacto requerido:*** Modelo de Casos de Uso y Diagrama de Clases

***Reglas de representación:*** El diagrama de casos de uso es requerido para identificar la Frontera del sistema. El diagrama de Clases es fuente principal de información para la identificación de las funciones de datos. Así, cada clase es un ILFs y sus atributos son candidatos a DETs. En cuanto a la relación de Asociación, tanto la clase que asocia y la clase asociada son consideradas simplemente como ILFs. En cambio, las relaciones de agregación y herencia del diagrama de clases aportan información para la identificación de RETs. Así una clase que es agregada en otra y una clase abstracta son candidatos a RETs. Las subclasses de una clase concreta son candidatas a ILFs o RETs. Al igual que la propuesta de Fetcke, los archivos lógicos que no son

representados en el Modelo de Objetos pero si en los casos de uso (Ejemplo: Mensajes de error, texto de ayuda, etc.) son también considerados como ILFs. Todo caso de uso que mantiene relación directa con algún actor y todo caso de uso que extiende a otro caso de uso son candidatos como Funciones Transaccionales, las cuales son de tres tipos: Entradas Externas (EI), Salidas Externas (EO) y Consultas Externas (EQ). La identificación de estos tipos se realiza a partir de la descripción de alto nivel de detalle de los casos de uso.

***Instrumento software de medición:*** No

***Verificación del procedimiento:*** Realiza el seguimiento de un proyecto real desarrollado en la Empresa de Tecnologías de Información SERPRO (Brasil) a lo largo de todo su ciclo de vida. De esta manera evalúa la exactitud de la medición comparando el tamaño estimado con el tamaño real de dicho proyecto.

### ***2.3.1.3. Propuesta de Bertolami (2003)***

Bertolami [49] propone medir el tamaño funcional en la etapa de elicitación de requisitos, mediante la medición de escenarios. Para esto, se define un conjunto de reglas de medición basados sobre el método de medición Mark II FPA [42]. A continuación, se presenta la información general de esta propuesta.

***Método de Medición:*** Mark II FPA v 1.3.1 [42].

***Contexto de desarrollo:*** Léxico Extendido del Lenguaje (LEL) [52], el cual es una representación hiper-textual de los símbolos del lenguaje del cliente en el contexto de la aplicación. Los escenarios que son construidos a partir del LEL representan descripciones parciales del comportamiento del sistema en un momento específico de la aplicación. Un escenario se compone

de: Recursos, Actores, Episodios (serie de acciones que detallan al escenario), Excepciones y Restricciones.

***Dominios funcionales:*** Sistemas de Información de Gestión.

***Artefacto requerido:*** Escenarios contruidos a partir del LEL.

***Reglas de representación:*** Dado que los escenarios representan las funciones del sistema, el alcance es comprendido por todos los escenarios descritos. Los usuarios son representados por los correspondientes actores de cada escenario. Un escenario es descompuesto en episodios, y cada episodio está compuesto por componentes tipo Entrada (E), Proceso (P) y Salida(S). Estos componentes E-P-S de un episodio contienen una serie de objetos los cuales son denominados Recursos (Entidades). El concepto de Tipos de *Elementos de Dato* (DET) es identificado en los componentes de tipo Entrada y Salida. El concepto de *Referencia a Entidades* es identificado en el componente de tipo Proceso. Por lo tanto, el tamaño funcional de cada episodio resulta de sumar el número total de DETs más el número total de entidades referenciadas. El tamaño del sistema será el tamaño funcional del conjunto de episodios identificados.

***Instrumento software de medición:*** No

***Verificación del procedimiento:*** Aplicado a cuatro casos estudio.

#### ***2.3.1.4. Propuesta de Lehne (1997)***

Lehne [56] llevó a cabo la medición de aplicaciones orientado a objetos que son modelados con el método OOram [57], un método basado en roles para las fases de análisis y diseño.

***Método de Medición:*** IFPUG FPA.

**Contexto de desarrollo:** método de desarrollo OOram que incluye diferentes vistas, tales como: vista de estímulo-respuesta, vista de relaciones, vista de colaboración, vista de interfaz, vista de escenario, vista de procesos, vista de diagramas de transición de estado y vista de especificación de métodos.

**Dominios funcionales:** Sistemas de Información de Gestión.

**Artefacto requerido:** únicamente el diagrama de colaboración es mencionado de manera explícita, a pesar de que otras primitivas son utilizadas para la identificación de funciones transaccionales, las cuales no son mencionadas con claridad.

**Reglas de representación:** La correspondencia definida entre las primitivas de OOram y los conceptos de IFPUG FPA es llevada a cabo de la siguiente manera: Los objetos/roles son identificados como FTRs, la información de los objetos/roles expresada en los diagramas de colaboración es identificada como ILFs, los EIFs no son considerados en la correspondencia con las primitivas de OOram. En cuanto a las funciones transaccionales (EIs, EOs y EQs) son identificadas en los mensajes de entrada y salida de la aplicación, no se menciona alguna primitiva de modelado.

**Instrumento software de medición:** No

**Verificación del procedimiento:** No es descrito.

#### **2.3.1.5. Propuesta de Uemura et al. (1999)**

Uemura et al. [55] definieron un conjunto de reglas para medir especificaciones a nivel de diseño. A continuación, se presenta la información general de esta propuesta.

**Método de medición:** IFPUG FPA 4.0 [50]

**Contexto de desarrollo:** UML versión 1.1

**Dominios funcionales:** Sistemas de Información de Gestión.

**Artefacto requerido:** diagramas de clases y diagramas de secuencia.

**Reglas de representación:** La frontera del software a medir es determinada por el conjunto de diagramas de secuencia. Los objetos con operaciones que cambian los atributos de otros objetos son considerados como ILFs, el resto de objetos son considerados como EIFs. Los RETs no pudieron ser identificados en los diagramas de clases ni diagramas de secuencia. Las funciones transaccionales son identificadas en los mensajes de los diagramas de secuencia.

**Instrumento software de medición:** Se implementó un instrumento de medición que tiene como entrada (artefacto a medir) las especificaciones elaboradas con la herramienta Rational Rose.

**Verificación del procedimiento:** Un estudio fue llevado a cabo con el fin de comparar los resultados obtenidos automáticamente con aquellos obtenidos por especialista en puntos de función de manera manual. Los autores, al observar que había una diferencia en cuanto al número de funciones transaccionales, propusieron un conjunto de condiciones en los diagramas de UML para ajustar estas diferencias. Esto fue verificado exitosamente en un segundo estudio [74].

#### **2.3.1.6. Propuesta de Abrahão et al. (2004)**

Abrahão propuso un procedimiento de medición, llamado OO-Method Function Points (OOmFP) [53]. Este procedimiento mide el de tamaño funcional de los esquemas conceptuales, modelados en la fase de análisis de OO-Method [54]. Las características generales de esta propuesta son:



**Método de medición:** IFPUG FPA 4.1 [43].

**Contexto de desarrollo:** Método de desarrollo OO-Method [54].

**Dominios funcionales:** Sistemas de Información de Gestión.

**Artefacto requerido:** Modelo conceptual de OO-Method, el cual comprende de cuatro modelos: modelo de objetos, modelo dinámico, modelo funcional y modelo de presentación.

**Reglas de representación:** En cuanto a las funciones de datos, las clases del modelo de objetos son consideradas como ILFs y las vistas legadas (filtros sobre alguna clase de un sistema pre-existente) consideradas como EIFs. En cuanto a las funciones transaccionales, los servicios de una clase o vista legada son considerados como EIs, los patrones de presentación, definido en el modelo de presentación, pueden ser considerados como EOs o EQs.

**Instrumento software de medición:** Se automatizó el procedimiento de medición propuesto, siendo incluido como un módulo de la herramienta Olivenova [119].

**Verificación del procedimiento:** Abrahão verificó parcialmente el diseño del procedimiento desde la perspectiva de la teoría de la medición. Un estudio experimental fue también llevado a cabo para verificar su reproducibilidad y exactitud con respecto a IFPUG FPA 4.1 [43]. Los resultados obtenidos muestran que OOmFP es reproducible y exacto. Además una aplicación del Modelo de Evaluación de Métodos (MEM)[126], [75] fue llevada a cabo con el fin de evaluar un conjunto de variables basadas en la percepción de eficiencia y efectividad del método.

### 2.3.2. Segunda generación

Entre los trabajos revisados tenemos a Bevo et al. [58], Jenner [59], Azzouz [60], Habela et al. [61], Poels [62], Diab et al. [63] y Nagano et al. [64],

#### 2.3.2.1. Propuesta de Bevo (1999)

Bévo et al. [58] relacionan las diferentes primitivas de UML con los conceptos del modelo de COSMIC FFP 2.0 [44]. A continuación, se presenta una descripción general de esta propuesta.

**Método de medición:** COSMIC-FFP v 2.0 [44].

**Contexto de desarrollo:** Lenguaje de Modelado Unificado UML 2.0.

**Dominios funcionales:** No se precisa

**Artefacto requerido:** Modelo de Casos de Uso y Diagrama de Clases UML.

**Reglas de representación:** La frontera del sistema a medir es comprendida por el Diagrama de Casos de Uso. Cada Caso de Uso es correspondido a un Proceso Funcional en COSMIC-FFP. Los movimientos de datos están representados por el concepto de Escenario, definido como una secuencia de interacciones representativas de un caso de uso. El concepto de Evento Desencadenador no tiene una representación directa en algún concepto de UML. Cada clase del diagrama de clases se constituye en un Grupo de Datos. Los atributos de dicha clase vienen a ser Atributos de Datos. Un Actor UML, tiene correspondencia directa con el concepto de Usuario en COSMIC-FFP. El concepto de Capa no tiene correspondencia directa con algún concepto en UML. Finalmente el tamaño funcional es igual a la suma de los movimientos identificados.

**Instrumento software de medición:** una herramienta llamada Metric Xpert fue implementada [73].

**Verificación del procedimiento:** La propuesta fue verificada con respecto a su exactitud. Para esto, 5 casos de estudio fueron medidos con la herramienta Metric Xpert y comparados con las respectivas medidas obtenidas por los expertos manualmente [73]. Como resultado de esta comparación, las diferencias encontradas para estos casos de estudio oscilaron entre 11% y 33%.

### 2.3.2.2. Propuesta de Jenner (2001)

La propuesta de Jenner [59] surge como resultado de discutir el aspecto de granularidad para los casos de uso sobre el trabajo realizado por Bevo [58]. Por consiguiente, las características generales de dicha propuesta son muy similares a la anterior propuesta.

**Método de medición:** COSMIC-FFP v 2.0 [44].

**Contexto de desarrollo:** Lenguaje de Modelado Unificado UML.

**Dominios funcionales:** No se precisa

**Artefacto requerido:** Diagrama de Casos de Uso, Diagrama de Secuencias y Diagrama de Clases.

**Reglas de representación:** Básicamente cada proceso funcional es representado por un diagrama de secuencia. Jenner considera que estos diagramas representan el nivel de abstracción más adecuado que los casos de uso. Los movimientos de datos son representados por los mensajes de interacción de un diagrama de secuencia. Asimismo, Jenner menciona que el concepto de Capa de COSMIC-FFP puede ser representado por el concepto de UML “Swimlanes”, el cual es definido como la organización de paquetes a un mismo nivel de abstracción.

**Instrumento software de medición:** El procedimiento de medición tiene soporte automático [66].

**Verificación del procedimiento:** Casos de estudio.

### 2.3.2.3. Propuesta de Azzouz (2003)

Azzouz [60] presenta una herramienta para la automatización del proceso de medición del tamaño funcional para el Proceso Unificado de desarrollo de software RUP (Rational Unified Process). El diseño de esta herramienta de medición está basado en la instanciación del método COSMIC-FFP.

**Método de medición:** COSMIC-FFP v 2.2 [24].

**Contexto de desarrollo:** Proceso Unificado de desarrollo de software (RUP).

**Dominios funcionales:** Sistemas de Información de Gestión.

**Artefacto requerido:** Modelo de Casos de Uso y Diagrama de Escenarios (UML)

**Reglas de representación:** Azzouz toma en cuenta el conjunto de reglas de representación propuesto por Bevo [58] y Jenner [59]. Sin embargo, a diferencia de la propuesta de Jenner, Azzouz determina que no existe una correspondencia directa del concepto de “capa” con algún concepto de UML. Por esta razón, este concepto es identificado manualmente. Además, en cuanto al concepto “evento disparador”, al no tener ninguna representación en UML, un nuevo estereotipo en los diagramas de casos de uso es creado para poder distinguir explícitamente este concepto. Azzouz, precisa la medición en tres niveles: para el nivel de Análisis de Requisitos, el artefacto requerido es el modelo de casos de uso. En el nivel de Análisis, el diagrama de escenarios es usado como artefacto de entrada. Finalmente en el nivel de Análisis/diseño,

los diagramas de escenario con mayor nivel de detalle son usados como artefactos para calcular el tamaño funcional.

**Instrumento software de medición:** La herramienta de medición de tamaño funcional ha sido integrada en el entorno de la herramienta Rational Rose [60].

**Verificación del procedimiento:** El caso de estudio “Rice Cooker” fue usado para verificar la herramienta implementada.

#### 2.3.2.4. Propuesta de Habela et al. (2005)

Habela et al. [61] adaptan el modelo de casos de uso desde una perspectiva orientado al diseño para aplicar las reglas descritas en el manual de COSMIC-FFP 2.2. Entre las principales características a destacar de esta propuesta son las siguientes:

**Método de medición:** COSMIC-FFP v 2.2 [24]

**Contexto de desarrollo:** UML versión 1.5 [65].

**Dominios funcionales:** Sistemas de Información de Gestión.

**Artefacto requerido:** Modelo de Casos de Uso: Especificación detallada de casos de uso. Para esto, una plantilla de especificación fue diseñada, la cual incluye referencias a catálogos de reglas de negocios, pre-condiciones, post-condiciones y una descripción en pasos de los escenarios principales y alternativos.

**Reglas de representación:** Habela asume que un caso de uso consiste de uno o más procesos funcionales. Los diferentes tipos de movimientos de datos son identificados en cada paso de los escenarios. Un paso especifica un conjunto de atributos de datos que son movidos. De esta forma, un escenario no puede estar limitado sólo a la descripción de

comunicaciones externas; sino que también a la descripción explícita de las lecturas y escrituras de datos persistentes. Las relaciones entre casos de uso (Inclusión, Extensión, y Generalización) son tomadas en cuenta para evitar redundancias en la medición. Esta medición es llevada a cabo a un nivel atributos de datos por requerir de una mayor precisión del tamaño funcional obtenido.

***Instrumento software de medición:*** El proceso de medición es llevado a cabo manualmente.

***Verificación del procedimiento:*** No se ha encontrado aún algún estudio de verificación o validación

#### ***2.3.2.5. Propuesta de Poels (2003)***

Poels [62] define un conjunto de reglas de representación para el método de desarrollo MERODE (Model Driven Existence-dependency Related Object Oriented Development) [67]. A continuación se resume las principales características de esta propuesta:

***Método de medición:*** COSMIC-FFP v 2.2 [24].

***Contexto de desarrollo:*** método de desarrollo MERODE [67], el cual comprende de un modelo conceptual organizado en tres capas jerárquicas. La capa más baja está constituida por el Modelo de Negocios, el cual es representado por el diagrama de clases (notación UML), Tabla Objeto-Evento y los Diagramas de Transición de Estado para modelar el comportamiento del sistema. La capa intermedia está constituida por el Modelo de Servicios, el cual especifica las facilidades del usuario final para generar eventos y ser transmitidos al Modelo de Negocios. El modelo de Servicios comprende de tres tipos de objetos no persistentes (objetos de entrada, salida y control). Finalmente la capa más alta está constituida por el

Modelo de Interfaz de Usuario que captura aspectos de presentación y dependen sobre las preferencias del usuario final

***Dominios funcionales:*** Sistemas de Información de Gestión.

***Artefacto requerido:*** Modelo de Negocios y Modelo de servicios.

***Reglas de representación:*** Poels realiza la representación de conceptos para las capas del Modelo de Negocios y Modelo de Servicios. Para la capa correspondiente al Modelo de Negocios, el concepto de usuario es constituido por los objetos de la capa intermedia del Modelo de Servicios. La frontera es el borde comprendido entre toda la capa del Modelo de Negocios y los usuarios. Un proceso funcional es representado por el conjunto de métodos de las clases que son activadas por algún mismo evento de la capa de Servicios. Un movimiento de dato es representado por cada método de clase que pertenece a un proceso funcional. Todos los tipos de movimientos de datos tienen una representación en MERODE a excepción del movimiento de dato tipo salida. Los grupos de datos son representados por cada una de las clases del Modelo de Negocios. En cuanto a la capa del Modelo de Servicios, los usuarios vienen a ser el conjunto de objetos que pertenecen a la capa del Modelo de Interfaz de Usuario. La frontera es el borde comprendido entre toda la capa del Modelo de Servicios y los usuarios. Los Procesos Funcionales son representados por el conjunto de objetos no persistentes que ofrecen una funcionalidad de entrada, salida y control. En esta capa, todos los tipos de movimientos de datos tienen una representación; por ejemplo, los movimientos de dato tipo entrada son representados en los métodos de los objetos de entrada, salida y control. Todos ellos pueden ser invocados por algún mensaje requerido o por algún mensaje de la capa del Modelo de Negocios.

**Instrumento software de medición:** El proceso de medición es manual, no se ha encontrado la automatización de las reglas definidas.

**Verificación del procedimiento:** Es validado desde el punto de vista de la teoría de la medición [68] utilizando el marco formal llamado DISTANCE [69]. Sin embargo, no hay resultados sobre evidencia empírica para métodos de tamaño funcional.

#### 2.3.2.6. Propuesta de Diab et al. (2004)

Diab [63] definió un conjunto de reglas de representación y medición de tamaño funcional para los sistemas de tiempo de real, los cuales en un inicio eran modelados con el lenguaje ROOM (Real Time Object Oriented Modeling) [70]. Este lenguaje fue soportado por la herramienta Object Time que posteriormente fue adquirido por Rational. Con la evolución de Object Time surge la herramienta RRRT (Rational Rose Real Time) para diseñar sistemas distribuidos y de tiempo real. A continuación, se describe las características principales de esta propuesta.

**Método de medición:** COSMIC-FFP v 2.2 [24].

**Contexto de desarrollo:** Un modelo RRRT puede ser observado desde dos perspectivas: estructura y comportamiento. La estructura de un modelo RRRT es comprendido por tres tipos de entidades: cápsulas, protocolos, y clases de datos. Una cápsula es una entidad activa que tiene aspectos estáticos y dinámicos. Un protocolo es un conjunto de mensajes. Y un mensaje corresponde a una petición de un servicio. Finalmente una clase de datos es la unidad básica para la representación de datos. El comportamiento de un modelo RRRT especifica los aspectos dinámicos de cada cápsula usando diagramas de transición de estado. Una transición de estado puede estar definido por acciones, sub-estados y transiciones.



**Dominios funcionales:** Sistemas de Tiempo Real

**Artefacto requerido:** modelo RRRT.

**Reglas de representación:** El concepto de frontera es representado por la demarcación conceptual de un conjunto de cápsulas. Este conjunto de cápsulas representan al concepto de Capa. Los procesos funcionales son representados por una o varias transiciones. Los tipos de movimientos de datos son representados en los mensajes y acciones. Los grupos de datos son representados por las clases de datos. Finalmente, el concepto de atributos de datos es representado por el conjunto de atributos elementales.

**Instrumento software de medición:** Este procedimiento de medición de tamaño funcional ha sido automatizado por una herramienta llamada  $\mu$ ROSE [63].

**Verificación del procedimiento:** las reglas definidas han sido validadas dentro del grupo de expertos del grupo COSMIC. Así mismo se ha aplicado a varios sistemas y comparado con los resultados obtenidos por varios expertos. Además, la automatización de las reglas ha facilitado la verificación de la repetibilidad y exactitud del procedimiento.

#### 2.3.2.7. Propuesta de Nagano (2003 )

Al igual que la propuesta de Diab, Nagano propone un procedimiento de medición para las aplicaciones de tiempo real [64]. Las características más importantes de esta propuesta son las siguientes:

**Método de medición:** COSMIC-FFP v 2.0 [44].

**Contexto de desarrollo:** método Shlaer-Mellor [71], un método orientado a objetos cuyos productos son especificados en xUML [72].

**Dominios funcionales:** Sistemas de tiempo real

**Artefacto requerido:** Diagramas de clase, de estado y de colaboración.

**Reglas de representación:** El conjunto de atributos y las relaciones entre objetos, especificados en los diagramas de clase, son considerados como Grupos de Datos candidatos. Los parámetros de mensajes y las señales de control, especificados en los diagramas de estado, son considerados también como Grupos de Datos candidatos. Nagano indica que si un grupo candidato es reconocido por los usuarios, entonces es representado como un grupo de dato. Para identificar el concepto de procesos funcionales, Nagano primero identifica a los eventos disparadores en los diagramas de colaboración. A continuación, la secuencia de movimientos de datos identificados a partir de los diagramas de estado para cada clase da lugar a los procesos funcionales.

**Instrumento software de medición:** No

**Verificación del procedimiento:** Nagano llevó a cabo sólo un estudio inicial para evaluar el grado de facilidad de uso de COSMIC-FFP en función a la productividad de los sujetos.

## **2.4. Discusión**

En la presente sección, un análisis comparativo es llevado a cabo entre todos los procedimientos de medición anteriormente descritos. En términos generales, cada una de estas propuestas instancian los conceptos generales de un *método de medición* en un determinado *contexto particular* y en uno o varios *dominios funcionales*. Además, esta medición es llevada a cabo sobre ciertos artefactos de software que pueden ser obtenidos en determinadas *fases de desarrollo*. Así mismo, tanto la *verificación* de la calidad de un procedimiento de medición como su *automatización* fueron también

considerados como criterios para llevar a cabo el presente análisis comparativo. A continuación este análisis es descrito.

**Con respecto al método de medición** seleccionado, se puede observar que tanto IFPUG FPA y COSMIC-FFP han tenido una mayor aceptación por la industria y la comunidad científica. Sin embargo, la rápida aprobación de COSMIC-FFP como método estándar en comparación con los otros métodos, la aplicabilidad a diferentes dominios de software y su facilidad de representar conceptos modernos de ingeniería de software justifican el mayor interés por este método reciente.

**Con respecto al contexto** en el que es instanciado un método de medición, se observa que el Lenguaje de Modelado Unificado, UML, es el contexto más utilizado para representar de manera abstracta los artefactos a ser medidos. Sin embargo, la calidad de la documentación de estos artefactos con UML no es asegurada, porque se necesita de un proceso que permita construir dichos artefactos de manera consistente. Por ejemplo, la especificación de los diagramas de secuencia o casos de uso con un adecuado nivel de detalle, no está dentro del alcance de UML. En cambio, las demás propuestas que consideran un determinado método de desarrollo tienen la ventaja de que los artefactos abstractos son más consistentes, lo que conlleva a una medición más exacta.

**Con respecto al dominio funcional**, los Sistemas de Información de Gestión (SIG) se constituyen en el dominio funcional que mayor atención ha recibido. Una razón está en la misma evolución de los métodos de medición, ya que los métodos tradicionales fueron inicialmente creados para medir el tamaño funcional de estos sistemas software orientados a la gestión. Además, el interés por este dominio continúa porque dichos sistemas actualmente tienen una fuerte presencia en el mercado

gubernamental, comercial e industrial. Sin embargo, en cuanto al dominio de los Sistemas de Tiempo Real (STR), únicamente dos trabajos fueron encontrados: Diab et al. y Nagano; siendo la propuesta de Diab et al, el procedimiento de medición más destacado. Se presume que con la estandarización de COSMIC-FFP, más propuestas irán surgiendo para este dominio funcional.

**Tabla 2. Procedimientos de medición de tamaño funcional**

<i>Procedimientos de Medición</i>	<i>Método de medición</i>	<i>Contexto</i>	<i>Dominio funcional</i>	<i>Fase de desarrollo de software</i>	<i>Automatizado?</i>	<i>Verificación</i>
<i>Fetcke et al. (1998)</i>	<i>IFPUG FPA v 4.0</i>	<i>OOSE</i>	<i>SIG</i>	<i>Especificación de requisitos y Análisis</i>	<i>No</i>	<i>No</i>
<i>Tavares et al. (1999)</i>	<i>IFPUG FPA v 4.1</i>	<i>UML</i>	<i>SIG</i>	<i>Especificación y Análisis</i>	<i>No</i>	<i>No</i>
<i>Bertolami et al. (2003)</i>	<i>MARK II FPA</i>	<i>LEL</i>	<i>SIG</i>	<i>Elicitación</i>	<i>No</i>	<i>No</i>
<i>Abrahão (2004)</i>	<i>IFPUG FPA v 4.1</i>	<i>OO-Method</i>	<i>SIG</i>	<i>Análisis</i>	<i>Si</i>	<i>Si</i>
<i>Uemura et al. (1999)</i>	<i>IFPUG FPA v 4.0</i>	<i>UML v 1.1</i>	<i>SIG</i>	<i>Análisis</i>	<i>Si</i>	<i>Parcial</i>
<i>Lehne (1997)</i>	<i>IFPUG FPA v 4.0</i>	<i>OORam</i>	<i>SIG</i>	<i>Análisis y Diseño</i>	<i>No</i>	<i>No</i>
<i>Bevo et al. (1999)</i>	<i>COSMIC-FFP v 2.0</i>	<i>UML</i>	<i>SIG</i>	<i>Especificación y Análisis</i>	<i>Si</i>	<i>Parcial</i>
<i>Jenner (2001)</i>	<i>COSMIC-FFP v 2.0</i>	<i>UML</i>	<i>SIG</i>	<i>Especificación de Requisitos</i>	<i>Si</i>	<i>No</i>
<i>Azzouz (2003)</i>	<i>COSMIC-FFP v 2.2</i>	<i>RUP</i>	<i>SIG</i>	<i>Especificación, Análisis y Diseño</i>	<i>Si</i>	<i>No</i>
<i>Habela et al. (2005)</i>	<i>COSMIC-FFP v 2.2</i>	<i>UML v 1.5</i>	<i>SIG</i>	<i>Diseño</i>	<i>No</i>	<i>No</i>
<i>Poels (2003)</i>	<i>COSMIC-FFP v 2.2</i>	<i>MERODE</i>	<i>SIG</i>	<i>Análisis</i>	<i>No</i>	<i>Parcial</i>
<i>Diab et al. (2004)</i>	<i>COSMIC-FFP v 2.2</i>	<i>RRRT</i>	<i>STR</i>	<i>Análisis</i>	<i>Si</i>	<i>Si</i>
<i>Nagano et al. (2003)</i>	<i>COSMIC-FFP v 2.0</i>	<i>Shalaer-Mellor</i>	<i>STR</i>	<i>Análisis</i>	<i>No</i>	<i>No</i>

**Con respecto a la fase de desarrollo** del software, los procedimientos de medición revisados en su mayoría usan artefactos de software elaborados en la fase de análisis. Incluso Lehne y Habela especifican la funcionalidad del sistema con mayor detalle, tomando en cuenta aspectos de diseño. La ventaja de estas propuestas, en fases de análisis y diseño, es que el tamaño funcional obtenido tiende a ser con mayor exactitud debido al detalle de información proveído. Por ejemplo, tal como se afirma en la propuesta de Habela, utilizar una escala de medida a nivel de atributos de datos sería más exacta que a un nivel de grupos de datos. Sin embargo, la desventaja es que el tamaño funcional resulta ser menos oportuno para poder estimar otros indicadores de suma importancia, tales como el costo del software, la productividad, etc. Entre las propuestas que afrontan esta desventaja, se tiene a Bertolami et al., Bevo, Jenner y Azzouz, quienes miden el tamaño funcional en etapas tempranas del proceso de desarrollo, tales como la elicitación y la especificación de requisitos. Sin embargo, Bevo y Jenner no identifican con claridad ciertos componentes básicos que contribuyen al tamaño funcional (por ejemplo: movimientos escritura y lectura) en las primitivas de UML.

**Con respecto a la automatización** del procedimiento, el número de propuestas basadas en COSMIC-FFP es mayor en comparación a las propuestas basadas en los métodos tradicionales. Generalmente la automatización de estos procedimientos de medición es integrada en otras herramientas de soporte de desarrollo de software. Esta automatización ayuda a reducir costos y evitar errores en la medición.

**Con respecto a la verificación** del procedimiento, se observa que la mayoría de las propuestas no han sido verificadas; a excepción de las propuestas de Abrahão y Diab et al. que mejor satisfacen a este criterio. Bevo y Uemura et al. reportan resultados iniciales acerca de la exactitud de sus

respectivas propuestas. Poels lleva a cabo esta verificación únicamente desde la perspectiva de la teoría de la medición. Así mismo, ciertos aspectos de la validación del diseño aún no han sido plenamente contemplados en las propuestas estudiadas (evaluación de la conformidad, análisis de las unidades de medida, etc.).

Concluyendo, tal como se observa en la Tabla 2, los procedimientos de medición con mayores ventajas son las propuestas de Diab y Abrahão. Sin embargo, estos procedimientos de medición han sido definidos para medir artefactos en la fase de análisis y diseño. Ello pone de manifiesto la necesidad de definir y consolidar un procedimiento de medición que mejore las propuestas que intentan medir el tamaño funcional a partir de una etapa más temprana. Además, aún resulta ser un reto poder cuantificar con mejor exactitud y precisión el tamaño del software en la fase de especificación de requisitos.





<b>Capítulo</b>	<i>Modelo de requisitos de OO-Method</i>
<b>3</b>	

*“Delivery is not necessarily the best time  
to discover the user requirements”.*

*- Alexander’s 17th Law of Requirements*

El presente capítulo describe el conjunto de técnicas que comprende la fase de ingeniería de requisitos del método de producción automática de software OO-Method, utilizado en esta Tesis como método de referencia.

### **3.1. Introducción**

Como se ha comentado con anterioridad, en los últimos años el interés por el desarrollo de software dirigido por modelos ha venido siendo cada vez mayor. La propuesta Model Driven Architecture (MDA) fue promovida por el grupo OMG (Object Management Group) [108] con el propósito de separar la lógica de las aplicaciones de la plataforma en la que será implementada. Esta separación permite reducir el impacto que tiene la evolución de las tecnologías sobre el desarrollo de las aplicaciones. Asimismo, se tiene la ventaja de que una misma especificación puede ser implementada en diferentes plataformas software. De este modo, con MDA los modelos se convierten en el insumo más importante del proceso de desarrollo de software.

OO-Method es un método basado en la transformación de modelos que sigue las directrices de MDA [54], [109]. Este método establece una clara distinción entre el *espacio del problema*, que está centrado en definir qué es lo que hace el sistema, y el *espacio de la solución*, centrado sobre cómo es implementado dicho sistema. El proceso de producción de software asociado a OO-Method comprende básicamente tres fases, a las que van asociadas respectivamente tres modelos genéricos: modelo de requisitos, modelo conceptual y modelo de ejecución.

Tal como se observa en la Figura 2, tanto el *modelo de requisitos* como el *modelo conceptual* permiten modelar de una manera abstracta la aplicación a desarrollar, sin detalles de implementación (en términos MDA, hablamos de un “PIM”, Platform Independent Model).

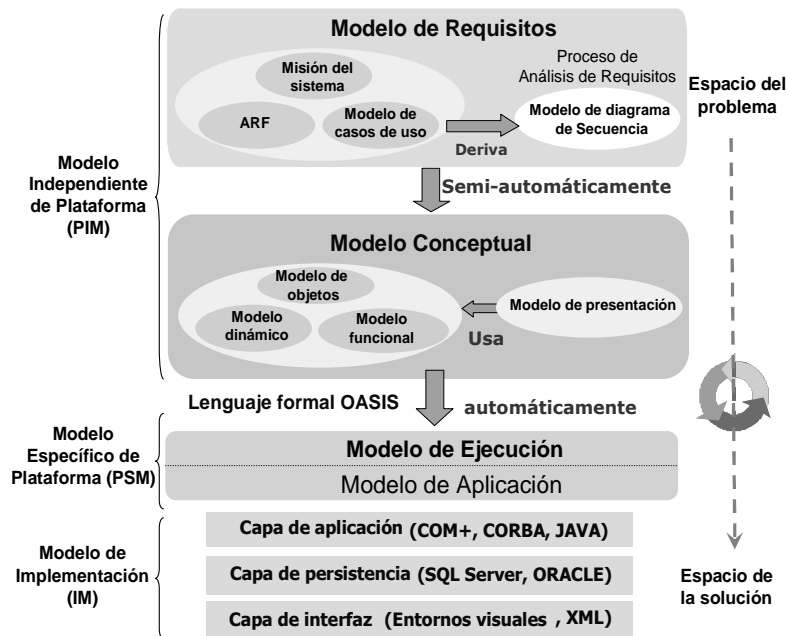


Figura 2. Proceso de desarrollo de OO-Method

Luego, ambos modelos independientes de plataforma son transformados en otro modelo que sí incluye detalles de implementación (en términos MDA, hablamos de un “PSM”, Platform-Specific Model). Esto es logrado mediante el *modelo de ejecución* que está compuesto por un conjunto de reglas de transformación que permiten definir un modelo de aplicación. Finalmente, cada uno de los elementos del modelo de aplicación es transformado en código ejecutable para diferentes plataformas de software (en términos MDA, se trata del “IM”, Implementation Model).

A continuación introducimos cada uno de estos tres modelos:

**Modelo de Requisitos**, el cual está compuesto por un conjunto de técnicas como la definición de la Misión del Sistema, el Árbol de Refinamiento de Funciones, el Modelo de Casos de Uso y el Modelo de Diagramas de Secuencia [77],[81]. Estos diagramas de secuencia son derivados semi-automáticamente a partir del modelo de casos de uso [106], [110] (Ver Figura 2).

**Modelo Conceptual**, compuesto por el Modelo de Objetos, Modelo Dinámico y Modelo Funcional, los cuales permiten capturar las propiedades estáticas y dinámicas de los requisitos funcionales del sistema. Además un Modelo de Presentación es también incluido, el cual permite la especificación de interfaces de usuario de una manera abstracta [111],[112]. El Modelo Conceptual es obtenido de manera semiautomática a partir de los diagramas de secuencia [77].

Un lenguaje de especificación formal, llamado OASIS [113], es utilizado para expresar con base formal cada una de las primitivas del Modelo conceptual. A partir de esta especificación, el modelo de ejecución determina cómo obtener el producto software final (espacio de la solución) en una manera automática.

**Modelo de Ejecución**, compuesto por un conjunto de reglas de transformación que permiten la transición del espacio del problema al espacio de la solución, mediante la definición de un modelo de aplicación que determina la forma en la que va a ser ejecutada la especificación resultante del esquema conceptual. A partir de este modelo se genera de manera sistemática y automática una aplicación software en la plataforma tecnológica seleccionada [114].

En la siguiente sección, se describe el conjunto de técnicas utilizadas en la fase de ingeniería de requisitos, ya que –como se ha visto en la introducción- es el correspondiente Modelo de Requisitos el que es relevante en el ámbito de ejecución de esta Tesis.

### ***3.2. Ingeniería de requisitos de OO-Method***

Esta fase de ingeniería de requisitos de OO-Method tiene como fin capturar las propiedades funcionales que el sistema requiere y a partir de un análisis de requisitos obtener semi-automáticamente su funcionalidad interna expresada mediante un conjunto de diagrama de secuencias. En [77], [81] se presentan un conjunto de técnicas que son descritas de manera resumida a continuación.

#### ***3.2.1. Misión del sistema***

Su objetivo es describir el propósito del sistema, sus responsabilidades y alcance. A través de la definición de su misión es posible determinar con precisión qué hará y qué no hará el sistema. Es de vital importancia concensuar con los usuarios, desde un inicio, el objetivo del sistema y tenerlo presente durante todas las fases del proceso de desarrollo del sistema. La misión del sistema comprende de tres elementos:

- **Definición:** corta descripción del tipo de sistema a ser diseñado.

- **Propósito:** indicación general del propósito del sistema.
- **Responsabilidades:** cada responsabilidad es una funcionalidad general del sistema.

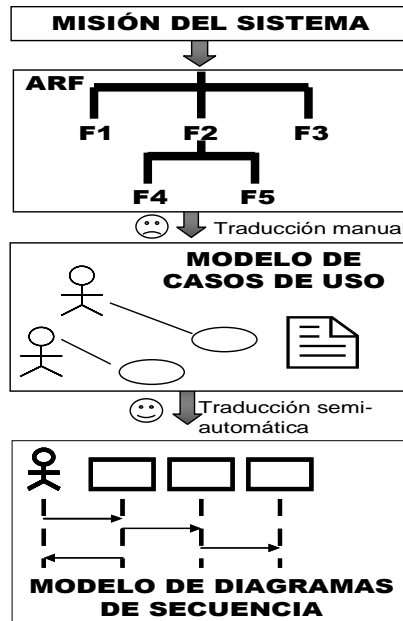


Figura 3. Ingeniería de Requisitos de OO-Method

### 3.2.2. *Árbol de refinamiento de funciones*

El Árbol de Refinamiento de Funciones representa la descomposición jerárquica de las funciones de un sistema. Tal como se observa en la figura 3, la raíz de este árbol representa la misión del sistema, definida anteriormente, y cada nodo hoja representa a una función elemental. En [115] se define una función como elemental si esta función es disparada por un evento enviado por un usuario del sistema o por la ocurrencia de un evento temporal. Los nodos intermedios del árbol son llamados grupos funcionales y usualmente representan un tipo de actividad o un área del negocio donde el sistema está siendo desarrollado.

El árbol resultante es una organización de interacciones externas que no dice nada acerca de la composición interna del sistema. Sin embargo, este árbol de refinamiento de funciones es un insumo muy útil para la identificación de casos de uso, pues permite determinar con claridad toda la funcionalidad del sistema a un mismo nivel de abstracción, facilitando su completitud.

### 3.2.3. *Modelo de casos de uso*

En OO-Method, los casos de uso son comúnmente utilizados como sinónimo de escenarios [116]. Un escenario describe una situación que ocurre en el universo del discurso, de la misma forma que lo hace un caso de uso [117].

El modelo de casos de uso permite capturar los requisitos funcionales del sistema desde la perspectiva del usuario. Dos tipos de casos de uso son utilizados para describir la funcionalidad del sistema:

*Casos de uso primarios* que representan las funciones más importantes del sistema y son directamente obtenidos a partir de los nodos hoja del Árbol de Refinamiento de Funciones.

*Casos de uso secundarios* que surgen con la finalidad de descomponer la complejidad de un caso de uso. Estos casos de uso complementan o extienden información a uno o varios casos de usos primario a través de las relaciones EXTEND e INCLUDE.

En una *relación de inclusión*, la funcionalidad definida en un caso de uso es incorporada o insertada por completo en otro, denominado *caso de uso base*. La inserción se efectúa en el lugar definido del caso de uso base; de este modo se ejecuta el caso de uso incluido, cuando el punto de inserción es identificado en el caso de uso base. Finalizada la ejecución del caso de uso incluido, se continúa con la secuencia de acciones del caso de uso base. Así, aunque el caso

de uso incluido pueda contener varios caminos debido, por ejemplo, a condicionales, todos estos caminos deben terminar de manera tal que el caso de uso base pueda continuar su trayectoria original de eventos.

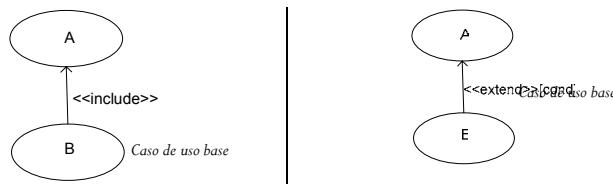
La realización de la funcionalidad especificada en el caso de uso base depende de los resultados obtenidos de la ejecución del caso de uso incluido. Sin embargo, estructuralmente, el caso de uso base es independiente de este último. Esto significa que no depende, por ejemplo, de los atributos y operaciones del caso de uso incluido. Nótese que el caso de uso incluido representa una funcionalidad encapsulada que puede ser reutilizada en muchos casos de uso. Un caso de uso puede ser incluido en más de un caso de uso y también es posible incluir en un caso de uso la funcionalidad de varios casos de uso.

En una relación de extensión, una instancia de un caso de uso (el caso de uso base) puede ser incrementada con la estructura y comportamiento definido en otro caso de uso (el caso de uso que extiende). Un caso de uso puede extender muchos casos de uso y, por otra parte, un caso de uso puede ser extendido por más de un caso de uso.

La funcionalidad del caso de uso base es independiente de la que describe el caso de uso que extiende. Esto significa que cada uno de los casos de uso relacionados por extensión tiene vida propia y podrían ser ejecutados por separado, sin necesidad de que la relación se concrete. El caso de uso base no depende entonces ni estructural ni funcionalmente del caso de uso que lo extiende y viceversa.

La relación de extensión contiene una condición que debe cumplirse para que la extensión del caso de uso base pueda efectuarse. Y uno o más puntos de extensión, es decir, posiciones o ubicaciones en el cuerpo del caso de uso base donde se incorporará la funcionalidad del caso de uso extensión

La Figura 4 muestra algunos tipos de relaciones entre casos de uso.



**Figura 4:** Relaciones entre casos de uso [81]

El diagrama de casos de uso representa una vista general de los casos de uso, sus relaciones de comunicación con las entidades externas (actores) y las relaciones existentes entre dichos casos de uso. Sin embargo, con el fin de especificar con mayor detalle la composición interna de un escenario, la descripción de un caso de uso es llevado a cabo mediante el uso de una plantilla (vea Figura 5).

Esta plantilla comprende de tres secciones: La primera sección es un resumen acerca del caso de uso. La segunda sección, describe el conjunto de pasos que ocurre durante el caso de uso. Son escritos de una manera conversacional entre el actor y el sistema. La tercera sección, comprende el conjunto de pasos que complementan la especificación de un caso de uso. Una condición debe decidir cual alternativa ejecutar y el flujo de control es transferido para esta alternativa.

Nombre			
Actores			
Cross-Reference			
Propósito			
Pre-condición			
Incluye a			
Extiende a	CU Nombre	Punto de extensión	Condición
Sección Curso Básico			
General	Comunicación Actor/Sistema		Respuesta al Sistema



Sección Alternativa		
General	Comunicación Actor/Sistema	Respuesta al Sistema

**Figura 5.** Plantilla de descripción de caso de uso [81].

### 3.2.4. Diagramas de secuencia

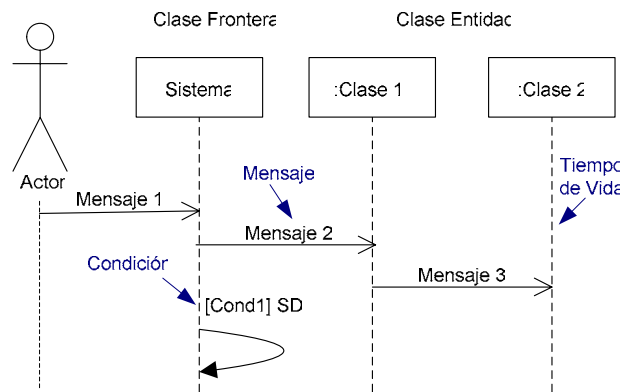
Los diagramas de secuencia son el resultado de un proceso de análisis de requisitos, el cual es soportado por un marco llamado Metamorfosis [110]. Este marco proporciona la base necesaria para el establecimiento de vínculos persistentes y controlables entre la especificación de los requisitos funcionales del sistema y el producto de su análisis. Metamorfosis permite la deducción semi-automática del modelo de diagrama de secuencias a partir de la descripción de los casos de uso [110], [118].

Los diagramas de secuencia permiten describir patrones de interacción mediante un conjunto de mensajes entre objetos o clases requeridos para realizar un comportamiento del sistema. Estos diagramas muestran la secuencia ordenada en el tiempo de los mensajes que envían y reciben genéricamente los objetos durante la ejecución de un escenario. Se define un mensaje como una comunicación entre dos objetos que se transmiten información con la finalidad de que se ejecute una actividad. Un mensaje representa los roles que deben cumplir tanto el objeto cliente como el objeto servidor, así como la acción que será ejecutada.

La notación a utilizar para su representación es semi-formal, ya que es proveída por UML con algunas extensiones que se han incorporado en los mensajes para identificar elementos del Modelo Conceptual OO-Method.

Básicamente, tal como se observa en la Figura 6, la estructura general de un diagrama de secuencia tiene dos dimensiones: La dimensión vertical que

representa el tiempo y la horizontal que representa el conjunto de clases del sistema.



**Figura 6.** Estructura y notación de un diagrama de secuencia [81]

En este contexto, OO-Method distingue dos tipos de clases:

**Clases Frontera:** Son las que manejan la comunicación con actores externos. Este tipo de clase encapsula el entorno dependiente del comportamiento y protege la integridad de los objetos entidad.

**Clases Entidad:** Representan el dominio de la vida real de objetos o conceptos que son internos al sistema, como por Ejemplo las “típicas” clases cliente, alquiler, empleado. Los actores externos usualmente no tienen contacto directo con los objetos entidad. En lugar de esto, ellos son accedidos a través de los objetos frontera. Una de las propiedades de las clases entidad son las restricciones de integridad que definen condiciones al conjunto de posibles estados en los que puede encontrarse un objeto a lo largo de su existencia.

Sin embargo, lo más característico de estos diagramas son los diferentes tipos de mensajes que envían o reciben cada una de las clases a través del tiempo. Cada uno de estos mensajes puede ser etiquetado por un

número secuencial para mostrar el orden de interacción respectivo. Generalmente son omitidos porque la ubicación física del mensaje muestra la secuencia relativa. Además un mensaje puede también ser etiquetado por:

- Alguna condición que permite la interacción si es satisfecha. La sintaxis para esta condición es: [Boolean-expresión] message-name.
- Alguna expresión que expresa iteración. Cuya sintaxis sería representada por:

[Lower-bound..upper-bound] DO message-name

FOR [expression] DO message-name

WHILE [expression] DO message-name

[Query-expression] DO message-name

[\*] DO message-name

Quando un conjunto de mensajes se ejecutan de manera iterativa o condicionada, estos mensajes son incluidos gráficamente en una caja rectangular como se muestra en Figura 6. La sintaxis es como sigue:

[Boolean-expression]

[Lower-bound...upper-bound] DO

FOR [expression] DO

WHILE [expression] DO

[query-expression] DO

Entre los *tipos mensajes* que se distinguen en los diagramas de secuencia [77], tenemos:

- *Mensajes de Señal*: Son aquellos que representan una interacción entre el actor y el sistema, su estereotipo es <<Signal>>. El sistema se

comunica con el exterior (representado por el actor) utilizando estos mensajes. La información recogida en estos mensajes permite realizar una interacción entre objetos de la clase sistema. Estos mensajes son muy importantes para la construcción de la interfaz de usuario. Este tipo de mensajes tendrá una de las siguientes propiedades:

*Input:* Aplicable a mensajes de tipo señal cuyo origen es una clase que representa a un actor del Diagrama de Casos de Uso y cuyo destino es la clase que representa al sistema.

*Output:* Aplicable a mensajes en los que el origen es el sistema y el destino es una clase actor del Diagrama de Casos de Uso.

- Mensajes de Servicio: Los mensajes estereotipados con la etiqueta <<service>> representan la modificación de estado de la clase receptora del mensaje cuando ocurre una interacción. El tipo de cambio puede ser de tres tipos:

*New:* Aplicable a mensajes cuyo destino es una clase del sistema en la cual se desea crear un nuevo elemento que cumpla todas las características de dicha clase.

*Destroy:* Aplicable a mensajes en los cuales se desea borrar un elemento de la clase destino.

*Update:* Aplicable a mensajes cuyo destino es una de las clases del sistema que modifica su estado.

- Mensajes de Consulta: Los mensajes de consulta son estereotipados con la etiqueta <<query>> y representan consultas sobre el estado de un objeto. Para que un objeto pueda conocer parte del estado de otro objeto establece una interacción entre ellos. El conjunto de

propiedades que el analista puede asignar a estos mensajes se presenta en la Tabla 3:

**Tabla 3.** Propiedades del mensaje tipo consulta [81].

Propiedades	Valor	Descripción
<b>Tipo</b>	Población	La consulta es ejecutada sobre la población de la clase receptora y el resultado es una lista de objetos.
	State	La consulta es ejecutada sobre un objeto u objetos relacionados. El resultado es un valor calculado.
<b>Resultado</b>	Nombre	Esto es el nombre de la variable que almacena el resultado. La sintaxis es: "nombre_resultado = nombre_mensaje"
<b>Descripción</b>	--	Descripción de la consulta.
<b>Multiplicidad</b>	0..1 0..M 1..1 1..M N..M	Número de objetos participantes en la consulta

- Mensajes de Conexión: Los mensajes con el estereotipo <<connect>> se utilizan para establecer una relación estructural entre los objetos participantes en la interacción.

**Tabla 4.** Propiedades del mensaje tipo conexión [81].

Propiedades	Valor	Descripción
<b>Mínima Multiplicidad</b>	0 1 ... M	Número mínimo de objetos en la clase receptora que puede ser conectado o desconectado del emisor de la interacción.
<b>Máxima Multiplicidad</b>	1 ... M	Número máximo de objetos en la clase receptora que puede ser conectado o desconectado del emisor de la interacción.
<b>Actividad</b>	Insert	La interacción es para conectar el receptor al emisor.



### 3.3. Herramienta de soporte

OlivaNova Model Execution (ONME) es un conjunto de productos que han sido implementados por CARE technologies [119] con el fin de dar soporte a OO-Method, un método que sigue los lineamientos MDA (Model Driven Architecture) [108].

Entre los principales productos de ONME se tiene [120]:

- *OlivaNova Modeler*, herramienta de edición y validación de modelos conceptuales OO-Method (PIMs).
- *OlivaNova Transformation Engines*, es una implementación de una herramienta que opera las transformaciones de PIM a PSM y las transformaciones de PSM a IM.
- *OlivaNova Model Examiner*, permite encontrar diferencias entre dos esquemas conceptuales.
- *OlivaNova Tester* permite probar la comunicación entre los programas del cliente y del servidor.
- *OlivaNova Migration Modeler* permite migrar datos debido a que el esquema conceptual ha sido modificado (evolución).
- *Olivanova Function Points Counter* permite medir el tamaño funcional de los proyectos basados sobre esquemas conceptuales.

La Figura 8 ilustra el proceso general de desarrollo con ONME.

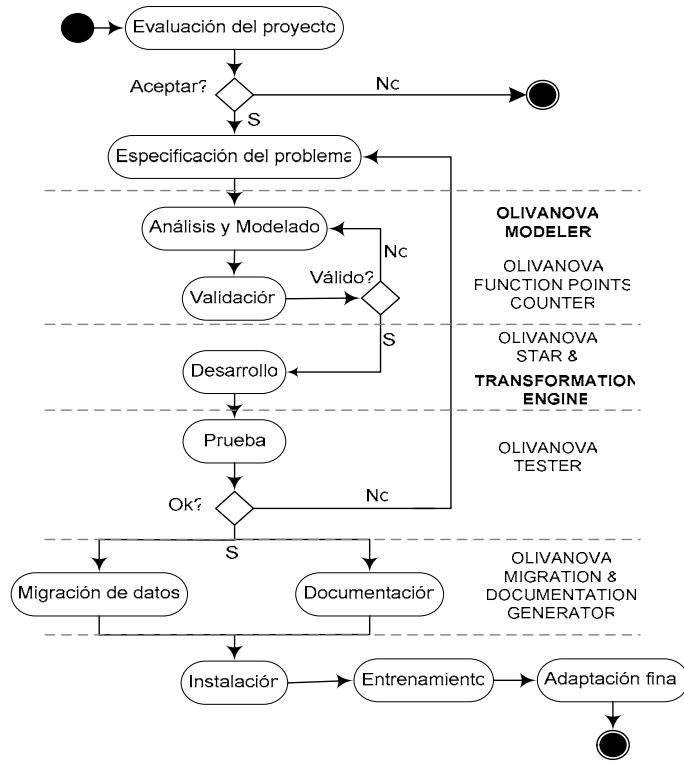


Figura 8. Proceso de desarrollo con ONME

Actualmente, ONME no contempla algún producto que de soporte a la especificación del problema en el ámbito del modelado de requisitos. Para cubrir esa carencia -dada la importancia de esta actividad- se desarrolló una herramienta, llamada RETO [121], en la Universidad Politécnica de Valencia y en estrecha colaboración con la empresa CARE-Technologies. El propósito de esta herramienta fue dar soporte al Modelo de Requisitos de OO-Method, capturando de esta forma la funcionalidad del sistema. Además poder generar automáticamente, mediante un Modelo de Trazabilidad, primeras versiones de esquemas conceptuales. El diseño de esta herramienta es explicado detalladamente en [81].



### **3.4. Conclusiones**

Después de introducir OO-Method como método basado en la transformación de modelos que es conforme con los principios de MDA, este capítulo se centra en presentar la estrategia de modelado de requisitos proporcionado por OO-Method. Ello es especialmente significativo considerando que este trabajo de Tesis tiene como objetivo primario el hacer posible la medición del tamaño funcional de un producto software a partir de su Modelo de Requisitos en el ámbito de OO-Method como marco de referencia.

El conjunto de técnicas a utilizar durante la fase de ingeniería de requisitos es por lo tanto descrito. Esta fase se inicia con la definición de la misión o propósito del sistema, se complementa con la especificación de un Árbol de Refinamiento de Funciones de acuerdo con ese propósito del sistema, y con el conjunto de diagramas de casos de uso que describen las acciones elementales resultantes de dicho árbol.

Finalmente, esta fase culmina con la obtención semi-automática de los diagramas de secuencia que son obtenidos a partir de cada caso de uso. Estos artefactos de software obtenidos en una etapa temprana del proceso de desarrollo de OO-Method se constituyen en el objeto a ser medido con un procedimiento de medición que será presentado en los siguientes capítulos.

<b>Capítulo</b>	<b><i>RmFFP: Un procedimiento de medición</i></b>
<b>4</b>	

*“Medir lo que es medible y hacer medible lo que no es”.*

*- Galileo Galilei*

El presente capítulo presenta un procedimiento de medición que ha sido diseñado de manera sistemática y rigurosa en base al modelo general del proceso de medición propuesto por Abran y Jacquet [34]. El propósito de este procedimiento es medir el tamaño funcional de las aplicaciones generadas con OO-Method a partir de una especificación de requisitos de alto nivel.

#### ***4.1. Introducción***

Tal como se mencionó en el capítulo 2, el método de medición de tamaño funcional con mayor repercusión ha sido el Análisis de Puntos de Función, ya que a partir de este método se han derivado una variedad de métodos con el fin de proveer una mayor aplicabilidad a diferentes dominios de software. Uno de estos métodos es COSMIC-FFP (Full Function Points), el cual ha sido aprobado como un método estándar mediante la ISO/IEC 19761 [22]. Este método a diferencia de los demás métodos estándares (IFPUG FPA, Mk II FPA y NESMA FPA) provee una mejor compatibilidad con conceptos modernos de la ingeniería de software y permite medir la funcionalidad del --

sistema a diferentes niveles de abstracción. Sin embargo, con el fin de automatizar completamente el proceso de medición, el carácter genérico de COSMIC-FFP necesita ser instanciado en un método de desarrollo.

A continuación se presenta el diseño de un procedimiento de medición, llamado RmFFP, el cual permite medir el tamaño funcional en una fase temprana del proceso de desarrollo de OO-Method.

## **4.2. Diseño de RmFFP**

Esta sección presenta el diseño de RmFFP, el cual ha sido guiado por una serie de pasos que forman parte del Modelo de Proceso de Medición [34]:

1. Definición de objetivos.
2. Caracterización del concepto a medir.
3. Diseño del metamodelo
4. Definición de reglas de asignación numérica.

A continuación vamos a presentarlos con más detalle.

### **4.2.1. Definición de objetivos:**

En términos de la plantilla GQM (Goal/Question/Metric) [80], el objetivo de este trabajo es: **definir un procedimiento de medición para el propósito de medir una especificación de requisitos con respecto a su tamaño funcional desde el punto de vista del analista en el contexto del Modelo de Requisitos de OO-Method.**

### **4.2.2. Caracterización del concepto a medir**

A partir de nuestro objetivo, identificamos que el concepto a medir es el *tamaño funcional*, el cual es definido por el estándar ISO/IEC 14143-1 como el *tamaño del software derivado de la cuantificación de los requisitos funcionales de usuario* [16]. Sin embargo, como este concepto es definido de manera muy

general, necesitamos aclarar qué se entiende por cuantificación de requisitos funcionales de acuerdo al método estándar COSMIC-FFP [22]. Según el principio de medición de COSMIC-FFP, esta cuantificación es entendida como el *número de movimientos de datos*. Por lo tanto, los movimientos de datos vienen a constituirse en el componente básico funcional a considerar.

#### 4.2.3. *Diseño del metamodelo*

El metamodelo de COSMIC-FFP fue diseñado con el fin de representar de manera más clara los diversos conceptos genéricos que son presentados en su manual de medición [24], así como también poder identificar las diferentes relaciones existentes entre dichos conceptos. La Figura 9 muestra el metamodelo de COSMIC-FFP, representado mediante un diagrama de clases de UML [17], que fue seleccionado por su sencillez, expresividad y popularidad.

Como se puede observar en este metamodelo, el objeto de interés a ser medido puede ser identificado desde muchos puntos de vista de medición. El *punto de vista* determina el nivel de detalle con que puede ser enfocado un objeto de interés; por ejemplo el detalle observado bajo el punto de vista de medición de un programador será mayor que la de un usuario final. Tal como se observa en el diagrama, este punto de vista es determinado por el propósito de la medición. El *propósito de la medición* define el porqué de la medición y para qué será utilizada. Este propósito ayuda también a determinar el *alcance de la medición*; es decir, determinar la funcionalidad a ser incluida en una medición particular.

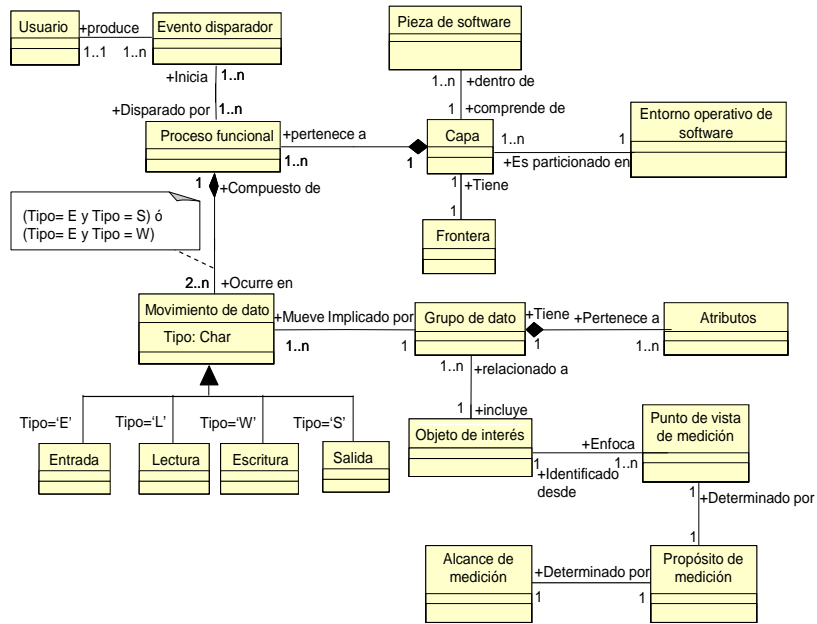


Figura 9. Metamodelo de COSMIC-FFP

El *objeto de interés*, identificado desde un punto de vista de medición, puede ser alguna entidad física u objeto conceptual descrito por un conjunto de atributos que pertenecen a un *grupo de datos*, siendo un *atributo de datos* definido como la unidad de información más pequeña. De este modo, una relación de agregación ha sido utilizada entre los conceptos grupo de datos y atributo de datos. Además, uno o más grupos de dato están directamente relacionados a un solo objeto de interés y están siempre implicados en algún movimiento de dato. Tal como se observa en la Figura 9, este *movimiento de dato* ocurre en un proceso funcional y puede ser de cuatro tipos: *entrada*, *lectura*, *escritura* y *salida*. Por ejemplo, un movimiento de dato de tipo entrada mueve un grupo de dato, a través de la frontera, desde un usuario hasta el proceso funcional que requiere dicho grupo de dato. Cada *proceso funcional* está compuesto de un mínimo de dos movimientos de datos: una entrada y una salida o una escritura. Esto es representado por medio de la cardinalidad

mínima de dos en el rol "Ocurre en" de la relación entre proceso funcional y movimiento de dato.

El conjunto de procesos funcionales ejecutados en el mismo nivel de abstracción constituye el concepto de capa. Una *capa* es el resultado de la partición funcional del entorno operativo de software. Así mismo, esta capa puede ser dividida en uno o más *piezas de software*. El *entorno operativo de software* es el conjunto de software que opera concurrentemente sobre un sistema específico. En un entorno de software multicapa, cada capa es un usuario de otra capa, ya que una capa utiliza los servicios funcionales proporcionados por otras capas subordinadas. Entre dos capas existe una *frontera*, la cual es definida como una interfaz conceptual entre el software bajo estudio y sus usuarios. Un *usuario* es definido como la persona o cosa que comunica o interactúa con el software.

Retomando el concepto de proceso funcional, se observa en la Figura 9 que este proceso es disparado por uno más eventos disparadores. Consecuentemente, un *evento disparador* es un evento que inicia uno o más procesos funcionales, que son provocados directa o indirectamente por un usuario.

#### **4.2.3.1. Reglas de representación**

Antes de presentar este conjunto de reglas de representación, es necesario definir el contexto de la medición, el cual es comprendido por tres conceptos del metamodelo: propósito de la medición, alcance de la medición y punto de vista de medición. El *propósito de la medición* usando RmFFP es medir el tamaño funcional de los sistemas software generados automáticamente a partir de una especificación de requisitos de OO-Method. En base a este propósito, el *punto de vista* utilizado para la medición viene a ser el del "analista", que permitirá revelar toda la funcionalidad del software a ser

desarrollada. Sin embargo, esta funcionalidad revelada necesita ser delimitada por el *alcance de la medición* que comprende la funcionalidad a ser medida.

A continuación, se describe el conjunto de reglas de representación que pudieron ser definidas para representar los conceptos del metamodelo de COSMIC-FFP en las respectivas primitivas del modelo de requisitos de OO-Method, las cuales fueron presentadas en el anterior capítulo (Vea Figura 7).

### **Capa de software**

En el modelo de requisitos de OO-Method, tanto la definición de la misión del sistema como la construcción del árbol de refinamiento de funciones permiten capturar los requisitos funcionales a un mismo nivel de abstracción funcional. Esto implica que una sola *capa de software* es considerada, ya que no existen diferentes niveles de abstracción. Además, no existe una correspondencia entre el concepto de capa y alguna primitiva particular de OO-Method en el contexto considerado.

### **Usuario**

En OO-Method un actor puede representar a un ente humano (persona que interactúa con la aplicación) o a un ente no humano (alguna aplicación externa). La representación del *concepto usuario* es trivial debido a su correspondencia directa con el actor de un diagrama de casos de uso.

La siguiente regla fue definida:

**Regla 1:** *Aceptar cada actor de un diagrama de casos de uso como un usuario del sistema.*

### **Frontera de software**

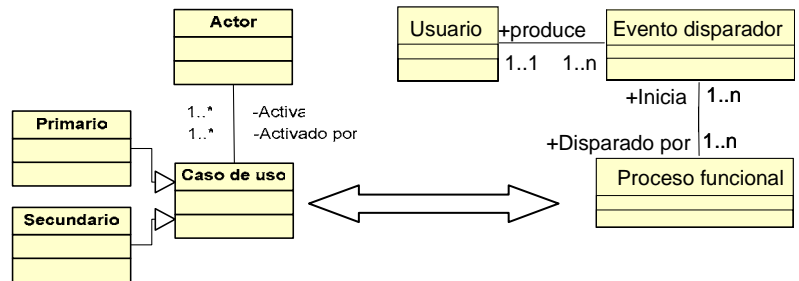
La representación de este concepto es más fácil una vez identificado los usuarios, ya que la frontera se encuentra justamente entre estos usuarios y

la funcionalidad del sistema comprendida por el alcance de la medición. Por consiguiente, un diagrama de casos de uso que muestra la interacción entre los actores y los casos de uso, se corresponde con el concepto de frontera de software. La regla definida es como sigue:

**Regla 2:** La frontera de software es el límite existente entre los actores y los casos de uso del diagrama de casos de uso.

**Proceso funcional**

Un proceso funcional es representado por un caso de uso, ya que todos casos del uso pueden ser activados por algún evento que es enviado directa o indirectamente por un actor. La Figura 10 muestra esta correspondencia entre ambos conceptos.



**Figura 10.** Correspondencia entre procesos funcionales y casos de uso

Dicha correspondencia da lugar a las dos reglas expuestas a continuación:

**Regla 3:** Aceptar cada caso de uso primario como un proceso funcional.

**Regla 4:** Aceptar cada caso de uso secundario como un proceso funcional.

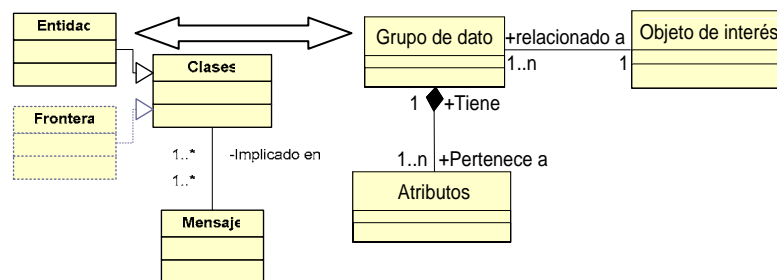
**Grupo de datos**

Tal como se muestra en la Figura 11, las clases de tipo entidad, especificadas en un diagrama de secuencia, son identificadas como grupos de



datos, ya que estas clases representan a una colección de objetos persistentes, que son descritos por un conjunto de atributos. La clase de tipo frontera no es considerada como un grupo de dato, debido a que este tipo de clase carece de persistencia. Por lo tanto, la regla es como sigue:

**Regla 5:** *Aceptar cada clase de tipo entidad que aparece en el diagrama de secuencias como un grupo de datos.*



**Figura 11.** Correspondencia entre grupo de datos y clases

Como un caso particular de OO-Method, los actores del diagrama de casos de uso son también considerados como grupos de datos, ya que todos estos actores se constituyen en potenciales clases de tipo entidad debido a su necesidad de ser persistentes y de ser descritos por un conjunto de atributos. La siguiente regla fue definida:

**Regla 6:** *Aceptar cada actor que participa en un diagrama de secuencias como un grupo de datos.*

**Atributo de datos**

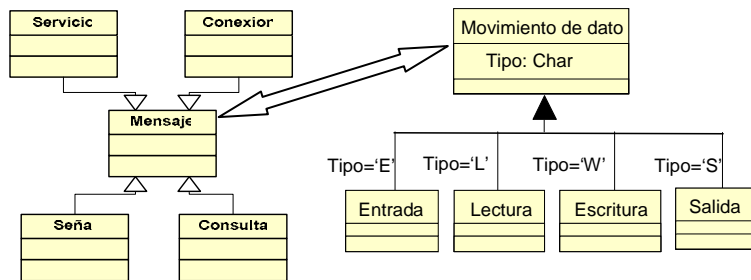
Los atributos de una clase podrían ser identificados directamente como atributos de datos. Sin embargo, en OO-Method, hay tres tipos de atributos: constante, variable, y derivado. Como un atributo de dato representa a la más mínima pieza de información, un atributo derivado no es considerado como un atributo de datos porque su valor depende de otros

atributos. Los atributos son especificados en el diagrama de secuencias como argumentos de los mensajes que permiten la creación de una instancia de clase. La siguiente regla fue definida:

**Regla 7:** *Aceptar cada argumento constante o variable de los mensajes con el estereotipo <<service/new>> como un atributo de dato.*

**Movimiento de datos**

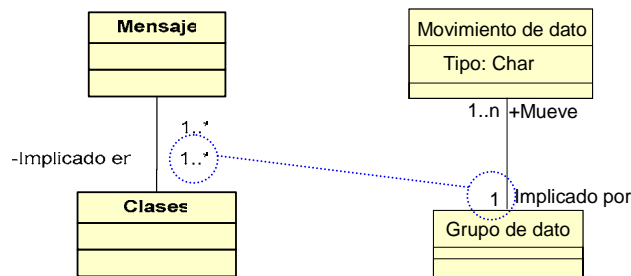
Tal como se muestra en la Figura 12, el concepto de movimiento de datos es especializado en cuatro tipos: entrada, lectura, escritura, y salida. Estos tipos de movimientos pueden ser identificados en los respectivos tipos de mensaje de interacción de OO-Method (señal, consulta, servicio y conexión), aunque también pueden ser representados en otras primitivas del modelo de requisitos.



**Figura 12.** Correspondencia entre tipos de movimientos de datos y tipos de mensajes de interacción

En la Figura 13 se observa, por un lado, la relación entre movimiento de dato y grupo de dato, la cual nos indica que en un grupo de dato puede estar implicado uno o más movimientos de datos, pero un movimiento de dato sólo mueve un grupo de dato. Por otro lado, con respecto a la relación entre mensaje y clases, se observa que una o más clases pueden ser implicadas por varios mensajes, y que estos mensajes pueden involucrar una o más clases. Por

consiguiente, un mensaje de interacción no es equivalente a un movimiento de dato, este dependerá básicamente de los grupos de datos implicados.



**Figura 13.** Relación entre movimiento de dato y grupo de dato

A continuación se presenta el conjunto de reglas que permiten identificar cada uno de estos tipos de movimiento de dato.

### Movimiento de dato-tipo entrada

La correspondencia entre un movimiento de dato de tipo entrada y un mensaje de interacción de tipo “señal” con el valor “input” es evidente, siempre y cuando este tipo de mensaje involucre por lo menos un atributo de algún grupo de dato especificado en el respectivo diagrama de secuencia.

La regla de representación es como sigue:

**Regla 8:** *Identificar un movimiento de dato de tipo entrada por grupo de dato implicado en un mensaje con el estereotipo <<signal>> y valor input.*

Dado que un mensaje de tipo “señal”, a diferencia de los otros tipos de mensajes, puede implicar atributos de diferentes clases, se adicionó dos consideraciones con el propósito de apoyar a la identificación de los movimientos de datos tipo entrada en un mensaje “señal”. Estas *consideraciones adicionales* son explicadas a continuación:

- *Consideración 1:* Como la recepción de un mensaje señal (<<signal>>) induce a la clase frontera (System) a enviar “n

mensajes” a “n líneas de vida” con al menos un parámetro (Vea Figura 14), el número de grupos de datos involucrados en un mensaje <<signal>> es determinado por el número de mensajes inducidos por la clase frontera.

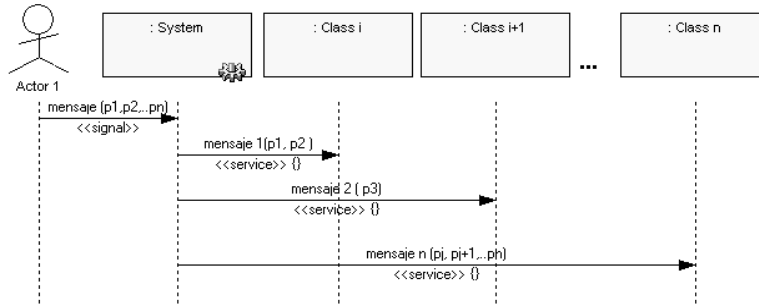


Figura 14. Fragmento de interacción tipo I

- *Consideración 2:* Como la activación de un mensaje de tipo <<connect>> se da cuando un mensaje de tipo <<service>> tiene la necesidad de establecer o eliminar vínculos entre los objetos de las clases (Vea Figura 15), el número de grupos de datos involucrados en un mensaje <<signal>> es determinado por la clase receptora del mensaje <<service>> más las clases receptoras del mensaje <<connect>>.

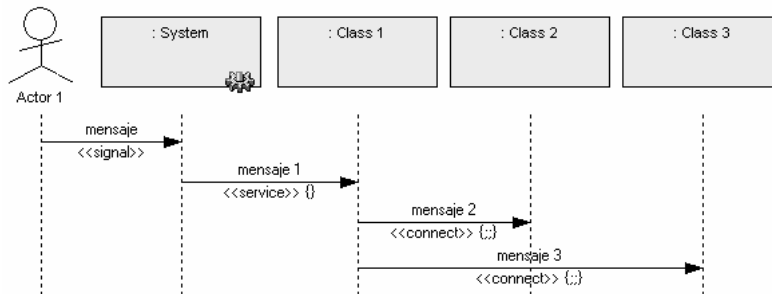


Figura 15. Fragmento de interacción tipo II

En la Figura 16 se observa tres mensajes con el estereotipo <<signal>> y valor input: el primer mensaje no es considerado como movimiento de dato porque no involucra atributos de ninguna clase. Sin embargo, los siguientes mensajes si son considerados como movimientos de datos tipo entrada. Obsérvese que el mensaje “introduce resto de datos”, teniendo en cuenta la segunda consideración, involucra atributos de las clases venta y coche, por consiguiente en este mensaje se identifica dos movimientos de dato de tipo entrada.

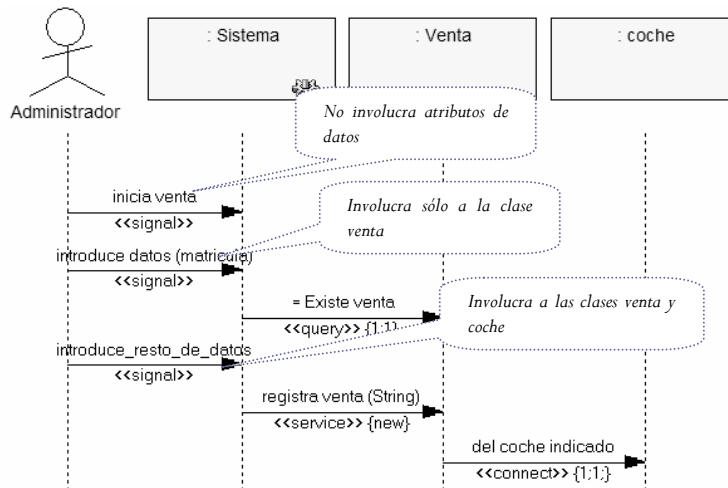


Figura 16. Identificación de movimientos de dato tipo entrada

### Movimiento de dato-tipo lectura

Para la identificación de este tipo de movimiento, se han considerado todas aquellas primitivas del modelo de requisitos que permiten expresar la recuperación de valores de atributos de una misma clase. A continuación se presenta las diferentes primitivas que se corresponden con un movimiento de tipo lectura.

- Todos los *mensajes de consulta* representan movimientos de datos de tipo lectura ya que estos tipos de mensajes implican leer el estado de los objetos de la clase receptora. Por lo tanto, la siguiente regla fue definida:

**Regla 9:** *Identificar un movimiento de datos de tipo lectura en todo mensaje con el estereotipo <<query>>.*

- Como la *condición de un mensaje* implica recuperar valores de atributos de la clase receptora que han de ser evaluados antes de la ejecución del mensaje, la siguiente regla fue definida:

**Regla 10:** *Identificar un movimiento de datos de tipo lectura en cada condición de mensaje, para cada grupo de datos involucrado en la especificación de la condición.*

Las siguientes tres reglas no están asociadas con primitivas de los diagramas de secuencia, pero éstas han sido definidas por su correspondencia con este tipo de movimiento.

- En la especificación de un caso de uso, una precondición puede ser definida para indicar una condición que debe ser satisfecha antes de la ejecución de los casos de uso. Teniendo en cuenta la regla 10, una precondición es también un movimiento de dato tipo lectura. Esta correspondencia es definida en la siguiente regla:

**Regla 11:** *Identificar un movimiento de dato de tipo lectura en cada precondición definida en la especificación de un caso de uso, para cada grupo de datos involucrado en la especificación de la precondición.*

- Una relación Extend contiene una condición que si es satisfecha extiende el caso de uso. La regla es definida como sigue:

**Regla 12:** *Identificar un movimiento de dato de tipo lectura en cada condición de una relación Extend.*

- Una restricción de integridad es una condición especificada en una clase y debería ser evaluada después de la ejecución de algún servicio de esa clase.

**Regla 13:** *Identificar un movimiento de datos de tipo lectura en cada restricción de integridad, para cada grupo de datos involucrado en la especificación de la restricción.*

### **Movimiento de dato-tipo escritura**

Un mensaje de servicio que permite la creación, destrucción y actualización del estado de los objetos de una clase, es considerado como movimiento de dato tipo escritura, ya que todos los cambios de estado son de objetos persistentes. La regla definida es como sigue:

**Regla 14:** *Identificar un movimiento de dato de tipo escritura en cada mensaje etiquetada con los estereotipos <<service/new>>, <<service/destroy>> y <<service/update>>.*

### **Movimiento de dato-tipo salida**

La relación entre un movimiento de dato salida y un mensaje signal con el valor output es trivial. La regla definida es como sigue:

**Regla 15:** *Identificar un movimiento de dato de tipo salida en cada mensaje etiquetada con el estereotipo <<signal>> y valor output.*

#### **4.2.3.2. Duplicación de movimiento de datos**

La duplicidad de movimientos de datos es un problema que afecta tanto a la exactitud como a la precisión del tamaño funcional. Sin embargo,

este problema no es tratado en el estándar ISO/IEC 19761 [22], pero si, un término llamado “*de-duplication*” es definido en el manual de COSMIC-FFP como el proceso de eliminar copias duplicadas de un listado [24]. De este modo, con el fin de eliminar la duplicidad de movimientos de datos que puede ser originada por ciertas primitivas del modelo de requisitos, se identificó cuatro casos, lo que dio origen a la definición de un conjunto reglas que son explicadas a continuación:

### Caso 1: Mensajes que involucran un mismo grupo de datos

Este caso se da cuando una consulta de datos es requerida, pero el analista decide especificarlo en dos consultas parciales para recuperar subconjuntos de atributos del mismo grupo de datos. Para propósitos de tamaño, estas dos consultas parciales serían consideradas como un único movimiento de dato de tipo lectura (vea Figura 17).

Por lo tanto, para evitar esta redundancia se definió la siguiente regla:

**Regla 16:** *Los n-mensajes del mismo tipo relacionados con el mismo objeto de la clase receptora y un mismo origen son identificados como un único movimiento de dato.*

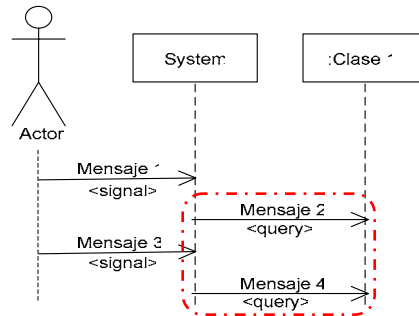


Figura 17. Ejemplo de duplicidad de movimientos de datos



### Caso 2: Mensajes de error, confirmación y advertencia

Veamos este caso con un ejemplo. Suponiendo que en la especificación de un caso de uso: “retirar dinero de un cajero” especificamos un mensaje de confirmación de usuario, un mensaje de advertencia si el usuario no seleccionó la cantidad de dinero a retirar, así como un mensaje de error si el usuario ingresó mal su clave de identificación. Estos tres mensajes son considerados como ocurrencias separadas del mismo tipo, por consiguiente se consideraría un sólo movimiento de dato de tipo salida. Generalizamos la situación expuesta en este ejemplo definiendo la siguiente regla:

**Regla 17:** *Los mensajes de error, confirmación y advertencia de un mismo proceso funcional son identificados una sola vez como un movimiento de dato tipo SALIDA.*

### Caso 3: Iteración de mensajes

Supongamos que un administrador de campo de golf necesita registrar cierta cantidad de hoyos. Para esto, se creará una instancia de la clase hoyo, con previa verificación de si el hoyo existe, esto se hará de manera repetitiva hasta que el número de hoyos sea satisfecho. Esta iteración de la ejecución de dos mensajes es representada por un bloque tal como se ilustra en la Figura 18.

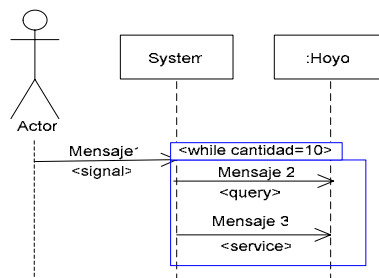


Figura 18. Ejemplo de ocurrencia de movimientos de datos

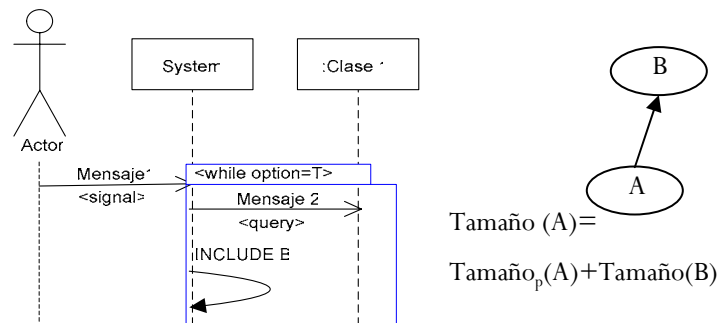
En este caso los dos movimientos de datos identificados dentro del bloque deben de ser considerados una sola vez y no diez veces, debido a que identificamos tipos de movimientos y no ocurrencias. De este modo, la siguiente regla fue definida:

**Regla 18:** El conjunto de movimientos de datos identificados en un bloque del diagrama de secuencia debe ser identificado una sola vez.

**Caso 4: Relaciones INCLUDE en un bloque**

Supóngase que se necesita realizar varios pedidos solicitados por diferentes clientes. Para hacer esto, especificamos el caso de uso “realización de un pedido” (A) que incluye otro caso de uso “información de cliente” (B). La Figura 19 ilustra de manera genérica la inclusión del caso de uso B dentro de un bloque. Para determinar el tamaño total del caso de uso A se considera una sola vez el tamaño funcional de B, a pesar de ser invocado varias veces, mientras se cumpla la condición del bloque. Por lo tanto, este caso es considerado como un caso particular del caso 3, y para tomarlo en consideración adecuadamente la siguiente regla fue definida:

**Regla 19:** Las relaciones INCLUDE especificadas dentro de un bloque del diagrama de secuencia deben ser identificadas una sola vez.



**Figura 19.** Ejemplo de ocurrencia, relación include

#### 4.2.4. Definición de reglas de asignación numérica

Hasta el momento, se tiene un conjunto de movimientos de datos identificados, pero éstos necesitan ser cuantificados y luego poder ser sumados adecuadamente para obtener un tamaño funcional por cada caso de uso, así como un tamaño funcional que represente a todo el sistema.

Para este fin, un conjunto de reglas fueron definidas. Vamos a exponerlas a continuación:

##### **A nivel de caso de uso (proceso funcional)**

La siguiente regla fue definida a nivel de caso de uso:

**Regla 20:** *El tamaño funcional de un caso del uso es igual a la suma de todos movimientos de datos identificados en el mismo caso de uso.*

Esta regla es expresada en la siguiente ecuación:

$$\text{Tamaño}(\text{Caso\_de\_uso}) = \sum_{i=1}^n f(x)_i$$

Donde:

$$f(x) = 1cfsu \quad \forall x \in P/P = \{entry, exit, read, write\}$$

Dos reglas adicionales fueron definidas debido a las relaciones que existen entre casos de uso. Así, para medir el tamaño funcional de un caso del uso extendido por uno o más casos secundarios del uso (CUS), se tiene la siguiente regla:

**Regla 21:** *El tamaño funcional de un caso de uso base extendido por otro conjunto de casos de uso secundarios es igual a la suma de los movimientos de datos identificados en cada caso de uso secundario más los movimientos de datos del caso de uso base*

En una manera semejante, para una relación incluye, el tamaño funcional de un caso de uso base (CUB), se tiene la siguiente regla:

**Regla 22:** El tamaño funcional de un caso de uso base que incluye otros casos de uso secundarios es igual a la suma de los movimientos de datos identificados en cada caso de uso secundario más los movimientos de datos del caso de uso base.

Estas dos reglas son expresadas en la siguiente manera:

$$\text{Tamaño}(CUB) = \sum_{i=1}^n \text{Tamaño}(CUS_i) + \text{Tamaño}_p(CUB)$$

Donde: La expresión 'Tamaño<sub>p</sub>(CUB)' significa el tamaño parcial del caso de uso base, obtenido al aplicar la regla 20.

### **A nivel de sistema (capa de software)**

Este paso consiste en agregar los resultados de aplicar la función de medición a todos los casos de uso primarios (CUP). Los casos de uso secundarios no son considerados en este nivel para evitar la redundancia en la medición. Por lo tanto, la regla definida es como sigue:

**Regla 23:** El tamaño funcional de una capa de software es igual a la suma de los tamaños funcionales de todos los casos de uso primarios.

Esta regla es expresada en la siguiente ecuación:

$$\text{Tamaño}(capa)_1 = \sum_{i=1}^n \text{Tamaño}(CUP_i)$$

### **4.3. Conclusiones**

En este capítulo se presentó el diseño del procedimiento de medición RmFFP, el cual permite medir el tamaño funcional desde una especificación de requisitos, obtenida con el método de desarrollo OO-Method.

Este procedimiento de medición ha sido diseñado siguiendo los pasos generales del modelo de proceso de medición propuesto por Jacquet y Abran.

Esto implicó caracterizar el tamaño funcional como concepto a ser medido por el procedimiento RmFFP. Esta caracterización fue llevada a cabo en base al método estándar COSMIC-FFP, el cual fue seleccionado principalmente por dos razones:

1. *Proveer una mejor compatibilidad con conceptos modernos de la ingeniería de software*, lo que facilitó representar las primitivas del Modelo de Requisitos de OO-Method, un método que permite que generar automáticamente aplicaciones software orientado a objetos.
2. *Proveer una mayor aplicabilidad a diferentes dominios de software*, lo que facilita a la evolución de RmFFP en conjunto con OO-Method. Actualmente OO-Method está siendo enriquecido por la definición de modelos que permiten modelar e implementar aplicaciones de sistemas pervasivos.

Por otro lado, en base al metamodelo de COSMIC-FFP, se definió un conjunto de reglas de representación con el fin de poder abstraer las primitivas del Modelo de requisitos de OO-Method que contribuyen con el tamaño funcional en base a cada uno de los conceptos del metamodelo de COSMIC-FFP. Para ello, se ha considerado el concepto de movimiento de datos como el componente básico para la obtención del tamaño funcional.

Cabe mencionar que los conceptos de evento disparador y capa no pudieron ser representados de manera explícita en alguna primitiva. Sin embargo, estas representaciones no fueron indispensables, ya que el concepto de evento disparador sirve de ayuda a la identificación de los procesos funcionales, los cuales ya son claramente identificados por los conceptos de Usuarios (Regla 1) y Frontera (Regla 2). En cuanto al concepto de capas, al no existir partición funcional del entorno operativo del software ya que los

requisitos funcionales tienen un mismo nivel de abstracción, este concepto no fue representado en alguna primitiva.

Así mismo, un conjunto de reglas adicionales fueron definidas para eliminar los movimientos de datos duplicados. Finalmente, cuatro reglas adicionales también fueron definidas, orientadas a obtener el tamaño funcional de cada caso de uso y de todo el sistema.



<b>Capítulo</b>	<b><i>Proceso de la aplicación de RmFFP</i></b>
<b>5</b>	

*“Si no puedes medirlo, tu no puedes mejorarlo”.*

*- Lord Kelvin*

El presente capítulo tiene como fin describir el proceso de aplicar el procedimiento de medición RmFFP, mediante la descripción de un conjunto de pasos a llevar a cabo para la obtención del tamaño funcional en el contexto de OO-Method. Un caso de estudio es también presentado con el fin de ilustrar este proceso.

### ***5.1. Introducción***

Según Desharnais y Abran [100], la aplicación de un método de medición implica, en la persona que mide, la ejecución de un proceso cognitivo. Dicho proceso se inicia con la *comprensión* del software a medir, la *interpretación* de su significado para la *identificación* de lo que es relevante y finalmente la *utilización* de las reglas de asignación numérica para la *obtención* de una solución. Sin embargo, en un contexto industrial, Desharnais y Abran identificaron dos tipos de dificultades que ocurren frecuentemente al aplicar un método de medición, los cuales pueden originar problemas de homogeneidad en los resultados de medición. [100]: 1.- el modelo de software puede no estar disponible, y si lo está, suele estar incompleto.



2.- La aplicación de los formalismos de un método de medición a una variedad de productos abstractos que pueden estar incorrectamente documentados.

Ambos problemas podrían ser aminorados con la definición de un procedimiento de medición de tamaño funcional, ya que la construcción del modelo de software podría ser llevada a cabo automáticamente, mediante la implementación de un conjunto de reglas de representación previamente definidas. Asimismo, la documentación del software puede ser mejorada, ya que se trata de un procedimiento de medición que es definido en un contexto particular, y el método de desarrollo seleccionado contribuirá a la obtención de una mejor documentación del software a medir.

Tal como se observa en Figura 20, la aplicación de un procedimiento de medición de tamaño funcional puede ser definida como un proceso que consiste en abstraer las primitivas relevantes de algún artefacto abstracto, de acuerdo con el metamodelo de un método de medición y cuantificar los elementos abstraídos para obtener un tamaño funcional.

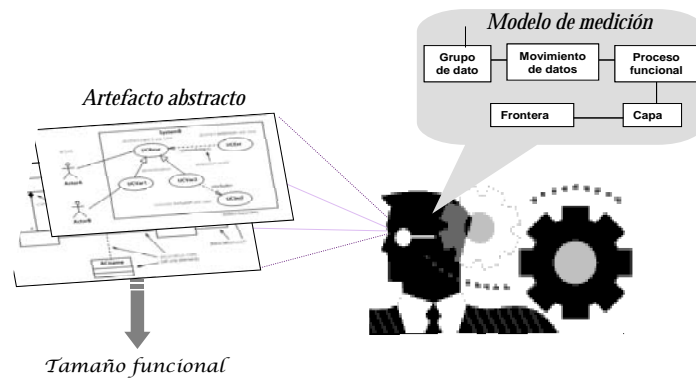
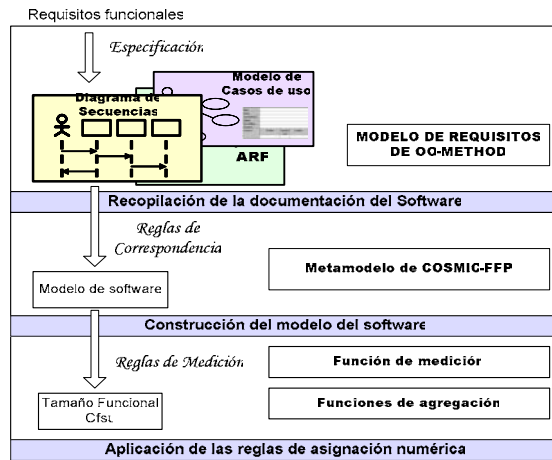


Figura 20. Aplicación de un procedimiento de medición

Con el fin de ilustrar el proceso general de la aplicación de RmFFP, se adaptó la segunda fase del modelo de proceso de medición, propuesto por Jacquet y Abran [34], tal como se ilustra en la Figura 21.



**Figura 21.** Proceso general de la aplicación de RmFFP, adaptado de [34]

Cada uno de estos pasos es explicado con mayor detalle en la siguiente sección.

## 5.2. El proceso de la aplicación de RmFFP

Con la finalidad de facilitar la aplicación de RmFFP, tres pasos generales fueron tomados en cuenta:

1. *La recopilación de la documentación del software*, donde la calidad de la documentación es un aspecto importante a considerar para facilitar la comprensión del software a medir, así como contribuir a la calidad de los resultados de medición.

2. La construcción del modelo del software, donde la calidad de dicho modelo influirá en su cuantificación, mediante el siguiente paso.
3. La aplicación de las reglas de asignación numérica, que permite obtener el tamaño funcional del sistema.

### 5.2.1. Recopilación de la documentación del software

Desde la perspectiva de los métodos de medición de tamaño funcional, el aspecto de interés es la funcionalidad, es decir “qué es lo que el software debe hacer”. Esta funcionalidad puede estar documentada por artefactos de ingeniería de software que son producidos antes de su implementación; o esta documentación puede no estar disponible. En tal caso, los requisitos funcionales pueden ser obtenidos a partir de artefactos que ya hayan sido implementados. La Figura 22 ilustra estos dos modos de documentación de la funcionalidad del sistema.

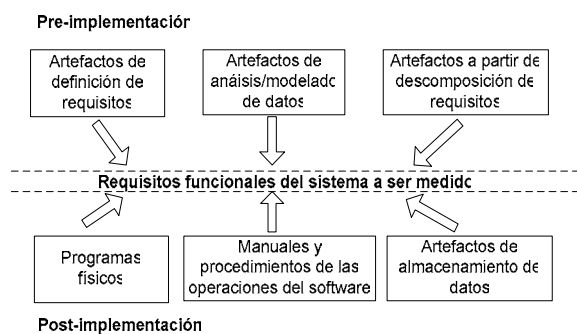


Figura 22. Modos de documentación de requisitos funcionales [24]

Con respecto a RmFFP, la funcionalidad del sistema es especificada de manera semi-formal usando el Modelo de Requisitos de OO-Method, consecuentemente RmFFP mide una especificación antes de que el sistema haya sido implementado, es decir modo Pre-implementación. Tal como se explicó en el capítulo 3, la fase de ingeniería de requisitos de OO-Method

culmina con la obtención semiautomática de los diagramas de secuencia a partir del modelo de casos de uso [106]. Estos diagramas de secuencia vienen a constituirse en el artefacto clave sobre el cual se llevará a cabo la medición. De este modo, la calidad de la especificación de estos diagramas de secuencia afectará a la calidad de los resultados de la medición.

El estándar IEEE 830 [107] describe un conjunto de características que debieran ser consideradas para una buena especificación de requisitos de software. Teniendo en cuenta estas características, OO-Method contribuye a que la especificación sea trazable, consistente, no ambiguo, y modificable. A continuación, se explica estas características y su influencia en la calidad de la medición del tamaño funcional.

*Trazable:* ya que es posible determinar el impacto que se ocasiona en el Modelo de Diagramas de Secuencia debido a los cambios realizados en el Modelo de Casos de Uso y viceversa. Esta trazabilidad hace posible obtener el tamaño funcional con mayor *exactitud* a partir de una etapa temprana. Es decir, el grado de proximidad entre el tamaño de la aplicación final y el tamaño obtenido en la especificación de los requisitos funcionales será mayor.

*Consistente:* ya que la deducción y especificación de los elementos de cada diagrama de secuencia a partir del Modelo de Casos de Uso es siempre llevada a cabo usando el mismo conjunto de criterios previamente definidos. Como estos criterios son independientes de razonamientos subjetivos, el tamaño funcional obtenido no se verá afectado por los diferentes niveles de detalle que podrían tener dichos diagramas si fueran estos obtenidos manualmente. Consecuentemente, la consistencia también favorecerá a la exactitud y precisión del tamaño funcional.

*No ambiguo:* El uso de una herramienta para la especificación semi-formal de los requisitos funcionales reduce su ambigüedad; favoreciendo de esta manera a que la comprensibilidad del software a medir sea mayor.

*Modificable:* Gracias a la transformación de modelos en la que está basada OO-Method, los posibles cambios que pudieran surgir hacen que sean fácilmente llevados a cabo de manera consistente. Esta característica afecta positivamente a la obtención del tamaño funcional.

Por lo tanto, la calidad de la documentación del software usando el Modelo de Requisitos de OO-Method favorece positivamente tanto a la comprensión del software a medir como a la exactitud y precisión de los resultados de la medición.

### 5.2.2. *Construcción del modelo del software*

Una vez recopilada la documentación apropiada, tal como se observa en la Figura 21, la construcción del modelo del software es llevada a cabo mediante la instanciación del metamodelo de COSMIC-FFP en un contexto específico de desarrollo, OO-Method. Este metamodelo ha sido descrito en el capítulo 4, (ver Figura 9).

El modelo del software es construido mediante la aplicación de un conjunto de reglas de representación, las cuales fueron definidas con el fin de abstraer las adecuadas primitivas de OO-Method (Ver Figura 7) de acuerdo al metamodelo de COSMIC-FFP. Sin embargo, al momento de aplicar estas reglas surgen preguntas con respecto a qué concepto del metamodelo debe ser primero representado o qué conceptos pueden ser representados simultáneamente. Para responder a estas preguntas, un diagrama de actividades UML fue elaborado con el fin de representar la secuencia de pasos del procedimiento de medición RmFFP.

Tal como se observa en la Figura 23, RmFFP inicia con la *definición del contexto de medición*, el cual incluye tres actividades: la identificación del propósito, el punto de vista y el alcance de la medición, que permitirá identificar la funcionalidad a ser incluida en la medición. Luego, se inicia la fase de representación, cuyo propósito es *construir el modelo del software a medir*. Para esto, tal como se muestra en la Figura 23, se aplica un conjunto de reglas en cada actividad. Esta fase culmina con la identificación de movimientos de datos, los cuales se constituyen en los componentes básicos que contribuyen a la obtención del tamaño funcional. El proceso de RmFFP culmina con la Fase de medición, la cual consiste en cuantificar el modelo de software construido previamente.

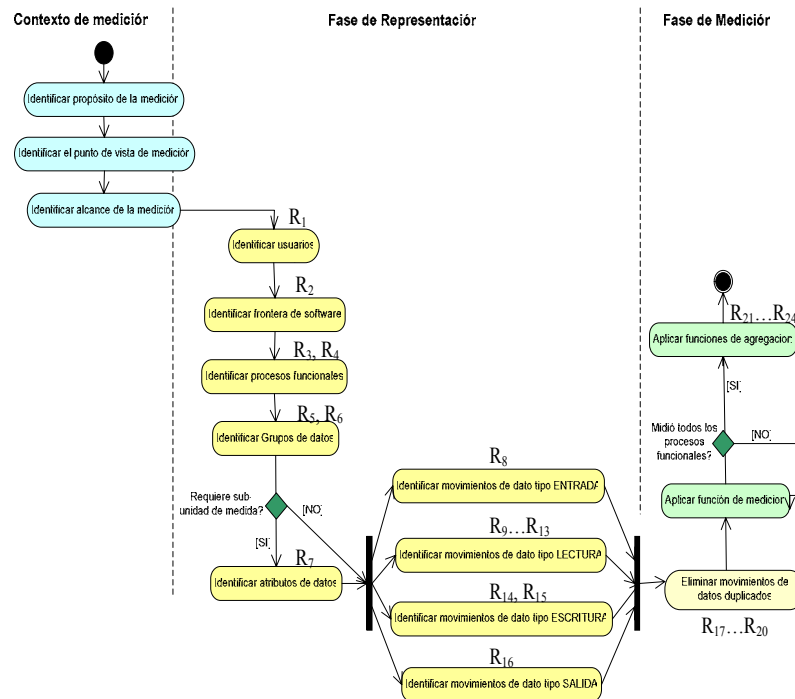


Figura 23. Proceso del procedimiento de medición RmFFP

A continuación describimos cada una de las actividades que corresponden al contexto de medición y fase de representación.

#### ***5.2.2.1. Identificar el propósito de la medición***

La identificación del propósito de la medición es la primera actividad a llevar a cabo. Es necesaria para definir el porqué de la medición y cuál será la utilidad del resultado obtenido. En base a este propósito, tanto el punto de vista como el alcance de medición son identificados.

#### ***5.2.2.2. Identificar el punto de vista de medición***

El propósito de esta actividad es determinar el nivel de detalle con el que la especificación de requisitos (artefacto a medir) puede ser vista y por ende medida. En RmFFP, tal como se explicó en el capítulo 3, el punto de vista de medición identificado es la del analista. Bajo este punto de vista se visualizará toda la funcionalidad que ha sido especificada, lo cual nos facilitará identificar el alcance de la medición.

#### ***5.2.2.3. Identificar el alcance de la medición***

Tal como se observa en la Figura 23, esta actividad es llevada a cabo una vez que el propósito y el punto de vista de medición hayan sido identificados. El resultado de ejecutar esta actividad es el conjunto de requisitos funcionales que serán incluidos en la medición, lo cual se constituye como artefacto de entrada para la fase de representación, que se inicia con la identificación de usuarios.

#### ***5.2.2.4. Identificar usuarios***

El propósito de esta actividad es identificar qué o quienes se comunican o interactúan con el software a medir. Su identificación facilita la identificación de la frontera y de los procesos funcionales.

#### ***5.2.2.5. Identificar la frontera del software***

El propósito de esta actividad es permitir distinguir sin ambigüedad la parte del software a ser medida.

#### ***5.2.2.6. Identificar procesos funcionales***

Para la identificación de los procesos funcionales es necesario haber identificado previamente la frontera, ya que los usuarios están fuera de la frontera, y éstos son los responsables de interactuar con algún proceso funcional. La aplicación de las reglas R3 y R4 permiten llevar a cabo esta actividad.

#### ***5.2.2.7. Identificar grupos de datos***

El propósito de esta actividad es identificar los datos que serán manipulados dentro de un proceso funcional. Esta actividad contribuye a la identificación de movimientos de datos que son implicados en un proceso funcional. La aplicación de las reglas R5 y R6 permiten la identificación de estos grupos de datos.

#### ***5.2.2.8. Identificar atributos de datos***

Tal como se observa en la Figura 23, esta actividad es optativa. La aplicación de la regla R7 dependerá de si la medición requiere ser llevada a cabo a un nivel de sub-unidades.

#### ***5.2.2.9. Identificar movimientos de datos***

Tal como se muestra en el diagrama de actividades (Figura 23), la identificación de los tipos de movimientos de datos (entrada, lectura, escritura y salida) es llevada a cabo a través de cuatro actividades que pueden ser efectuadas una vez que los procesos funcionales y los grupos de datos o atributos de datos hayan sido identificados. La aplicación de nueve reglas (R8-



R16) permite la identificación de estos tipos de movimientos de datos, los cuales se constituyen en los componentes básicos del modelo del software a medir en la siguiente fase.

### **5.2.3. Aplicación de las reglas de asignación numérica**

Tal como se ilustra en la Figura 21, una vez construido el modelo del software, el siguiente paso general es la aplicación de las reglas de asignación numérica. Para esto, tres actividades son llevadas a cabo, las cuales son descritas a continuación.

#### **5.2.3.1. Eliminar movimientos de datos duplicados**

El propósito de esta actividad es verificar si los movimientos de datos han sido identificados correctamente. Para esto, todos los movimientos de datos identificados como duplicados son eliminados mediante la aplicación de cuatro reglas (R17-R20) que fueron definidas en el anterior capítulo.

#### **5.2.3.2. Aplicar función de medición**

El propósito de esta actividad es cuantificar los movimientos de datos correctamente identificados en todos los procesos funcionales. Para esto, la función de medición, que es conforme con el método estándar COSMIC-FFP, es aplicada.

#### **5.2.3.3. Aplicar funciones de agregación**

La presente actividad permite obtener un valor cuantitativo (tamaño funcional) a nivel de proceso funcional y a nivel de capa. Para esto, las reglas (R21-R24) relacionadas a las funciones de agregación son aplicadas, siendo el tamaño funcional expresado en unidades Cfsu (Cosmic functional size unit).

A continuación se ilustra la aplicación de RmFFP en un caso de estudio.

### 5.3. Aplicando RmFFP en un caso de estudio

El caso de estudio utilizado trata sobre una compañía ubicada en un área turística que necesita automatizar la gestión de alquiler de vehículos y otras actividades derivadas tales como el mantenimiento y reparación de vehículos, alquiler de extras (teléfono, chofer, etc.), compra y venta de vehículos.

La especificación de los requisitos funcionales de dicho caso de estudio es presentada en el Anexo A. Este caso de estudio fue documentado de manera semi-formal, mediante la definición de la misión de sistema, la construcción del árbol de refinamiento de funciones, la obtención del diagrama de casos de uso y del diagrama de secuencias.

A partir de esta documentación, se procedió con la *construcción del modelo del sistema* “alquiler de vehículos” aplicando las siguientes actividades:

- *Identificar propósito de la medición*

El propósito es medir el tamaño funcional de los requisitos del sistema “alquiler de vehículos”, el cual será generado automáticamente con el método OO-Method.

- *Identificar punto de vista de la medición*

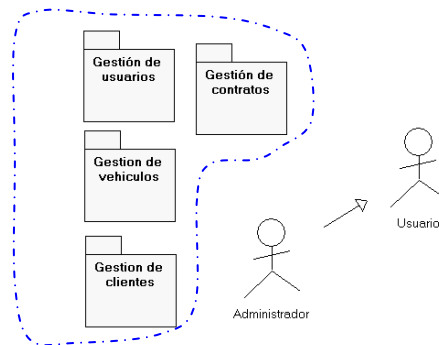
El punto de vista de medición es el del analista.

- *Identificar alcance de la medición*

El alcance de la medición es comprendido por todos los requisitos funcionales especificados para el desarrollo del sistema “alquiler de vehículos”.

- *Identificar usuarios*

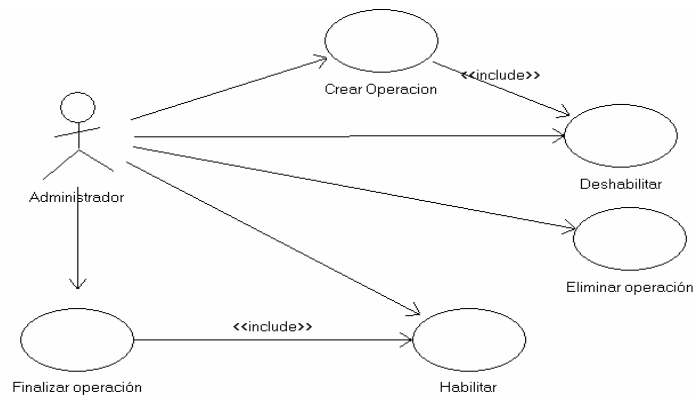
Aplicando la regla 1, los actores del diagrama de casos de uso que se muestra en la Figura 24, son identificados como usuarios.



**Figura 24.** Diagrama de casos de uso general

- *Identificar frontera*

Aplicando la regla 2, la frontera del sistema es el borde identificado entre los usuarios y los grupos funcionales: gestión de usuarios, gestión de contratos, gestión de vehículos y gestión de clientes (vea Figura 24). Cada grupo funcional consta de subgrupos funcionales, y éstos de casos de uso. En la Figura 25 se muestra al sub-grupo funcional Gestión de Operaciones que pertenece al grupo funcional Gestión de vehículos.



**Figura 25.** Sub grupo funcional: Gestión de Operaciones

- *Identificar procesos funcionales*

Aplicando la regla 3, todos los casos de uso primarios que aparecen en el árbol de refinamiento de funciones (Ver Figura 1 del Anexo A) son considerados como procesos funcionales. Asimismo aplicando la regla 4, los casos de uso secundarios, (asignación de extras, deshabilitar, habilitar, crear seguro, vender, entregar, eliminar seguro) son identificados también como procesos funcionales. En el sistema alquiler de vehículos se identificó un total de treinta y cinco procesos funcionales.

- *Identificar grupos de datos*

Aplicando la regla 5, los actores Administrador y Usuario son identificados como grupos de datos. Además, aplicando la regla 6, las clases de cada uno de los diagramas de secuencia son identificados como grupos de datos: coche, precio, venta, garaje, seguro, compañía de seguro, operación, cliente, contrato, contrato extras, tipo de extras.

- *Identificar movimientos de datos*

Aplicando las nueve reglas (R8-R16), los diferentes tipos de movimientos de datos fueron identificados principalmente en los mensajes de los diagramas de secuencia. A continuación, la aplicación de estas reglas son ilustradas únicamente para el caso de uso “crear operación” (ver Figura 25).

Como cada caso de uso (proceso funcional) es representado por uno o más diagramas de secuencia, la Figura 26 muestra la funcionalidad interna del caso de uso “crear operación”.

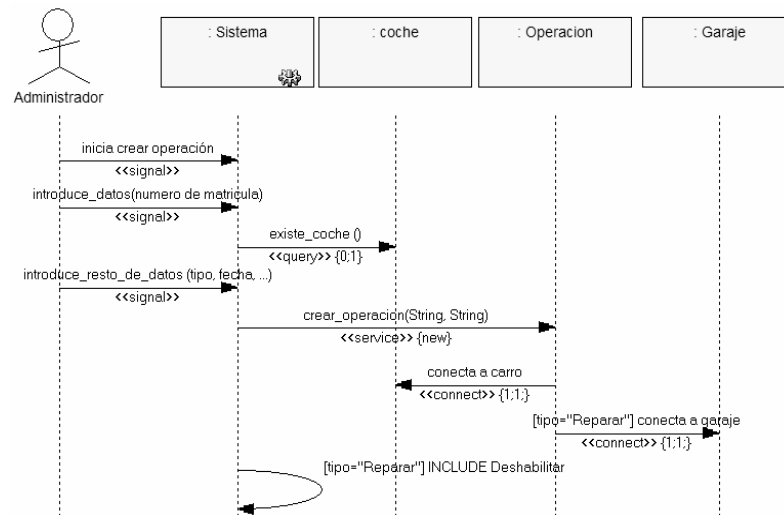


Figura 26. Diagrama de secuencia para “Crear operación”

El escenario se inicia con la solicitud de crear la operación realizada por medio del mensaje signal “crear operación”. Este mensaje no es identificado como movimiento de datos ya que no hay datos involucrados en dicho mensaje. Sin embargo, aplicando la regla 8, un movimiento de dato de tipo ENTRADA es identificado en el mensaje “introduce datos” ya que el grupo de dato “coche” es involucrado en este mensaje. Luego, mediante el mensaje con el estereotipo *query*, el sistema verifica si el coche existe con el número de matrícula indicado; aplicando la regla 9, este mensaje es identificado como un movimiento de datos de tipo LECTURA.

Si el coche existe, el resto de datos es introducido por medio del mensaje “introduce resto de datos”, en el cual, aplicando nuevamente la regla 8, se identifican tres movimientos de datos de tipo ENTRADA, ya que el mensaje involucra atributos de tres grupos de datos: coche, operación y garaje.

Aplicando la regla 14, un movimiento de tipo ESCRITURA es identificado en el mensaje “crear\_ operación”, el cual permite crear una

operación llevada a cabo sobre un coche; y la misma que será efectuada en un determinado garaje si la operación a realizar es de reparación o mantenimiento. Esto es representado por una condición del mensaje “conecta a garaje”, el cual, aplicando la regla 10, es identificado como un movimiento de dato de tipo LECTURA.

Por último, tal como se observa en la Figura 26, se invoca al caso de uso “Deshabilitar” sólo si el coche requiere ser reparado. Esto implica cambiar su estado a deshabilitado.

- *Eliminar movimientos de datos duplicados*

En el mensaje “introduce\_resto\_de\_datos” se identifican tres movimientos de tipo ENTRADA, ya que tres grupos de datos son involucrados en dicho mensaje. Sin embargo, aplicando la regla 16, un movimiento de dato es eliminado debido a la duplicidad que existe con el mensaje “introduce\_datos”, puesto que este mensaje ya incluye al grupo de dato “coche”.

- *Aplicar la función de medición*

Se asigna el valor de 1 Cfsu a cada movimiento de datos identificado en los respectivos procesos funcionales.

$$f(x) = 1Cfsu \quad \text{Donde } x \text{ es un tipo de movimiento de datos}$$

Finalmente, para obtener el tamaño funcional del sistema, la agregación de los movimientos de datos cuantificados es llevada a cabo tanto a nivel de proceso funcional como a nivel de capa.

- *Aplicar la función de agregación a nivel de proceso funcional*

Para ilustrar la aplicación de las reglas relacionadas con la agregación a nivel de proceso funcional, retomamos el ejemplo del caso de uso “crear operación”.

En primer lugar, una vez cuantificados los tipos de movimientos de datos que ocurren en el proceso funcional “Crear operación”, aplicamos la regla 20, que es expresada en la fórmula (1), para obtener un tamaño parcial de 6 Cfsu.

$$Tamaño_p(crear\_operacion) = \sum_{h=1}^m f(entrada)_h + \sum_{i=1}^n f(lectura)_i + \sum_{j=1}^p f(escritura)_j \quad (1)$$

$$Tamaño_p(crear\_operacion) = 3 + 2 + 1$$

$$Tamaño_p(crear\_operacion) = 6 \text{ Cfsu}$$

En segundo lugar, como el caso de uso base “crear operación” (CUB) incluye al caso de uso secundario “deshabilitar” (CUS), la regla 22 es aplicada para calcular el tamaño funcional total del caso de uso base. Esta regla es expresada en la fórmula (2). El tamaño del caso de uso “deshabilitar” fue de 3 Cfsu, el cual fue obtenido mediante la aplicación de RmFFP. Reemplazando estos valores, se obtiene que el tamaño funcional para el caso de uso “crear operación” es de 9 Cfsu.

$$Tamaño(CUB) = \sum_{i=1}^n Tamaño(CUS_i) + Tamaño_p(CUB) \quad (2)$$

$$Tamaño(crear\_operacion) = 3 + 6$$

$$Tamaño(crear\_operacion) = 9 \text{ Cfsu}$$

- *Aplicar la función de Agregación a nivel de capa*

Como el alcance de la presente medición es conformada por toda la funcionalidad del sistema incluida en una sola capa, el tamaño total del sistema es obtenido aplicando la regla 23, expresada en la fórmula (3).

$$\text{Tamaño}(capa)_1 = \sum_{i=1}^{28} \text{Tamaño}(\text{Caso\_de\_uso\_primario}_i) \quad (3)$$

Por lo tanto, se obtuvo un tamaño de 124 Cfsu, como resultado de sumar los tamaños funcionales de los 28 casos de uso primarios del sistema “Alquiler de vehículos”.

La Tabla 5 resume los resultados obtenidos aplicando las diferentes reglas previamente definidas. La primera columna lista el conjunto de casos de uso primarios y secundarios, los cuales fueron identificados como procesos funcionales. Las siguientes cuatro columnas representan la cantidad de movimientos de datos de tipo entrada, lectura, escritura y salida que ocurren en cada proceso funcional. En las tres últimas columnas se representa los valores obtenidos al aplicar las reglas relacionadas a las funciones de agregación. Así, los valores de la antepenúltima columna representan al tamaño parcial del proceso funcional sin considerar las relaciones con otros casos de uso. En cambio, la siguiente columna indica el tamaño adicional que representa a todos los casos de uso secundarios que están relacionados con el caso de uso base. Por ejemplo, el caso de uso “Comprar” que se muestra en la Figura 27, está relacionado directamente con tres casos de uso secundarios (vender, crear seguro y deshabilitar). El valor 22 representa el tamaño total de estos 3 casos de uso secundarios.

Finalmente, la última columna representa el tamaño total del sistema, el cual es resultado de considerar únicamente a los casos de uso primarios.



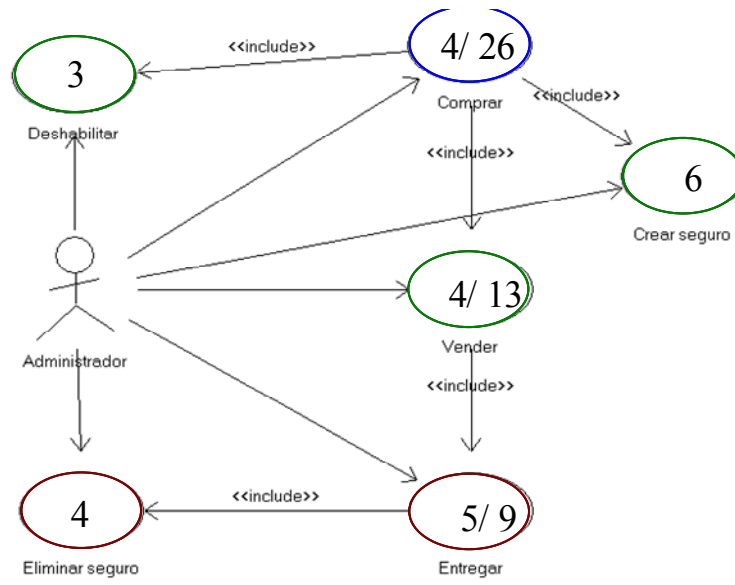


Figura 27. Tamaño funcional de los casos de uso Gestión de vehículos

Tabla 5. Medición del Sistema de Alquiler de vehículos

MOVIMIENTOS DE DATOS		ENTRADA	LECTURA	ESCRITURA	SALIDA	Size : R20	Size: R21,R22	Funcional Size
ID	PROCESOS FUNCIONALES							
1	Comprar	2	1	1		4	22	26
2	Vender	2	1	1		4	9	
3	Entregar	2	1	2		5	4	
4	Crear tarifa	1	1	1		3		3
5	Eliminar tarifa	1	1	1		3		3
6	Modificar tarifa	1	1	1		3		3
7	Crear garaje	1	1	1		3		3
8	Elimina garaje	1	1	1		3		3
9	Modificar garaje	1	1	1		3		3
10	Crear seguro	3	1	2		6		
11	Eliminar seguro	1	1	2		4		
12	Modificar seguro	1	1	1		3		3
13	Crear compañía de seguro	1	1	1		3		3
14	Eliminar compañía de seguro	1		1		2		2
15	Modificar compañía de seguro	1	1	1		3		3
16	Crear operación	3	2	1		6	3	9
17	Deshabilitar	1	1	1		3		
18	Habilitar	1	1	1		3		
19	Eliminar operación	1		1		2		2
20	Finalizar operación	1	1	1		3	3	6
21	Crear cliente	1	1	1		3		3
22	Eliminar cliente	1	1	1		3		3
23	Modificar cliente	1	1	1		3		3
24	Alquilar	3		3		6	5	11
25	Modificar contrato	1	1	1		3		3
26	Devolver	1	4	4	1	10		10
27	Crear Tipo de extra	1	1	1		3		3
28	Eliminar tipo de extra	1		1		2		2
29	Modificar tipo de extra	1	1	1		3		3
30	Asignar extras	3		2		5		
31	Crear usuario	1	1	1		3		3
32	Eliminar usuario	1		1		2		2
33	Modificar usuario	1		1		2		2
34	Ascender	1		1		2		2
35	Destituir	1		1		2		2
		<b>CAPA 1</b>						<b>124</b>

## 5.4. Conclusiones

En este capítulo se describió un conjunto de pasos que permite guiar la aplicación del procedimiento de medición RmFFP. Estos pasos son llevados a cabo a lo largo de 3 etapas generales del modelo genérico del proceso de medición propuesto por Abran y Jacquet [34]:

1. recopilación de la documentación de software,
2. construcción del modelo de software
3. aplicación de las reglas de asignación numérica.

En cuanto a la *recopilación de la documentación del software*, se observó que toda especificación de requisitos obtenida con OO-Method, se caracteriza por ser principalmente trazable, consistente, no ambigua y modificable; características que influyen de manera positiva tanto a la comprensión del software a medir como a la exactitud y precisión de los resultados de la medición.

En cuanto a la *construcción del modelo del software*, se describió una secuencia de pasos que permitieron definir el contexto de medición e instanciar el metamodelo de COSMIC-FFP en base a la aplicación de las reglas de representación. Por consiguiente, la calidad del modelo de software a medir dependerá de la calidad de la especificación de los requisitos funcionales y de la calidad de las reglas de representación, definidas en el capítulo anterior.

En cuanto a la *aplicación de las reglas de asignación numérica*, se describió una serie de pasos que permitieron eliminar los posibles movimientos de datos duplicados, cuantificar dichos movimientos y aplicar las funciones de agregación con el fin de obtener un tamaño funcional.

Finalmente, se ilustró el proceso de aplicación de RmFFP en un caso real, con el fin de facilitar su comprensión. Sin embargo, cabe mencionar que este procedimiento de medición fue también aplicado varias veces a otros casos de estudio, tales como: Gestión de campo de Golf y Gestión de Servicios de Mantenimiento de un Hospital, lo que contribuyó a la evolución de RmFFP.



## *Validación del diseño*

*“Las proposiciones matemáticas, en cuanto tienen que ver con la realidad, no son ciertas; y en cuanto que son ciertas, no tienen nada que ver con la realidad”.*

*- Albert Einstein*

Conscientes de que las medidas de software juegan un rol importante para asegurar tanto la calidad del proceso como del producto en si, es necesario que estas medidas sean fiables para que puedan ser utilizadas por los administradores, programadores, etc. Sin embargo, a pesar de existir una gran cantidad de “*métricas de software*”, éstas en su mayoría no han sido utilizadas en la práctica. Una de las razones, justamente ha sido la falta de un marco de validación que permita validar estas medidas propuestas.

Sellami y Abran identificaron varios tipos de validación que debieran ser considerados para validar los distintos tipos de conceptos relacionados a todo el proceso de medición: Validación del diseño del método de medición, validación del instrumento de medición, validación de los resultados de medición y validación de la utilización de estos resultados [88].

La validación del diseño consiste en comprobar que el método de medición realmente mida lo que deba medir. Es decir, se trata de comprobar si un método de medición captura correctamente los atributos de interés a ser

medidos para ser representados numéricamente. En la literatura, este tipo de validación ha sido comúnmente referida como validación teórica, la cual está basada en la satisfacción de la “condición de representación”. Este tipo de condición es satisfecha si una estructura empírica, que cumple con una serie de axiomas o condiciones, puede ser utilizada para construir una representación dentro de los números reales [89]. La satisfacción de esta condición no es suficiente para validar el diseño de un método de medición. Sin embargo, el diseño de varios métodos o procedimientos de medición de tamaño funcional han sido validados únicamente desde el punto de vista de la teoría de la medición, entre ellos tenemos a Zuse [91], Fetcke [92], Abrahão [93] y Poels [68].

El presente capítulo tiene como fin validar el diseño del procedimiento de medición RmFFP. Para esto, a continuación se describe los aspectos que se consideraron para llevar a cabo esta validación.

### **6.1. Alcance de la validación del diseño**

Teniendo en cuenta los cuatro pasos de la fase de diseño del Modelo de Proceso de Medición [34]:

- 1) definición de objetivos,
- 2) selección del metamodelo para representar el software a medir,
- 3) caracterización del concepto a ser medido y
- 4) definición de reglas de asignación numérica.

La validación teórica (teoría de la medición [91]) como se mencionó anteriormente no es suficiente para validar el diseño de un método o procedimiento de medición, ya que sólo involucra a la conjunción de los pasos 3 y 4. Es decir, la validación teórica consigue que las reglas de asignación numérica representen adecuadamente la caracterización empírica del atributo

a medir. Sin embargo, debido a que hay otros aspectos a considerar, tales como: asegurar que el atributo a medir sea correctamente caracterizado, que el modelo del software a medir sea correctamente representado y que las reglas de asignación numérica sean correctamente definidas, se consideró identificar previamente el alcance de la validación del diseño de RmFFP. Para esto, se incluyó:

- Una evaluación de conformidad con el estándar ISO/IEC 19761 [22], ya que RmFFP es un procedimiento de medición que comprende de un conjunto de reglas que han sido definidas para instanciar el metamodelo de COSMIC-FFP y cuantificar el modelo construido. La evaluación de conformidad involucra a los tres últimos pasos del Modelo de Proceso de Medición.
- Un análisis de RmFFP desde una perspectiva de los conceptos de la metrología [1], el mismo que involucrará a los pasos de caracterización del concepto a medir y definición de las reglas de asignación numérica.
- Finalmente, una validación teórica basada en la teoría de la medición, que como se dijo involucra a la conjunción de los pasos 3 y 4.

La Figura 28 muestra el alcance de la validación del diseño de RmFFP.



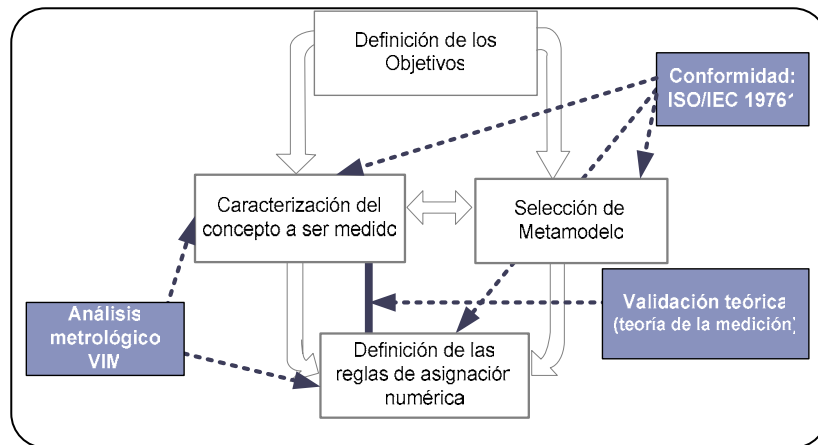


Figura 28. Alcance de la validación del diseño

En las siguientes secciones, cada uno de estos puntos es descrito con mayor detalle.

## 6.2. Evaluación de conformidad

El propósito de la presente evaluación de conformidad es determinar si el procedimiento de medición RmFFP es conforme con el método estándar ISO/IEC 19761:2003. De este modo, el conjunto de reglas que han sido definidas para representar los conceptos del metamodelo de COSMIC-FFP en las respectivas primitivas del modelo de requisitos de OO-Method son evaluadas en cuanto a su completitud. Es decir, si el procedimiento de medición RmFFP contempla todos los conceptos fundamentales del método estándar COSMIC-FFP.

Para llevar a cabo esta evaluación, algunos elementos enumerados a continuación han sido considerados.

*En cuanto a la documentación a evaluar:* el procedimiento RmFFP comprende de una guía que resume el conjunto de reglas a ser aplicadas (ver Anexo B). Esta documentación será utilizada por los evaluadores.

En cuanto al *instrumento de evaluación a utilizar*: Se diseñó una lista de preguntas para facilitar el proceso de evaluación de conformidad. Esta lista comprende de 12 partes que corresponden a los fundamentos de COSMIC-FFP. La onceava y doceava parte de este listado son complementarias, por lo que no son determinantes para la evaluación de la conformidad de RmFFP. Cada parte consta de una o más preguntas, las cuales son respondidas en función a su satisfacción mediante respuestas simples: “sí” o “no”. Esta lista de preguntas es presentada en el Anexo C.

En cuanto al *evaluador*: con el propósito de incrementar la objetividad y consistencia de la evaluación de conformidad, un grupo de tres personas expertas en medición de software y certificadas en COSMIC-FFP fueron los responsables de verificar la evaluación realizada.

#### **6.2.1. Proceso de evaluación**

El proceso llevado a cabo para la evaluación de conformidad del procedimiento RmFFP ha sido adaptado del estándar ISO 14143-2 [17]. Tal como se observa en la Figura 29, las preguntas de evaluación que son complementarias son registradas como preguntas que no contribuyen a la evaluación. En caso de que la pregunta no es complementaria, el evaluador busca información relevante en la documentación de RmFFP relacionada a la pregunta; si no es encontrada la información, se registra como ausencia de información y su justificación. En caso de encontrar información relacionada a la pregunta, se registra la ubicación o ubicaciones de toda la información relevante. Una vez localizada la información, el evaluador decide si esta información satisface a la pregunta; si no satisface a la pregunta, se registra la pregunta como “no aprobada” y se justifica las razones de la decisión. En caso de que es satisfecha, el evaluador registra la pregunta como “aprobada”; si la pregunta no es resuelta, ella es registrada como “no pudo ser resuelta” y se justifica las respectivas razones de esta decisión.

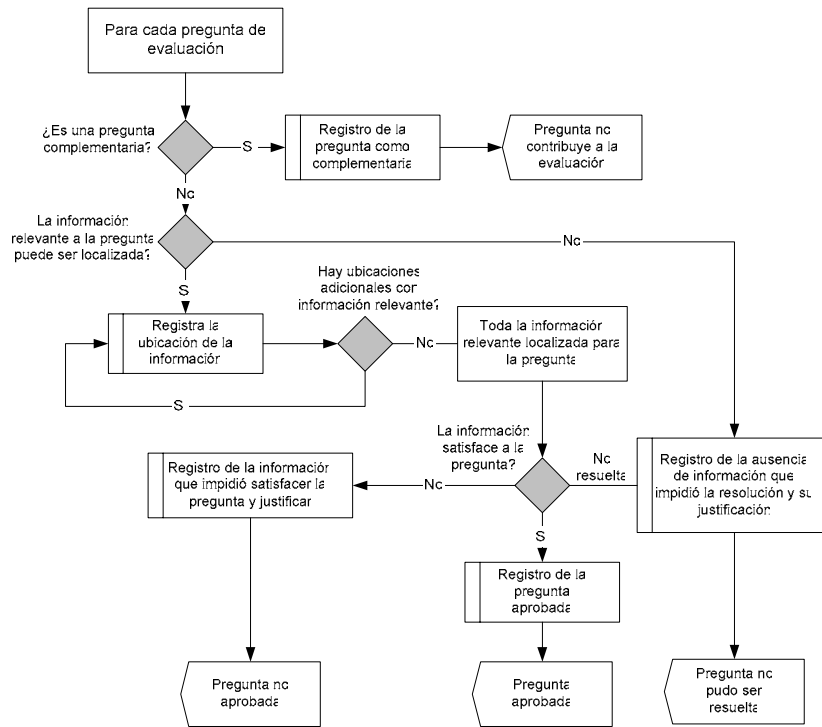


Figura 29. Proceso de análisis para cada pregunta, Adaptado de [17].

A continuación se presenta los resultados de conformidad de RmFFP obtenidos.

### 6.2.2. Resultado de la evaluación de conformidad

Llevada a cabo la evaluación de conformidad, las respuestas obtenidas para cada una de las preguntas del cuestionario diseñado (Vea Anexo C) son presentadas en la En la Tabla 6.

En esta tabla se observa que la mayoría de las preguntas relacionadas a cada una de las partes contempladas en la ISO/IEC 19761 fueron registradas como *aprobadas*, a excepción de las siguientes partes que fueron observadas ya que no fueron satisfechas:

Parte 4: Identificación de las capas de software.- de tres preguntas una no fue satisfecha debido a que en RmFFP no se identificaron capas subordinadas por estar conformada únicamente por una capa (Requisito 6.4.2.b). Sin embargo, cabe mencionar que este concepto no es representado en ninguna primitiva específica del modelo de requisitos de OO-Method.

Parte 6: Identificación de procesos funcionales.- de cuatro preguntas una no fue satisfecha debido a que en RmFFP los eventos disparadores no fueron considerados para la identificación de procesos funcionales (Requisito 6.6.a). Sin embargo, estos procesos funcionales son identificados por los actores, que son los responsables de ejecutar de manera directa o indirecta un evento disparador.

Parte 11: Personalización local.- no fue contemplada en RmFFP debido a que no fue necesaria para la medición de los requisitos funcionales (Requisito 7). Esta parte es considerada como complementaria.

Por lo tanto, se concluye que el procedimiento de medición de tamaño funcional conocido como RmFFP versión 1.0 es conforme a los conceptos y principios de la ISO/IEC 19761:2003.

**Tabla 6.** Respuestas de la evaluación de conformidad

Parte	Pregunta	Satisface?
1. Unidad	1.1	Si
	1.2	Si
2. Propósito y alcance	2.1	Si
	2.2	Si
3. Identificación de requisitos funcionales	3.1	Si
4. Identificación de capas de software	4.1	Si
	4.2	Si
	4.3	No

Parte	Pregunta	Satisface?
5. Identificación de fronteras de software	5.1	Si
6. Identificación de procesos funcionales	6.1	Si
	6.2	No resuelta
	6.3	Si
	6.4	Si
7. Identificación de grupos de datos	7.1	Si
	7.2	Si
8. Identificación de movimientos de datos	8.1	Si
	8.2	Si
9. Clasificación de movimientos de datos	9.1	Si
	9.2	Si
	9.3	Si
	9.4	Si
10. Obtención del tamaño funcional	10.1	Si
	10.2	Si
	10.3	Si
	10.4	Si
11. Personalización local	11.1	No resuelta
12. Reporte de la medición	12.1	Si
	12.2	Si

Estos resultados fueron verificados por 3 personas certificadas en COSMIC-FFP.

### 6.3. Análisis metrológico del diseño

El propósito de la presente sección es analizar el diseño de RmFFP en base a un conjunto de conceptos del VIM. Estos conceptos son introducidos a continuación.

#### 6.3.1. Analizando un modelo sobre conceptos de metrología

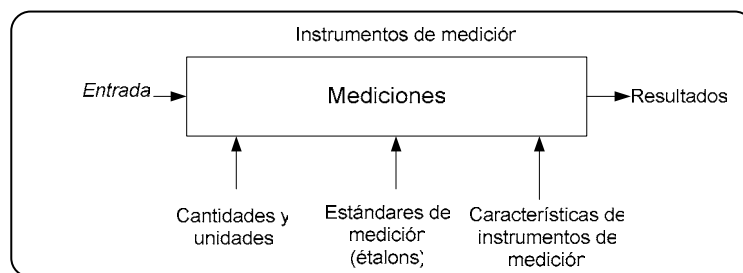
Un modelo sobre los conceptos de metrología, que son definidos en el VIM[1], fue propuesto por Sellami y Abran con el propósito de mejorar la

presentación y comprensión de dichos conceptos y las inter-relaciones existentes [11].

Este documento ISO presenta 120 conceptos, los cuales fueron organizados en 6 categorías:

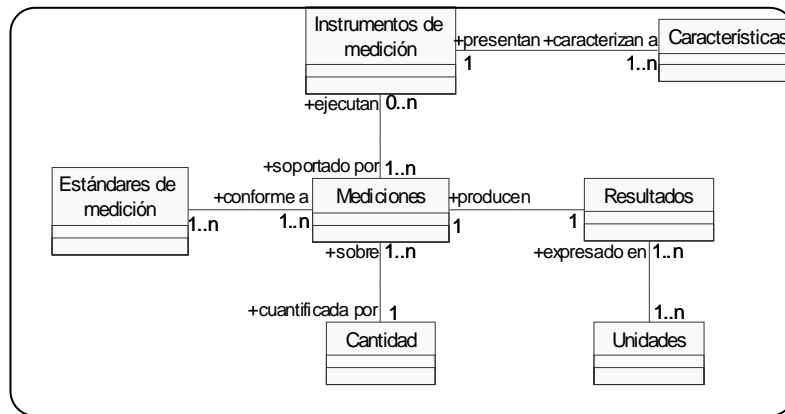
1. cantidades y unidades,
2. mediciones,
3. resultados de medición,
4. instrumentos de medición,
5. características de instrumentos de medición
6. estándares de medición.

Sellami y Abran consideran estas 6 categorías como un modelo de alto nivel, cuya representación fue similar a la de un proceso de producción: entrada, salida, variables de control y el proceso en sí (Vea Figura 30).



**Figura 30.** Modelo de proceso de las categorías de metrología [11]

La Figura 31 representa el metamodelo de los conceptos que fueron considerados como “categorías”. Este metamodelo ha sido elaborado con el fin de enfatizar las relaciones de la Figura 30 mediante el uso de roles y cardinalidades.



**Figura 31.** Meta-modelo de los conceptos de alto nivel del VIM

Cada “categoría” fue modelada mediante una organización topológica de los respectivos conceptos involucrados [11]. Por ejemplo, los 22 conceptos que forman parte de la categoría “cantidades y unidades” fueron organizados en 4 sub-conjuntos: Dimensión, Sistemas de cantidades, Unidad de medición y Valores.

Para llevar a cabo un análisis metrológico del diseño de las medidas de tamaño del software, primero es necesario identificar qué categorías del VIM debieran ser considerados durante la fase de diseño y qué conceptos de estas categorías debieran ser tomadas en cuenta.

Con respecto a la primera interrogante, Sellami y Abran [88] realizaron un análisis inicial entre estas categorías del VIM [1] y las fases del Modelo de Proceso de Medición (vea Tabla 7), quienes identificaron que la fase de utilización de resultados no era contemplada por el VIM. Sin embargo, las fases de aplicación y análisis de resultados si eran contempladas con mayor profundidad que la fase de diseño. Esta fase fue correspondida únicamente con la categoría de “cantidades y unidades”.

**Tabla 7.** Categorías VIM y el Modelo de Proceso de Medición, ajustada de [88].

<b>Modelo del proceso de medición</b>	Diseño del método de medición	Aplicación del método de medición	Análisis de los resultados	Utilización de los resultados
<b>Categorías del VIM</b>	-Cantidades y unidades - Fundamentos de medición - Estándares de medición	-Proceso de Medición Instrumentos de Medición Características de instrumentos -Estándares de medición	-Resultados -Estándares de medición	Estándares de medición

A partir de esta correspondencia inicial, observamos que las categorías de “medición” y “estándares de medición” debieran ser también consideradas para la fase de diseño, ya que es necesario tomar en cuenta sobre qué *fundamentos de medición* se llevará a cabo el diseño de un procedimiento. Además, qué tipo de *estándar de medición*, (Ver Tabla 8), ha de ser considerado. Un estándar de medición es entendida en el VIM como una medida material, un instrumento de medición, o un material de referencia para definir, realizar, conservar o reproducir una unidad o uno o más valores de cantidad para servir como una referencia. La utilización de estos estándares en todas estas fases contribuirá a la calidad de las mediciones.

Por consiguiente, para un análisis del diseño de las medidas de tamaño del software bajo una perspectiva de la metrología, tres categorías del VIM debieran ser consideradas: *cantidades y unidades*, *fundamentos de la medición* y *estándares de medición*. Este análisis del diseño es llevado a cabo sobre la caracterización del concepto a ser medido y sobre la definición de las reglas de asignación numérica.

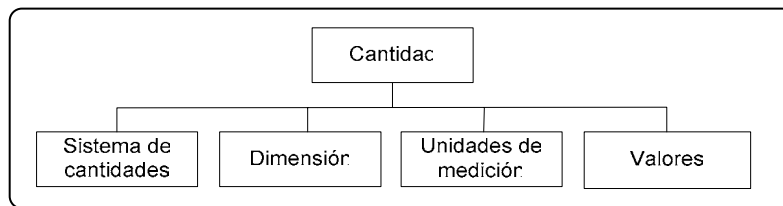


**Tabla 8.** Estándares de medición VIM [1]

Estándares de medición	Conservación de un estándar
Estándar Internacional	Calibración
Estándar Nacional	Material de referencia
Estándar Primario	Material de referencia certificado
Estándar Secundario	
Estándar de referencia	
Estándar de trabajo	
Estándar de transferencia	
Estándar con características portables	

Con respecto a la segunda interrogante, los conceptos a considerar para cada una de estas categorías identificadas son:

- *Cantidades y unidades:* los conceptos de esta categoría están organizados en cuatro sub-categorías: sistema de cantidades, dimensión, unidades de medición y valores (Vea Figura 32).



**Figura 32.** Topología de alto nivel de la categoría “cantidades y unidades” [11].

Tal como se observa en la **Tabla 9**, 17 conceptos están organizados dentro de estas subcategorías; de los cuales: 9 conceptos son apropiados para el análisis del diseño de medidas de tamaño, 8 conceptos no fueron considerados para este análisis, la razón es explicada con mayor detalle en la siguiente sección. Además, el concepto de “valor verdadero” fue considerado como indeterminado (I), debido a que no existe “medición perfecta”, más aún, en la medición del software, donde intervienen varios factores externos que podrían afectar a la obtención de este “valor verdadero”.

**Tabla 9.** Cantidades y unidades”: conceptos para el análisis de las medidas de tamaño

Sistema de cantidades		Unidad	
Cantidad base	√	Símbolo de unidad	√
Cantidad derivada	X	Unidad base	√
<b>Dimensión de una cantidad</b>		Unidad derivada	X
Cantidad de dimensión	X	Unidad coherente	X
<b>Valores</b>		<b>Sistema de unidades</b>	√
Valor verdadero	I	Sistema coherente de unid.	X
Valor verdadero convencional	√	Sistema internacional de unid.	X
Valor numérico	√	Sistema Off de unidad	X
Escala de referencia convencional	√	Múltiplos de una unidad	√
		Sub múltiplos de una unidad	√

- *Fundamentos de la medición*: dos conceptos debieran ser considerados para el diseño de las medidas de software: 1) “principio de la medición”, que representa la base científica de una medición. 2) “método de medición” que representa una secuencia lógica de operaciones utilizada en una medición.

- *Estándares de medición*: este concepto es entendido en el VIM como una medida material, un instrumento de medición, o un material de referencia para definir, realizar, conservar o reproducir una unidad o uno o más valores de cantidad para servir como una referencia.

### 6.3.2. *Análisis metrológico del diseño de RmFFP*

Un análisis del diseño de RmFFP en base a las categorías identificadas (cantidades y unidades, fundamentos de medición y estándares de medición) es descrito a continuación.

### 6.3.2.1. Cantidades y unidades

En cuanto al “*sistema de cantidades*”, conformada por el conjunto de “*cantidades base*” y “*cantidades derivadas*”. Una “*cantidad base*” es una cantidad independiente de cualquier otra cantidad<sup>1</sup>, y una “*cantidad derivada*” es una cantidad dependiente de otras cantidades base. El procedimiento RmFFP se enfoca únicamente en la medición del “tamaño funcional” considerada como una “*cantidad base*”. Cabe mencionar que las cantidades derivadas a partir del tamaño funcional (por ejemplo la productividad) no son contempladas por RmFFP.

En cuanto a la “*dimensión de una cantidad*”, expresión de una cantidad representada como un producto de potencias, cuyos factores son *cantidades base*. En RmFFP la dimensión es unidimensional, ya que está representada por un solo factor.

En cuanto a la “*unidad de medición*”, la “*unidad base*” es el movimiento de dato, cuyo “*símbolo*” es Cfsu. Los otros conceptos como múltiplos y submúltiplos de una unidad no han sido aún definidos. Sin embargo, surge la pregunta de si estos conceptos podrían ser definidos. Para responder a esta pregunta, previamente analizamos los conceptos relacionados a la sub-categoría de “*valores*”.

El “*valor verdadero*” es un valor consistente con la definición de una cantidad particular y sería obtenido por una medición perfecta. Por consiguiente, este valor verdadero es por naturaleza indeterminado, más aún en la medición del software, donde intervienen varios factores externos que podrían afectar a la obtención de este “*valor verdadero*”. Sin embargo, un “*valor*

---

<sup>1</sup> Atributo de un fenómeno, cuerpo o sustancia que puede ser distinguido cualitativamente y determinado cuantitativamente [1].

*verdadero convencional*” sí que es considerado a través del valor obtenido por algún experto. Para facilitar esto, el grupo COSMIC [94], a partir de este año, ha intensificado un proceso de certificación del método COSMIC-FFP. Actualmente existen 10 países (Reino Unido, Canadá, Finlandia, Noruega, Australia, Irlanda, Italia, Emiratos Arabe, Holanda y España) que cuentan con expertos certificados en este método estándar. En cuanto al “*valor numérico*”, este valor es obtenido mediante la aplicación de la función de medición definida en el manual de COSMIC-FFP [24]. Finalmente, en cuanto al concepto “*escala de referencia convencional*” en RmFFP es dado por el conjunto de valores numéricos que son discretos, cuyo valor mínimo es 1 movimiento de datos y su valor máximo no es establecido. Actualmente RmFFP reconoce que esta escala es expresada en movimientos de “grupos de dato” que es un conjunto de atributos de datos. Aunque también esta escala podría ser expresada en movimientos de “atributos de datos”, sin embargo no es del todo adecuada ya que no se cuenta con una especificación detallada de todos los atributos del sistema.

Por lo tanto, retomando los conceptos de múltiplos y sub-múltiplos, a la pregunta si eran posible definirlos, actualmente no sería posible definirlos, ya que sería necesario conocer la relación de equivalencia entre las diversas formas de expresar un *dato*, tales como: grupos de datos, atributos de datos. Esta equivalencia es difícil de determinarla, ya que un grupo de datos puede incluir un número finito pero no limitado de atributos de datos.

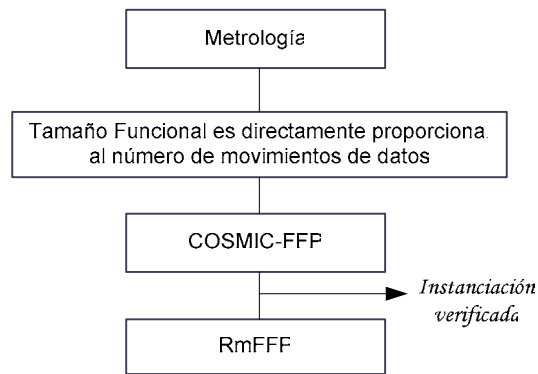
En la Tabla 10 se lista el análisis de RmFFP en base a los conceptos que pertenecen a la categoría “cantidades y unidades”. Este análisis ha sido ajustado a partir de [13].

**Tabla 10.** Análisis de “cantidades y unidades” en RmFFP

<b>Metrología</b>	<b>RmFFP</b>
Cantidad base	Tamaño funcional
Cantidad de dimensión	Unidimensional
Símbolo de unidad	Cfsu
Unidad base	Movimiento de dato
Sistema de unidades	Conformada únicamente por la unidad base
Valores	
Valor verdadero convencional	Obtenido por un experto
Valor numérico	Obtenido por la aplicación de la función de medición
Escala de referencia convencional	Conjunto de valores numéricos, valor mínimo: 1 movimiento de dato y valor máximo: no es establecido

### 6.3.2.2. *Fundamento de la medición*

El “principio de medición” representa la base científica de una medición. A partir de este principio de medición, un “método” es definido en un sentido general y luego es instanciado en un “procedimiento de medición”. En la Figura 33 se observa que RmFFP representa la instanciación del método estándar COSMIC-FFP cuyo principio de medición es claramente establecido [24]. De este modo, RmFFP forma parte de la jerarquía de conceptos. La relación de instanciación entre COSMIC-FFP y RmFFP ha sido verificada en la sección de evaluación de conformidad.



**Figura 33.** Aplicación del fundamento de la medición

### 6.3.2.3. Estándares de medición

En lo que respecta a la medición del tamaño funcional del software, la ISO/IEC 19761[22] es un método estándar que sería catalogado como un “estándar primario”, ya que los valores obtenidos con este método de medición son aceptados sin referenciar a otros estándares relacionados con la misma cantidad (tamaño funcional). El diseño de RmFFF se basó sobre este tipo de estándar. Un instrumento de evaluación de conformidad (ver Anexo C) fue elaborado con el fin de contribuir a la conservación de la trazabilidad entre la ISO/IEC 19761 y el procedimiento de medición RmFFF.

## 6.4. Validación teórica

Zuse [91], Kitchenham [95] y Fenton [96] afirman que una medida es válida teóricamente si se comprueba la condición de representación. Esto significa que si una estructura relacional empírica cumple con una serie de axiomas o condiciones, esta estructura puede ser utilizada para construir una representación numérica [89]. Analizando esta condición de representación en la fase de diseño del modelo de proceso descrito (vea Figura 28), se tiene que a través del paso: *caracterización del concepto a ser medido* se define una *estructura relacional empírica* y con el paso: *definición de reglas de asignación numérica* se

define una *estructura relacional numérica*. La correspondencia entre ambas estructuras (relación entre ambos pasos) debe cumplir con la condición de representación.

#### 6.4.1. Trabajos relacionados

Entre los trabajos que han sido realizados para intentar validar teóricamente los métodos de tamaño funcional tenemos a Zuse [91], Fetcke [92], Poels [68] y Abrahão [93].

##### 6.4.1.1. Propuesta de Zuse (1998)

Zuse [91] aplicó los teoremas de la estructura extensiva modificada<sup>2</sup> para llevar a cabo la validación de las medidas del método de tamaño funcional FPA en el nivel de escala tipo ratio.

Para poder ilustrar la estructura del método de FPA con la estructura extensiva, Zuse rescribe la fórmula de puntos de función de la siguiente manera:

$$\begin{array}{l}
 FP = CU * (0.65 + 0.01 * TCF) \\
 CU = \text{Conteo desajustado} \\
 TCF = \text{Factores de complejidad técnica}
 \end{array}
 \quad \Longrightarrow \quad
 \begin{array}{l}
 \text{Fórmula re-escrita} \\
 FP' = CU * k \\
 FP' = CU \\
 \text{Donde:} \\
 k = 0.65 + 0.001 * TCF
 \end{array}$$

Según Zuse, el tamaño funcional en base al método de medición FPA desde el punto de vista de la teoría de la medición se reduce a considerar sólo la medida de puntos de función desajustados (CU). Esto es debido a que el valor constante “k” no afecta al sistema relacional empírico, sólo al sistema relacional numérico.

---

<sup>2</sup> Un sistema relacional es una estructura extensiva modificada si cumple los axiomas de orden débil, positividad, asociatividad, conmutatividad débil, monotonicidad débil [89].

Considerando una medición extensiva, una medida asume una regla de combinación aditiva relacionada a cierta operación de concatenación “ $\circ$ ” tal como sigue:  $CU (P1 \circ P2) = CU (P1) + CU (P2)$ .

Donde:

CU representa los puntos de función sin ajustar.

P1 y P2 son los proyectos medidos por FPA.

$P1 \circ P2$  es la concatenación de los proyectos P1 y P2.

Para responder a la pregunta: ¿cuál sería la operación de concatenación empírica  $P1 \circ P2$ ?, Zuse asumió que los pesos de las funciones transaccionales y de datos para ambos proyectos son todos iguales. Si aplicamos  $CU (P1 \circ P2)$  como la suma de los tamaños para las funciones lógicas de los dos proyectos, en algunos casos esto puede ocurrir, pero en otros casos donde la misma función lógica puede aparecer en ambos proyectos, esto no sería posible. Este hecho le permitió sugerir la siguiente regla de combinación:

$$CU (P1 \circ P2) = CU (P1) + CU (P2) - CU (\text{intersección } (P1, P2))$$

Las reglas de combinación no aditivas pueden asumir una estructura extensiva y pueden por ende representar escalas de tipo ratio. La interrogante es verificar si esta regla de combinación presenta las propiedades de una estructura extensiva. Para esto, Zuse observó que la segunda parte de la regla ( $CU (\text{intersección } (P1, P2))$ ) viola la estructura extensiva, por ser idempotentes. Esto implica que el axioma de Arquímedes es violado, consecuentemente, la medida de puntos de función no ajustados (CU) no adopta una estructura extensiva.



#### 6.4.1.2. Propuesta de Fetcke (2001)

Fetcke propone una representación generalizada del modelo de abstracción del software [97]. La formalización de esta vista generalizada ha permitido un estudio de suposiciones consideradas por las variantes de FPA. Fetcke presenta dos condiciones adicionales al *axioma de orden débil*. Estos son el *axioma de dominancia* y el *axioma de monotonicidad*.

Fetcke afirma que el *axioma de dominancia* requiere que el *axioma de orden débil* adoptado por una medida de puntos de función concuerde con la instancia de una aplicación. Por ejemplo, si se tiene una instancia X de una aplicación X', el tamaño funcional de esta instancia debe ser más pequeño que el de la aplicación X'. Esto se ve reflejado en la siguiente fórmula:  $X \subseteq X' \Rightarrow FP(X) \leq FP(X')$ .

La demostración de los resultados, según Fetcke, es que IFPUG FPA viola este axioma mientras que Mark II FPA y FFP sí los cumple.

En cuanto al *axioma de monotonicidad*, dado el tamaño funcional de tres aplicaciones X, X', X'', si una aplicación X es más pequeña que X', dada una extensión de ambas aplicaciones con los mismos tipos de funciones X'', no puede darse que la aplicación extendida  $X \cup X''$  sea más grande que la aplicación  $X' \cup X''$ . Esto se ve reflejado en:  $FPC(X) \leq FPC(X') \Leftrightarrow FPC(X \cup X'') \leq FPC(X' \cup X'')$ .

Aplicando este axioma para los métodos de medición de tamaño funcional, Fetcke observó que IFPUG FPA y FFP 1.0 violan el axioma de monotonicidad. Sin embargo los métodos Mark II FPA y FFP 2.0 cumplen con el axioma de monotonicidad.

Los resultados del estudio de Fetcke permitieron la validación de estos métodos como una medida de tamaño funcional para el tipo de escala

ordinal. Estos axiomas no permiten la validación de las medidas en un tipo de escala más sofisticado.

#### **6.4.1.3. Propuesta de Abrahão (2004)**

Abrahão en [93] valida de manera parcial el diseño de un procedimiento de medición de tamaño funcional denominado OOmFP, el cual ha sido definido de acuerdo al método de medición IFPUG FPA. Para esto, Abrahão aplica un marco formal denominado DISTANCE [69]. OOmFP [53] incluye la medición de cuatro perspectivas: datos, proceso, comportamiento, y presentación que forman parte del método OO-Method [54]. Abrahão también valida la extensión de OOmFP para aplicaciones web, mediante la incorporación de una perspectiva navegacional que caracteriza al método de desarrollo OOWS [98].

A diferencia de las anteriores propuestas, Abrahão demostró que utilizando los conceptos de disimilaridad y distancia es posible validar las medidas de tamaño funcional para un tipo de escala ratio. Sin embargo, el tamaño funcional obtenido con OOmFP no resulta sólo de agregar directamente estas medidas, sino que necesitan ser clasificadas como simple, medio o complejo para así obtener posteriormente la cantidad de puntos de función correspondiente. Abrahão no consideró esta última parte para la validación de OOmFP.

#### **6.4.1.4. Propuesta de Poels (2003)**

Poels [68] aplicó el marco formal DISTANCE [69] basado en la teoría de la medición para validar el método de tamaño funcional COSMIC-FFP [45] para los sistemas orientado a objetos.

Esta propuesta a diferencia de los anteriores trabajos ofrece un procedimiento de la construcción de la medida para modelar propiedades de

artefactos de software y para definir las correspondientes métricas. Poels utiliza un tipo de estructura de proximidad llamada estructura de proximidad aditiva segmentada [99], la cual permite la definición de una medida con el tipo de escala ratio. Poels en primer lugar describió el tamaño funcional usando una estructura relacional que satisface los axiomas de la estructura de proximidad segmental aditiva. En segundo lugar, un espacio de métrica con segmentos aditivos [99] fue construido y usado para definir un tipo de escala ratio. Poels concluyó que la medida construida por medio del Modelo de Proceso de DISTANCE no es equivalente a la medida definida en el metamodelo del método de medición COSMIC-FFP.

#### 6.4.1.5. Discusión

Esta sección, discute cada uno de los trabajos revisados, los cuales son resumidos en la Tabla 11.

**Tabla 11.** Resumen de trabajos sobre validación teórica

<b>Características</b>	<b>Propuesta de Zuse</b>	<b>Propuesta de Fetcke</b>	<b>Propuesta de Abrahão</b>	<b>Propuesta de Poels</b>
¿En qué conceptos se basa?	Estructura extensiva modificada	Axiomas de Dominancia y Monotonicidad	Estructuras de Proximidad segmentada aditiva	Estructuras de Proximidad segmentada aditiva
¿Qué tipo de escala aborda?	Ratio	Ordinal	Ratio	Ratio
¿Qué se valida?	FPA	FPA, Mark II FPA, FFP	OOmFP	COSMIC-FFP
¿Validado?	No	No	No	Parcialmente

Como se observa en dicha tabla, hasta el momento no se ha conseguido validar teóricamente el método de medición FPA. Uno de los problemas radica en las diferentes escalas de medición que ocasionan diferentes transformaciones admisibles dentro del método FPA y que estas transformaciones de escala no son permitidas de acuerdo a la teoría de la

medición. Por tal razón, tanto Zuse como Abrahão sólo abordan la validación para el tipo de escala ratio del metamodelo de medición de FPA.

Abrahão validó el conjunto de medidas directas identificadas en el procedimiento de medición OOmFP, la cual está basada en IFPUG FPA. Para esto, Abrahão utilizó el marco formal DISTANCE [69] que está basado en la estructura de proximidad segmentada aditiva (Ver Apéndice A).

Así mismo, la estructura extensiva modificada, que utilizó Zuse, resultó ser insuficiente para la concatenación de tamaños funcionales de dos o más proyectos software, debido a que la operación de concatenación en una estructura relacional empírica no se corresponde con la operación aditiva definida para una estructura relacional numérica representada en una escala ratio.

En cuanto al trabajo de Fetcke al abordar la validación de las medidas bajo una escala de tipo ordinal, resultó ser más difícil llevar a cabo esta validación en base a la teoría de la medición, ya que el teorema de la unicidad no es establecido para este tipo de escala [89]. Por este motivo, Fetcke siguió una orientación basada en propiedades, definiendo los axiomas de dominancia y monotonicidad para los métodos variantes de FPA tales como Mark II FPA y FFP.

Por otro lado, Poels en base a la estructura de proximidad segmentada positivamente logró representar el tamaño funcional de COSMIC-FFP en un nivel de escala ratio. Sin embargo, la medida construida por medio del Modelo de Proceso de DISTANCE no fue equivalente a la medida definida en el metamodelo del método de medición COSMIC-FFP. El problema radicó en la incorrecta identificación de la abstracción de referencia, ya que el tamaño mínimo obtenido con COSMIC-FFP puede ser de 1 Cfsu y no de 2 Cfsu como lo indicó Poels.

En la siguiente sección, se introduce brevemente el proceso del marco formal DISTANCE [69], el cual será utilizado para validar teóricamente las medidas obtenidas con el procedimiento de medición RmFFP. Las razones de por qué se ha utilizado este marco son las siguientes:

- Proporciona un marco que utiliza el concepto de distancia y disimilaridad basado en estructuras de proximidad [99].
- Provee un marco sistemático a través de un modelo de proceso a seguir para la construcción de una medida.
- Soporta el tipo de escala ratio para la caracterización de los atributos a ser medidos, mejorando de esta manera la propuesta de Zuse [91].

#### 6.4.2. *Modelo de proceso DISTANCE*

El marco DISTANCE [69] proporciona procedimientos constructivos para modelar los atributos de interés del software y definir las correspondientes medidas. Los atributos del software son modelados como distancias conceptuales entre las entidades del software que son caracterizadas y otras entidades del software que sirven como puntos de referencia. Estas distancias son medidas mediante funciones matemáticas que satisfacen el conjunto de axiomas del espacio de métricas.

El proceso de construcción de medidas basadas en la distancia consiste de cinco pasos, los cuales deben ejecutarse en el orden establecido. Dicho proceso se inicia con la solicitud de una petición para encontrar o construir una función de medición para un atributo de software *attr* que caracteriza las entidades de software en un conjunto *P*. Para esto, primero se debe encontrar una abstracción de la medición para las entidades de software. Las entidades de software de interés deben ser modeladas de manera tal que enfatizen el atributo en cuestión. Esto significa que el modelo deberá permitir observar hasta qué punto una entidad de software es caracterizada por el

atributo. El resultado del primer paso es un conjunto de entidades de software  $M$  que puede ser usado como abstracciones de medición o modelos de las entidades de software del conjunto  $P$  para el atributo de interés  $attr$ . Además se debe definir una función de abstracción  $abs: P \rightarrow M$ .

Si el primer paso tiene éxito, podrán ejecutarse en paralelo dos caminos de actividades. Por un lado, definir el conjunto  $M$  como un espacio de métrica. Para esto, se modela las distancias entre los elementos de  $M$  como secuencias de transformaciones elementales ( $Te$ ). Tales secuencias representan una serie de cambios atómicos aplicados a un elemento de  $M$  para llegar a otro elemento de  $M$ . El número de cambios atómicos necesarios para transformar un elemento en otro determina la distancia entre estos elementos. La salida formal de este paso es un conjunto de tipos de transformaciones elementales sobre  $M$ , que deben usarse para construir las secuencias. A continuación se define una métrica:  $M \times M \rightarrow R$  para cuantificar la distancia entre los elementos de  $M$ .

Por otro lado, se necesita determinar como se ve el modelo de la entidad de software en  $P$ , en el caso de que la entidad fuera caracterizada por el menor valor teórico de  $attr$ . Entonces, este hipotético modelo “nulo” o modelo de referencia puede ser utilizado como punto de referencia o norma para la medición. El resultado de este paso es la definición de una función  $ref: P \rightarrow M$  que devuelve para cada entidad de software en  $P$  una abstracción de referencia para  $attr$  en  $M$ . Después de haber ejecutado los dos caminos de actividades, existe un paso final en el proceso de medición basado en la distancia. Este último paso es el que expresa la idea básica de esta aproximación. El atributo de software  $attr$  se define y se mide como una distancia específica dentro del espacio de métrica  $M$ . El punto hasta el cual  $attr$  caracteriza una entidad de software  $p \in P$  se define mediante la distancia entre

el modelo actual de  $p$  para el atributo  $attr$  (i.e.,  $abs(p)$ ) y el modelo de referencia para  $attr$  (i.e.,  $ref(p)$ ). Cuanto mayor sea la distancia, más difiere la abstracción de la medición real del modelo de referencia que se ha establecido y así es mayor el grado con que  $attr$  caracteriza  $p$ . Por eso, el valor de  $attr$  para  $p$  es el valor obtenido a través de la métrica  $\delta$  para el par  $(abs(p), ref(p))$ . Finalmente, la salida formal del último paso es la medida  $\mu: P \rightarrow R$  definida tal que  $\forall p \in P: \mu(p) = \delta(abs(p), ref(p))$

### 6.4.3. Validando teóricamente RmFFP

En esta sección, se presenta la validación teórica del procedimiento de medición de tamaño funcional RmFFP. Este tipo de validación es demostrada a través de la correspondencia existente entre los pasos del modelo de proceso de medición “*caracterización del concepto a ser medido*” y “*definición de las reglas de asignación numérica*”. Para validar esta correspondencia, tal como se observa en la Figura 34, utilizamos el modelo de proceso del marco formal DISTANCE [7]. Los tres primeros pasos de DISTANCE permiten la comprensión empírica del atributo de interés a ser medido por medio de la definición de una *estructura de proximidad aditiva segmentada*. El penúltimo paso, cuantificar las distancias entre las abstracciones de la medición, permite describir el sistema relacional numérico mediante la definición de un *espacio de métrica con segmentos aditivos*. Finalmente, el último paso del proceso de DISTANCE permite establecer una correspondencia entre el sistema relacional empírico y el sistema relacional numérico. Consecuentemente una medida es construida válidamente si todos los pasos del modelo de proceso de DISTANCE son llevados a cabo exitosamente.

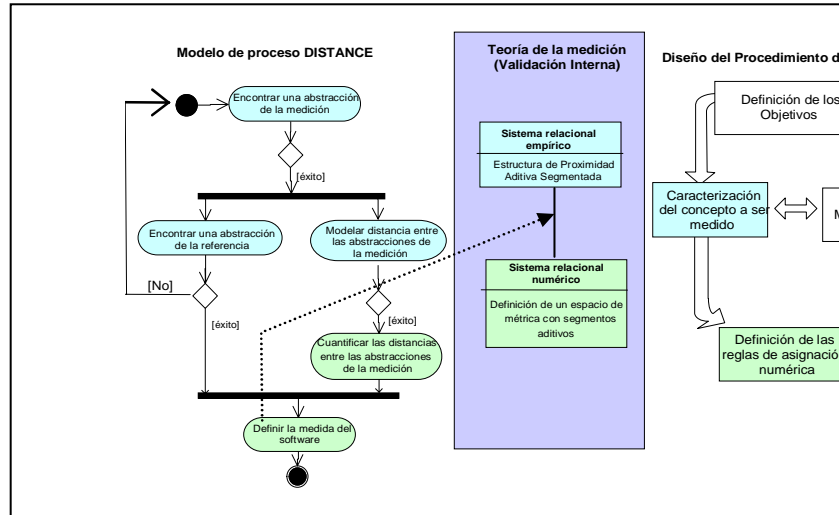


Figura 34. Validación teórica de RmFFP usando DISTANCE

En el procedimiento de medición RmFFP, el atributo a ser medido es el tamaño funcional, el cual es expresado por la cantidad de movimientos de datos. En la Tabla 12, se presenta el conjunto de medidas identificadas para las diferentes primitivas del modelo de requisitos de OO-Method que contribuyen al tamaño funcional.

Tabla 12. Medidas del procedimiento de medición RmFFP

Entidad	Atributo	Medidas
Diagrama de Secuencia	Cantidad de movimientos de datos tipo entrada	Número de movimientos de datos tipo entrada identificado por cada diferente grupo de dato involucrado en el mensaje signal con el valor input (NMSI).
	Cantidad de movimientos de datos tipo salida	Número de movimientos de datos tipo salida identificados por cada diferente grupo de dato involucrado en el mensaje signal con valor output (NMSO).



Entidad	Atributo	Medidas
	Cantidad de movimientos de datos tipo lectura	Número de movimientos de datos tipo lectura identificados por cada diferente grupo de dato involucrado en el mensaje query (NMQ)
		Número de movimientos de datos tipo lectura identificados por cada diferente grupo de dato involucrado en la condición de mensaje (NCM)
	Cantidad de movimientos de datos tipo escritura	Número de movimientos de datos tipo escritura identificados por cada diferente grupo de dato involucrado en el mensaje service/new (NMSN)
		Número de movimientos de datos tipo escritura identificados por cada diferente grupo de dato involucrado en el mensaje service/destroy (NMSD)
		Número de movimientos de datos tipo escritura identificados por cada diferente grupo de dato involucrado en el mensaje service/update (NMSU)
	Modelo de Casos de Uso	Cantidad de movimientos de datos tipo lectura
Número de movimientos de datos tipo lectura identificados por cada diferente grupo de dato involucrado en la condición de relación de extend (NCRE)		

A continuación, la aplicación del modelo de proceso de DISTANCE es ilustrado para validar la medida NMSI.

#### 6.4.3.1. Encontrar una abstracción de medición

La primera condición según DISTANCE es que una pieza de software sea modelada de tal forma que el atributo sea enfatizado. Este modelo es denominado abstracción de medición, y contiene sólo las entidades que contribuyen para determinar el tamaño funcional de dicha pieza. De

acuerdo al metamodelo de COSMIC-FFP, los componentes funcionales base de una pieza de software son los movimientos de datos que se llevan a cabo en los respectivos procesos funcionales. Por lo tanto, una abstracción de medición adecuada para el Modelo de Requisitos OO-Method es el conjunto de movimientos de datos identificados en las primitivas relevantes de dicho modelo. A continuación describimos la notación utilizada en la definición de la función de abstracción de medición encontrada para el atributo “Cantidad de movimientos de datos tipo entrada”.

- Sea UMROO, el Universo de Modelos de Requisitos OO-Method.
- Sea UMDE el Universo de Movimientos de Datos tipo Entrada.
- La notación “ $\wp(UMDE)$ ” es denotado como un conjunto de conjuntos de movimientos de datos tipo entrada UMDE.
- La operación de proyección de MDE retorna para un MROO que pertenece a UMROO el conjunto de movimientos de datos tipo entrada, el cual es un elemento de  $\wp(UMDE)$

La abstracción de medición es definida por:

$$abs_{NMSI} : UMROO \rightarrow \wp(UMDE) : MROO \rightarrow MDE(MROO)$$

#### 6.4.3.2. Modelar distancias entre abstracciones de medición

El objetivo de este paso es modelar distancias entre los elementos del conjunto de  $\wp(UMDE)$ . La condición en este paso es que un conjunto de funciones homogéneas sobre  $\wp(UMDE)$  puede ser definido aplicando repetidamente estas funciones para que algún conjunto de movimientos de datos tipo entrada pueda ser transformado dentro de otro conjunto de movimiento de datos tipo entrada. Como  $\wp(UMDE)$  es un conjunto de conjuntos, solamente dos funciones homogéneas son requeridas: una función

que añade un movimiento de dato tipo entrada a un conjunto y otra función que disminuye un movimiento de dato tipo entrada de un conjunto. En base a estas dos funciones homogéneas una estructura de proximidad aditiva segmentada puede ser construida para definir una noción de distancia entre conjuntos de movimientos de datos tipo entrada.

Sea  $\wp(UMDE)$  un conjunto y  $T_e$  un conjunto de funciones homogéneas sobre  $\wp(UMDE)$ .

$T_{e-NMSI} = \{t_{0-NMSI}, t_{1-NMSI}\}$  donde  $t_{0-NMSI}$  y  $t_{1-NMSI}$  son definidas:

$$t_{0-COSMIC-FFP} = \wp(UMDE) \rightarrow \wp(UMDE) : s \rightarrow s \cup \{e\} \text{ con } e \in UMDE$$

$$t_{1-COSMIC-FFP} = \wp(UMDE) \rightarrow \wp(UMDE) : s \rightarrow s - \{e\} \text{ con } e \in UMDE$$

“s” es un conjunto que pertenece a  $\wp(UMDE)$  y que puede ser transformado en otro por medio de las funciones homogéneas que pertenecen a  $T_e$ .

Entonces se tiene que la estructura  $(\wp(UMDE), \bullet \geq_{MDE})$  es una estructura de proximidad aditiva segmentada puesto que cumple con los axiomas definidos en el Apéndice A.

#### 6.4.3.3. Encontrar una abstracción de referencia

La condición de este paso es identificar la abstracción de medición para el modelo de requisitos OO-Method con el tamaño funcional teóricamente más bajo. Como la abstracción de medición es dado por el conjunto de movimiento de datos tipo entrada, el valor teórico más bajo sería el conjunto sin ningún movimiento de dato tipo entrada. Por lo tanto, la abstracción de referencia es dado por:

$$ref_{NMSI} : UMROO \rightarrow \wp(UMDE) : MROO \rightarrow \phi$$

Teniendo en cuenta el metamodelo de COSMIC-FFP, el proceso funcional más pequeño es constituido por dos movimientos de datos, los cuales pueden ser una entrada y una salida o escritura. Sin embargo, se ha considerado el conjunto vacío y no el conjunto formado por dos movimientos de datos ya que se ha tomado en cuenta el *valor teórico* más bajo.

#### **6.4.3.4. Cuantificar las distancias entre abstracciones de medición**

De acuerdo a la teoría de la medición, la noción de distancia definida por una estructura de proximidad aditiva segmentada puede ser representada por una métrica con segmentos aditivos. Es decir, el mínimo número de movimientos de datos tipo entrada que deben ser añadidos o removidos de un conjunto para transformarlo en otro conjunto, se califica como una métrica con segmentos aditivos para la estructura de proximidad definida sobre  $\wp(UMDE)$ . La longitud de la secuencia mas corta de transformaciones elementales tomado por ejemplo desde  $s$  a  $s'$  es igual a la cardinalidad de la diferencia simétrica entre  $s$  y  $s'$ . De esta manera la métrica  $\delta_{NMSI}$ , que permite la cuantificación de distancias entre elementos de  $\wp(UMDE)$ , es definida de la siguiente manera:

$$\delta_{NMSI} : \wp(UMDE) \times \wp(UMDE) \rightarrow R : (s, s') \rightarrow |s - s'| + |s' - s|$$

#### **6.4.3.5. Definir la medida de software**

Este último paso de acuerdo al marco de DISTANCE, la cantidad de movimientos de datos tipo entrada del modelo de requisitos OO-Method:  $MROO \in UMROO$  puede ser medido como sigue:

$$\begin{aligned}\mu_{NMSI} &= \delta(abs_{NMSI}(MROO), ref_{NMSI}(MROO)) \\ \mu_{NMSI} &= |abs_{NMSI}(MROO) - ref_{NMSI}(MROO)| + |ref_{NMSI}(MROO) - abs_{NMSI}(MROO)| \\ \mu_{NMSI} &= |MDE(MROO) - \phi| + |\phi - MDE(MROO)| \\ \mu_{NMSI} &= MDE(MROO)\end{aligned}$$

Consecuentemente la medida NMSI es correctamente definida como el número de movimientos de datos tipo entrada identificados en cada mensaje signal con el valor input.

Aplicando los mismos pasos, las demás medidas de la Tabla 12 han sido validadas en [128].

## 6.5. Conclusiones

En este capítulo se ha validado el diseño de RmFFP en base a tres aspectos: evaluación de su conformidad con la ISO/IEC 19761, análisis de la calidad del diseño de RmFFP bajo un punto de vista de la metrología, y su validación teórica usando el marco formal DISTANCE.

La evaluación de conformidad de RmFFP fue verificada por un grupo de profesionales certificados en COSMIC-FFP de la empresa El Corte Inglés, quienes tienen experiencia en lo que respecta a medición del software. Para esta evaluación, se adaptó la segunda parte del estándar ISO/IEC 14143, diseñando un instrumento como soporte para dicha evaluación. Los resultados obtenidos permitieron concluir que RmFFP es un procedimiento de medición de tamaño funcional conforme con el método estándar COSMIC-FFP (ISO/IEC 19761).

Así mismo un análisis metrológico fue también llevado sobre el concepto a medir por RmFFP tal como el tamaño funcional, y sobre las reglas de medición definidas. Este análisis permitió corroborar que RmFFP si

considera la mayoría de los conceptos relacionados con la medición del tamaño del software, a excepción de los conceptos: “múltiplos y sub-múltiplos”, los cuales aún no han sido estudiadas. Finalmente se validó también teóricamente el procedimiento RmFFP usando el marco formal DISTANCE, el cual es llevado a cabo sobre la caracterización del concepto a ser medido y la definición de reglas de asignación numérica.



*“No es difícil tomar decisiones cuando conoces  
cuales son los valores”.*

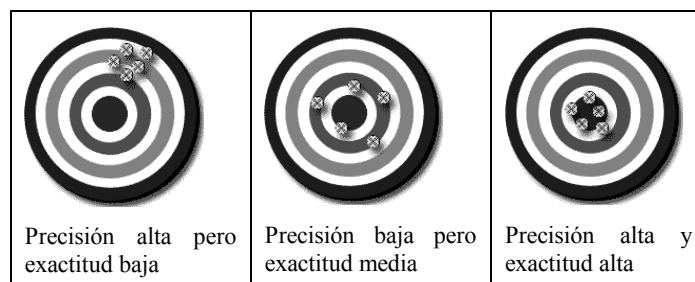
*- Roy Disney*

### ***7.1. Introducción***

La calidad del diseño de un procedimiento de medición no es suficiente para asegurar la calidad de los resultados [100]. Es necesario demostrar con evidencia real que este procedimiento de medición es útil, en el sentido de que provee resultados fiables.

El VIM [1] y el estándar ISO 5725 [28] definen a la exactitud como la proximidad entre un resultado de prueba y el valor de referencia aceptado. Asimismo, la precisión es definida como la proximidad entre los resultados de pruebas independientes obtenidos bajo ciertas condiciones estipuladas de reproducibilidad y repetibilidad [28]. La Figura 35 ilustra claramente la diferencia entre estos dos conceptos de exactitud y precisión, los cuales generalmente han sido referidos de una manera indistinta.

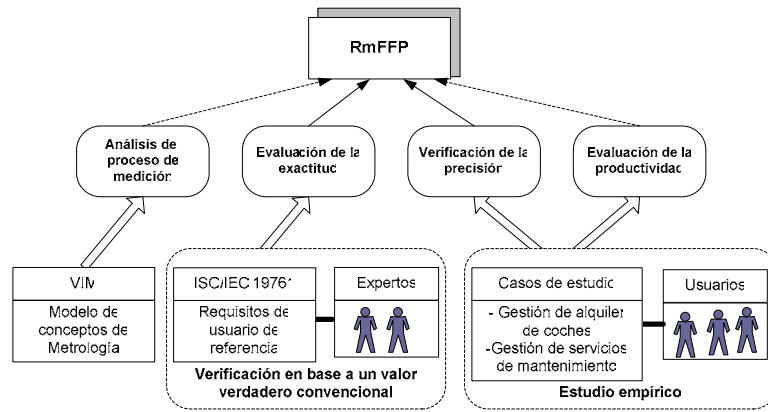




**Figura 35.** Analogía entre exactitud y precisión

En cuanto a la reproducibilidad y repetibilidad, según el estándar ISO 5725 [28] y el informe técnico ISO/IEC 14143-3 [18], nos referimos a condiciones de reproducibilidad, si los resultados son obtenidos usando el mismo objeto a medir bajo diferentes condiciones de medición (el observador, condiciones de uso, el tiempo, el instrumento de medición, etc.). En cambio, nos referimos a condiciones de repetibilidad, si los resultados son obtenidos usando el mismo objeto a medir bajo las mismas condiciones de medición [18].

El presente capítulo tiene como fin evaluar la fiabilidad de los resultados de medición obtenidos con RmFFP en términos de su exactitud y precisión, la cual es verificada sólo bajo condiciones de reproducibilidad. Un análisis previo del proceso de medición en base a los términos de metrología es también llevado a cabo, así como una evaluación de la productividad de los usuarios al usar RmFFP. La Figura 36 muestra estas actividades y los documentos de soporte (estándares, casos de estudio, modelos) necesarios para llevar a cabo la evaluación de la aplicación de RmFFP.



**Figura 36.** Alcance de la evaluación de la aplicación de RmFFP

A continuación se explica el análisis llevado a cabo sobre el proceso de medición de RmFFP.

## 7.2. Análisis metrológico de la aplicación de RmFFP

Teniendo en cuenta la relación establecida entre las categorías del VIM y el Modelo de Proceso de Medición descrito en el capítulo 6, la aplicación de un procedimiento de medición está relacionada con las siguientes categorías: *proceso de medición*, *instrumentos de medición*, *características de los instrumentos de medición* y *estándares de medición*.

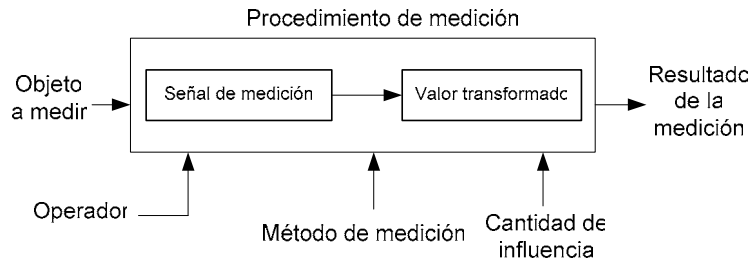
Un análisis con respecto a la categoría *instrumentos de medición* y sus *características* no ha sido llevado a cabo, ya que la automatización de RmFFP no está dentro del alcance del presente trabajo de investigación. No obstante, este análisis se plantea como trabajo futuro.

A continuación se presenta el análisis metrológico con respecto al proceso de medición.

### 7.2.1. Proceso de medición

Según el modelo de metrología [11], tal como se observa en la Figura 37, los conceptos involucrados en un proceso de medición son el *objeto a*

*medir, operador, procedimiento de medición, método de medición, señal de medición, valor transformado, cantidad de influencia en la medición y resultados de la medición.*



**Figura 37.** Sub-conceptos de la categoría “Medición” [11]

Para verificar si cada uno de estos conceptos ha sido considerado en el proceso de medición de RmFFP, un análisis de correspondencia ha sido llevado a cabo, el cual es resumido en la Tabla 13.

El concepto *operador* es comprendido por los usuarios que usan RmFFP para llevar a cabo una medición determinada. Este uso puede ser de *manera manual*, donde el usuario utiliza una guía de medición (ver Anexo B); o puede ser de *manera automatizada*, donde el usuario utiliza la herramienta RETO que sería enriquecida con un módulo que automatizará todo el proceso de RmFFP, a excepción de la definición del contexto de medición. Para ambas formas de uso es necesario que los usuarios estén familiarizados con el Modelo de Requisitos de OO-Method [77].

En cuanto al concepto *objeto a medir*, RmFFP toma a la especificación de requisitos funcionales de alto nivel como el objeto bajo medición. Estas especificaciones son especificadas de acuerdo al proceso de ingeniería de requisitos de OO-Method [81].

En cuanto al *método de medición*, COSMIC-FFP versión 2.2 [24] fue el método estándar seleccionado para la definición del procedimiento RmFFP.

En cuanto a la *cantidad de influencias* que afecta al proceso de medición de RmFFP, se consideró a la calidad de la especificación de los requisitos funcionales, ya que algunos atributos de calidad de la especificación (trazabilidad, consistencia, claridad, etc.) afectan positivamente a la medición y otros atributos que no son controladas por la herramienta RETO, tal como la completitud funcional, afectan negativamente.

En cuanto a la *señal de medición*, este concepto es definido por el VIM como la cantidad que representa al objeto a ser medido [1]. Esta señal de medición en el procedimiento de RmFFP corresponde al modelo de software que es obtenido al aplicar las reglas de representación. Este modelo es el resultado de abstraer las primitivas relevantes del objeto a medir (especificación de requisitos) que contribuyen con el tamaño funcional.

En cuanto al *valor transformado* -definido como el valor de la señal de medición- está constituido por el conjunto de movimientos de datos identificados al construir el modelo del software. El valor transformado está relacionado con el objeto a medir, mientras que el *resultado de la medición* está relacionado con el atributo a medir. En RmFFP este resultado de medición es expresado cuantitativamente en unidades Cfsu.

Finalmente, tal como se observa en la Tabla 13, el proceso de medición de RmFFP basado en el método estándar COSMIC-FFP sí considera todos los conceptos de metrología relacionados con la categoría de “medición”.

**Tabla 13.** Correspondencia entre conceptos de metrología y el proceso de RmFFP

<b>Metrología</b>	<b>Proceso de RmFFP</b>
Objeto a medir	RmFFP mide especificaciones de requisitos funcionales, las cuales son especificadas con el método de desarrollo OO-Method

<b>Metrología</b>	<b>Proceso de RmFFP</b>
Operador	Los usuarios que utilizan RmFFP, en particular los ingenieros de requisitos y analistas.
Señal de medición	El modelo de software obtenido al aplicar las reglas de representación (mapping).
Valor transformado	Movimientos de datos identificados
Método de medición	COSMIC-FFP
Cantidad de influencia	La cantidad que influye en la medición de RmFFP es básicamente la calidad de la especificación funcional
Resultado de medición	Valor numérico que expresa el tamaño funcional en Cfsu.

### 7.2.2. Estándares de medición

En cuanto a esta categoría, se tiene como material de referencia a una guía de medición (Anexo B) que ha sido elaborada para aplicar RmFFP. Se proporciona además de un caso de estudio en el que se ilustra su aplicación.

## 7.3. Evaluación de la exactitud de RmFFP

La definición de la exactitud de un método de medición de tamaño funcional es claramente presentada en la ISO/IEC 14143-3 [18]. Sin embargo, actualmente no se ha encontrado ningún estándar que permita su verificación. En la presente sección se describe en primer lugar el contexto en el que se ha llevado a cabo la verificación, la estrategia a seguir, así como la aplicación inicial de dicha estrategia en un caso de estudio.

### 7.3.1. Contexto y estrategia

Teniendo en cuenta los estándares ISO 5725 [28], ISO/IEC 14143-3 [18] y el vocabulario de metrología [1], un *valor o resultado de referencia* es necesario para poder evaluar la exactitud de una medida. Este valor de referencia, según el Instituto Nacional de Investigación y Pruebas de Suecia [101], puede ser obtenido de tres formas:

- Valor establecido en base a principios científicos

- Valor certificado en base a un trabajo experimental por alguna organización nacional o internacional.
- Valor certificado en base a un trabajo experimental colaborativo bajo el auspicio de un grupo de científicos o ingenieros.

Además, según la ISO/IEC TR 14143-4 [19] un valor de referencia es establecido al aplicar un *método de medición de referencia* en determinados *Requisitos de Usuario de Referencia* (RUR). Los RUR son definidos por el mismo informe técnico [19] como un conjunto estándar de Requisitos de Usuario que deben ser documentados de tal forma que puedan ser entendidos por un usuario especializado y deben estar libres de ambigüedades e inconsistencias. Sin embargo, a pesar de que este informe técnico incluye un conjunto de requisitos de usuario tanto para el dominio de sistemas de información de gestión como de tiempo real, tales requisitos aún están siendo revisados para que puedan ser considerados como RUR.

Para la evaluación de la exactitud de RmFFP, el valor de referencia considerado ha sido el valor obtenido por un grupo de expertos del consorcio COSMIC [94], quienes utilizaron a COSMIC-FFP [22] como método de referencia y como RUR a las especificaciones de dos casos estudio extraídos de la ISO/IEC TR 14143-4, “*Sistema de Control de Válvulas*” y “*Sistema Automático de Reemplazo de Líneas*”. COSMIC-FFP es considerado como un *método de referencia*, debido a que

- Es conforme a la ISO/IEC 14143-1 de acuerdo a la ISO/IEC 14143-2;
- Es aplicable para el mismo dominio funcional que cubre RmFFP;
- Está disponible públicamente y su mínima efectividad para propósitos de evaluación ha sido verificada.

Por lo tanto, la estrategia a seguir es ilustrada en la Figura 38, en la cual básicamente podemos diferenciar las siguientes actividades:

- Medición llevada a cabo por un grupo de expertos usando el método de referencia COSMIC-FFP.
- Medición llevada a cabo por el mismo autor del procedimiento de medición RmFFP. Para esto, previamente se especifica el caso de estudio seleccionado usando el Modelo de Requisitos de OO-Method.
- Evaluación de los resultados obtenidos usando RmFFP en base a los resultados de referencia.

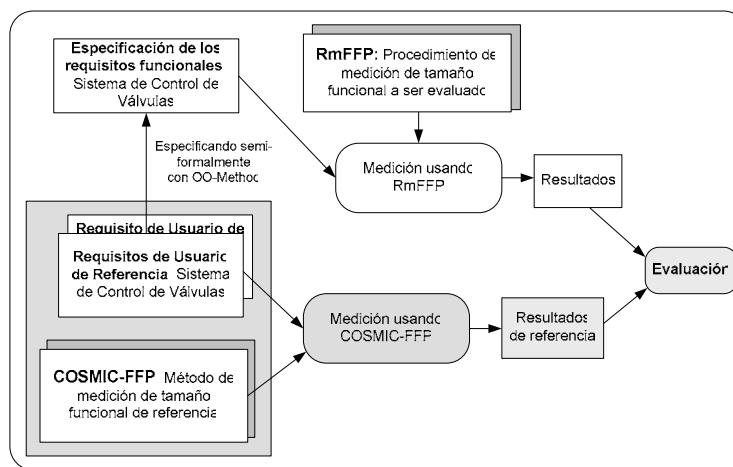


Figura 38. Estrategia para evaluar la exactitud, adaptado de [19]

### 7.3.2. Medición con COSMIC-FFP

Tal como se observa en la Figura 38, la medición con COSMIC-FFP ha sido llevada a cabo sobre un *Sistema de Control de Válvulas (SCV)*, cuya funcionalidad fue especificada textualmente en lenguaje natural y revisada por expertos del dominio de sistemas de tiempo real. La documentación relacionada con su especificación funcional y los resultados de la medición son

presentadas en [104]. La tabla 12 muestra los movimientos de datos identificados y los resultados obtenidos para todo el sistema SCV.

**Tabla 14.** Resultados del SCV usando COSMIC-FFP [104].

Identificación de movimientos de datos	Grupo de datos	Tipo de movimiento	Cfsu
Recibe señal de ciclo de reloj	Reloj	Entrada	1
Recibe señal de sensor Gc	Gc	Entrada	1
Recibe señal del sensor Su	Su	Entrada	1
Recibe señal del sensor Idl	Idl	Entrada	1
Lectura de parámetros ajustados de la válvula	Parámetros	Lectura	1
Lectura del tiempo T desde la RAM	T	Lectura	1
Lectura de ET desde la RAM	ET	Lectura	1
Lectura de Erev desde la RAM	Erev	Lectura	1
Lectura de PSrev desde la RAM	PSrev	Lectura	1
Lectura de Vs desde la RAM	Vs	Lectura	1
Envío de T a la válvula de control	T	Salida	1
Escritura de T en la RAM	T	Escritura	1
Tamaño funcional total en Cfsu			12

Este caso de estudio, tal como se observa en la Figura 38, fue también medido con el procedimiento de medición RmFFP, el cual es descrito a continuación.

### 7.3.3. Medición con RmFFP

Los requisitos de usuario de referencia del Sistema de Control de Válvulas son previamente especificados de manera semi-formal usando OO-Method, para luego ser medidos usando el procedimiento de medición



RmFFP, mediante la aplicación de reglas de representación y de asignación numérica. Cada uno de estos pasos es explicado a continuación.

### 7.3.3.1. Especificación funcional del sistema SCV

La documentación de los Requisitos Funcionales del caso de estudio SCV fue llevado a cabo en base al Modelo de Requisitos de OO-Method. Para esto, en primer lugar la misión del Sistema de Control de Válvulas fue identificado como “controlar el tiempo de cierre de una válvula, el cual está abierto por defecto y se cierra para embragar el mecanismo de cambio de engranaje”.

A partir de esta misión del sistema, se identificó una sola función elemental y por ende un único caso de uso: “Controlar el tiempo de cierre de la válvula”. En cuanto a los actores identificados, de acuerdo a la especificación presentada en [19], se tiene a tres tipos de sensores que proveen señales al SCV y la válvula que recibe el tiempo calculado por dicho SCV. La Figura 39 muestra el diagrama de casos de uso, donde se muestra la interacción de los actores con el sistema SCV.

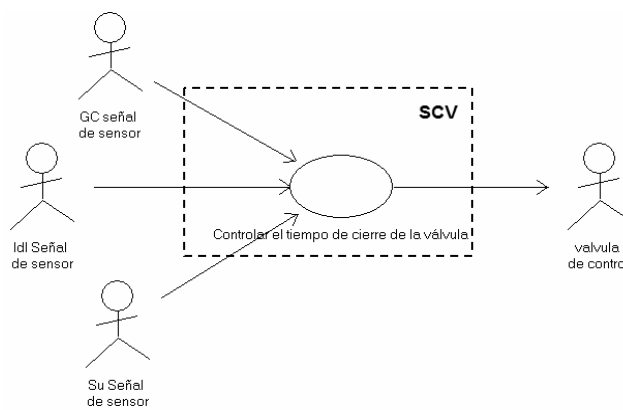


Figura 39. Diagrama de casos de uso de SCV

La interacción interna de este caso de uso es representada en dos diagramas de secuencia que corresponden a la Figura 40 y Figura 41 respectivamente.

### ***7.3.3.2. Aplicación de las reglas de representación***

Para construir el Modelo de Software que será posteriormente cuantificado, se aplicaron un conjunto de reglas que fueron descritas en el capítulo 4 a lo largo de los siguientes pasos:

- *Identificación del contexto de la medición:* El propósito es medir el tamaño funcional del caso de estudio SCV para verificar posteriormente su exactitud. El alcance de la medición comprende de los requisitos funcionales de usuario especificados con el Modelo de Requisitos de OO-Method. El punto de vista de medición considerado fue el del analista.

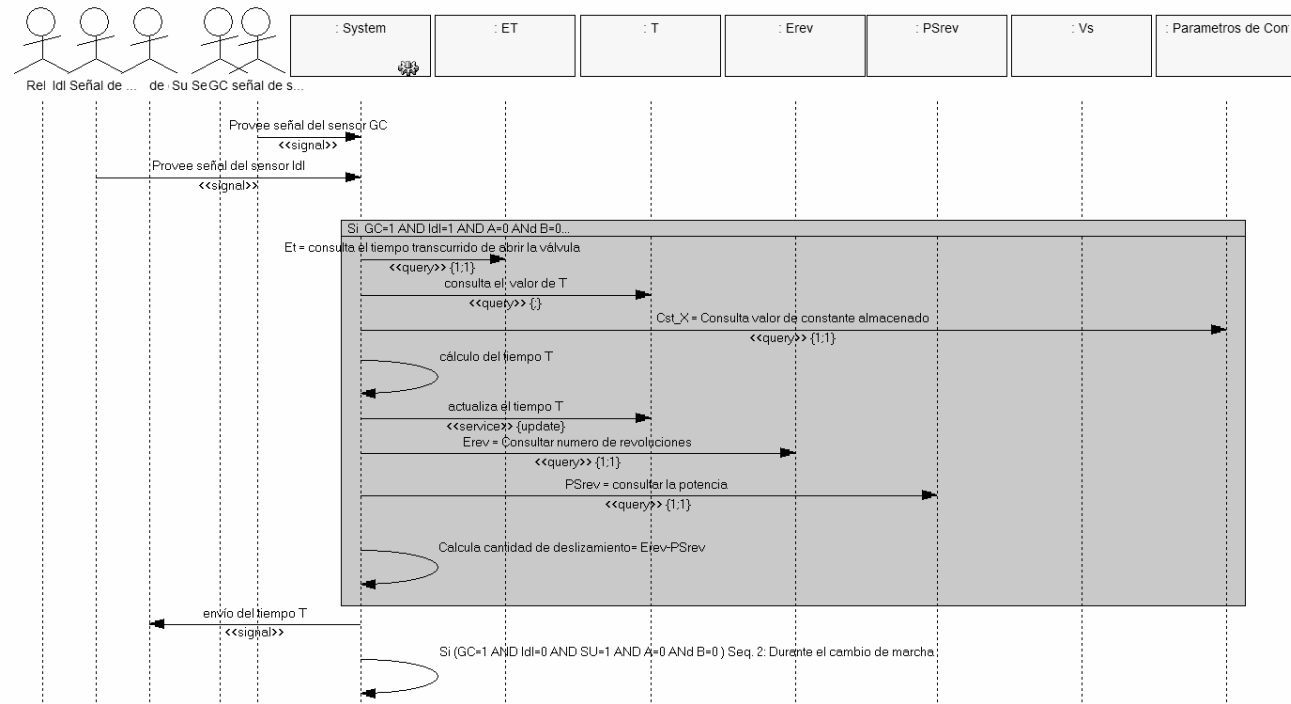


Figura 40. Diagrama de secuencia: Control de tiempo bajo condiciones normales

Un Procedimiento de Medición de Tamaño Funcional

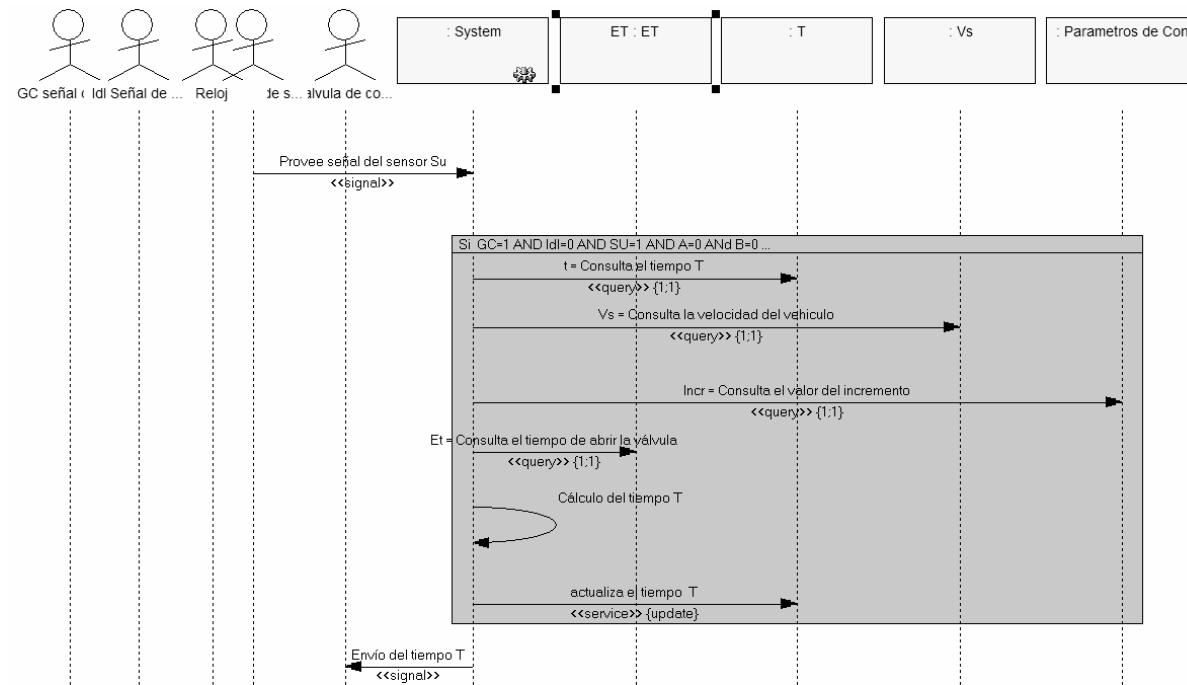


Figura 41. Diagrama de secuencia: Control de tiempo en cambio de marcha

- *Identificación de la frontera del software:* Aplicando la Regla 1 identificamos como usuarios del sistema a los actores del diagrama de casos de uso (sensor Gc, sensor Su, sensor Idl y la válvula de control). Según la Regla 2, la frontera es comprendida por el borde entre los usuarios y el sistema SCV. Por consiguiente el diagrama de casos de uso que se presenta en la Figura 39 nos permite identificar la frontera del software a medir.
- *Identificación de procesos funcionales:* Aplicando la Regla 4, el caso de uso primario “Controlar el tiempo de cierre de válvula” fue identificado como único proceso funcional.
- *Identificación de grupos de datos:* Aplicando la Regla 5, los actores ilustrados en la Figura 39 son identificados como grupos de datos. Además, aplicando la Regla 6, las diferentes clases de los dos diagramas de secuencia (Figura 40 y Figura 41) son también identificados como grupos de datos.
- *Identificación de los movimientos de datos:* Los diferentes tipos de movimientos de datos identificados en el proceso funcional “Controlar el tiempo de cierre de válvula” son resumidos en la Tabla 15.

**Tabla 15.** Movimientos de datos identificados con RMFFP

Nro	Mensajes del Diagrama de secuencia	Movimientos de datos
1	Provee señal del sensor GC	Entrada (Regla 8)
2	Provee señal del sensor Idl	Entrada (Regla 8)
3	Condición para ejecutar un conjunto de mensajes (Si GC=1, Idl=1, A=0, B=0)	Lectura (Regla 10)
4	Consulta del tiempo transcurrido	Lectura (Regla 9)
5	Consulta el último valor de T durante el último ciclo	Lectura (Regla 9)
6	Consulta el valor constante Cst_X	Lectura (Regla 9)
7	Actualiza el nuevo tiempo T	Escritura (Regla 14)

Nro	Mensajes del Diagrama de secuencia	Movimientos de datos
8	Consulta el número de revoluciones	Lectura (Regla 9)
9	Consulta el valor de la potencia PSrev	Lectura (Regla 9)
10	Provee señal del sensor Su	Entrada (Regla 8)
11	Condición para ejecutar un conjunto de mensajes (Si GC=1, Idl=0, Su=1, A=0, B=0)	Lectura (Regla 10)
12	Consulta el último valor de T durante el último ciclo	Lectura (Regla 9)
13	Consulta la velocidad del vehículo	Lectura (Regla 9)
14	Consulta el valor INCR según la velocidad	Lectura (Regla 9)
15	Consulta del tiempo transcurrido	Lectura (Regla 9)
16	Actualiza el nuevo tiempo	Escritura (Regla 14)
17	Muestra el nuevo tiempo T	Salida (Regla 16)

- *Eliminación de movimientos de datos duplicados:* Aplicando la Regla 17, los mensajes 12, 14, 15 y 16 fueron identificados como movimientos de datos duplicados porque involucran los mismos grupos de datos que fueron considerados en los mensajes con el mismo estereotipo 5, 6, 4 y 7 respectivamente.

#### 7.3.3.3. Aplicación de las reglas de asignación numérica

- *Aplicando la función de medición:* Se asigna el valor de 1 Cfsu a cada uno de los movimientos de datos.
- *Aplicando la función de agregación:* Mediante la Regla 21, obtuvimos un tamaño funcional de 13 Cfsu para el único proceso funcional identificado y por ende el tamaño del sistema SCV.

#### 7.3.4. Evaluación de la exactitud de RmFFP

La exactitud de la medición es obtenida mediante la proximidad existente entre el valor de referencia y el valor de prueba obtenido. Para llevar a cabo una comparación entre ambos resultados es necesario determinar un valor de incertidumbre, ya que toda medición está siempre sujeta a error. De

este modo, un resultado de la medición debiera ser presentado en la siguiente forma:  $Y = y \pm U$

Donde:

“U” representa el valor de la incertidumbre

“y” representa el valor de la medición obtenida

Actualmente no existen estudios sobre la determinación de la incertidumbre para la medición del software, tal como si existen en otras áreas, siendo el método GUM, descrito en la “Guía para la Expresión de la Incertidumbre en la Medición”, uno de los más referenciados [105].

A pesar de no tener determinado este valor U, un estudio inicial de la exactitud ha sido llevado a cabo. La Tabla 16 muestra los valores obtenidos tanto con el método de referencia COSMIC-FFP así como con RmFFP.

**Tabla 16.** Comparación de movimientos de datos identificados con COSMIC-FFP y RmFFP

Tipo de Movimiento de dato	Valor de referencia (Cfsu)	Valor con RmFFP (Cfsu)	Diferencia
Entrada	4	3	1
Lectura	6	8	2
Escritura	1	1	0
Salida	1	1	0
<b>Total</b>	12	13	3

Según estos resultados, una diferencia de 1 Cfsu (mínimo valor de diferencia existente) fue obtenida con respecto al tamaño total del sistema SCV. Sin embargo, analizando los tamaños por tipo de movimiento de dato se observa que la proximidad es menor para el movimiento de dato tipo lectura. Una causa que origina esta diferencia, está en las condiciones para determinar

si la operación de abrir una válvula es lenta o rápida. Usando RmFFP, estas condiciones han sido tomadas en cuenta como movimientos de dato tipo lectura; sin embargo, estas mismas condiciones no fueron consideradas como movimientos de dato por el grupo de expertos que usaron COSMIC-FFP.

#### ***7.4. Precisión y productividad: un estudio empírico***

Tal como se observa en la Figura 36, un estudio empírico fue llevado a cabo para evaluar la precisión de RmFFP y la productividad de los usuarios al aplicar este procedimiento. La precisión de RmFFP ha sido verificada en términos de su reproducibilidad, atributo definido en el informe técnico ISO/IEC 14143-3 [18]. A pesar de que la productividad no es considerada por este informe técnico, su evaluación ha sido también llevada a cabo con el fin de tener algún indicativo de si las reglas definidas en el procedimiento RmFFP eran fáciles de ser aplicadas manualmente.

A continuación se describe el presente estudio empírico que fue llevado a cabo de acuerdo al marco de ingeniería de software empírica propuesto por Wholin et al. [79]

##### ***7.4.1. Definición***

Para la definición del objetivo del presente estudio se utilizó la plantilla Goal/Question/Metric (GQM) [78]. Este objetivo es definido a continuación:

***“Analizar el procedimiento de medición RmFFP para el propósito de verificar su reproducibilidad y evaluar la productividad de los usuarios desde el punto de vista del investigador en el contexto de estudiantes de informática que aplican RmFFP para obtener el tamaño funcional de las especificaciones de requisitos de OO-Method”.***



Dos preguntas de investigación fueron direccionadas a partir de este objetivo:

- ¿Será RmFFP un procedimiento de medición reproducible?
- ¿La productividad de los usuarios será mayor en comparación con informes encontrados en la literatura?

#### **7.4.2. *Diseño experimental***

##### **7.4.2.1. *Selección de sujetos***

Los sujetos seleccionados fueron estudiantes de la Facultad de Informática de la Universidad Politécnica de Valencia con conocimiento similar en el uso del Modelo de Requisitos de OO-Method. Estos estudiantes estuvieron matriculados en el curso “Ambientes de desarrollo del software”, cursado desde febrero hasta junio del 2005. El experimento fue organizado como parte obligatoria de este curso. El presente estudio experimental estuvo conformado por 35 estudiantes, número total de matriculados en el curso seleccionado.

##### **7.4.2.2. *Objeto experimental***

Los objetos, sobre el cual el presente estudio empírico fue llevado a cabo, son las especificaciones de requisitos usando OO-Method para los casos de estudio “Alquiler de Vehículos” y “Gestión de Servicios de Mantenimiento”.

##### **7.4.2.3. *Selección de variables***

Teniendo en cuenta la clasificación de variables propuesta por Juristo y Moreno [78], identificamos en primer lugar las *variables respuesta* definidas como el resultado del experimento, el cual debe ser cuantificable. Concretamente, los datos que son obtenidos a partir del experimento son las *medidas de tamaño funcional* obtenidas y el *tiempo de demora* para medir una especificación de requisitos. A partir de estas variables se analizó:

La *reproducibilidad*, mediante la obtención del grado de variabilidad existente en las medidas resultantes de aplicar RmFFP.

La *productividad*, obtenida por el número de unidades de tamaño Cfsu que pueden ser medidas por una unidad de tiempo.

En segundo lugar, identificamos los *factores* definidos como las características que afectan a las variables respuesta. En el presente estudio, el *procedimiento de medición*, que usan los sujetos para llevar a cabo las mediciones, es un factor que afecta a las variables respuesta.

En tercer lugar, identificamos los *parámetros* definidos como las características que no influyen o no se desea que influyan a las variables respuesta. En el presente estudio se consideró como parámetros a la *calidad de la especificación de requisitos de los casos de estudio*, *la similaridad de conocimientos en medición de tamaño funcional* y *la familiaridad de los sujetos usando el método OO-Method* y *la herramienta de especificación RETO*.

#### 7.4.2.4. Formulación de hipótesis

Las siguientes hipótesis fueron definidas:

**Hipótesis 1:** A la pregunta si RmFFP es reproducible:

*Hipótesis nula*,  $H_{10}$ : Las medidas de tamaño funcional no son reproducible al aplicar RmFFP.

*Hipótesis alternativa*,  $H_{11}$ : Las medidas de tamaño funcional son reproducible al aplicar RmFFP.

**Hipótesis 2:** A la pregunta si los usuarios son más productivos usando RmFFP en comparación con reportes encontrados en la literatura:

*Hipótesis nula*,  $H_{20}$ : Los usuarios son menos productivos usando RmFFP cuando es comparado a otros informes encontrados en la literatura.

*Hipótesis alternativa*, H2<sub>1</sub>: Los usuarios son más productivos usando RmFFP cuando es comparado a otros informes encontrados en la literatura.

#### **7.4.2.5. Instrumentos**

Los instrumentos utilizados en este estudio fueron los objetos experimentales y los materiales de entrenamiento.

El objeto experimental fue una especificación de requisitos de OO-Method del caso de estudio “Alquiler de Vehículos”. Las especificaciones funcionales de este caso de estudio fueron llevadas a cabo por los mismos estudiantes; sin embargo, para la tarea de medición, utilizamos la especificación propuesta por el profesor del curso.

Los materiales de entrenamiento fueron los siguientes: un conjunto de diapositivas instructivas sobre el modelo de requisitos OO-Method y el procedimiento de medición RmFFP; un caso de estudio que describe un ejemplo de la aplicación de RmFFP, una guía de medición, y otro caso de estudio para verificar el entrenamiento realizado.

#### **7.4.2.6. Tareas experimentales**

En primer lugar, se inició con un conjunto de tareas que tenían como fin desarrollar un nivel de experticia requerido para medir especificaciones de requisitos usando RmFFP. Para esto, se aplicó un método de entrenamiento llamado demostración/práctica [102]. En la parte de demostración se consideró las siguientes tareas:

- Presentación del modelo de requisitos de OO-Method.
- Uso de la herramienta RETO.
- Presentación del procedimiento de medición RmFFP.

- Ilustración del uso de RmFFP con un módulo del caso de estudio “Gestión de Servicios de Mantenimiento”.

En la parte práctica se consideró las siguientes tareas:

- Especificación de un caso de estudio de “Alquiler de Vehículos” usando la herramienta RETO.
- Aplicación guiada de RmFFP a módulos del caso de estudio “Gestión de Servicios de Mantenimiento” (los estudiantes pueden clarificar sus dudas).
- Verificación de lo aprendido mediante la medición de ciertos módulos del caso de estudio “Gestión de Servicios de Mantenimiento”

El tiempo utilizado para esta sesión de entrenamiento fue de ocho horas distribuidas en cuatro días.

En segundo lugar, una vez llevado a cabo el entrenamiento de los sujetos, se inició la tarea de medición, donde cada sujeto aplicó RmFFP en el caso de estudio “Alquiler de Vehículos”. La especificación de este caso de estudio fue la misma para todos los sujetos, la cual fue revisada por el autor del Modelo de Requisitos de OO-Method [81]. Esta tarea fue llevada a cabo para obtener el tamaño funcional y el tiempo de medición obtenido por cada sujeto.

#### **7.4.3. Ejecución**

Para que el diseño del estudio empírico fuera ejecutado adecuadamente, la interacción entre los sujetos fue controlada para evitar plagios. No hubo límite en el tiempo para medir los objetos experimentales.

#### **7.4.4. Análisis e interpretación**

El análisis y la interpretación de los resultados han sido llevados a cabo con respecto a la reproducibilidad de las medidas obtenidas y con

respecto a la productividad de los sujetos usando RmFFP. Este análisis fue llevado a cabo usando la herramienta estadística SPSS versión 12.0. A continuación se describe los resultados obtenidos.

#### 7.4.4.1. Análisis de la reproducibilidad de RmFFP

Este análisis de la reproducibilidad ha sido llevado a cabo en 2 niveles: en primer lugar con respecto a las medidas de tamaño funcional obtenidas para toda la capa de software (aplicación), y en segundo lugar un análisis con respecto a las medidas de tamaño funcional obtenidas por tipo de movimiento de datos.

##### *Nivel 1: Por capa de software*

Para medir el grado de variación entre las medidas de tamaño funcional obtenidas por los diferentes sujetos usando RmFFP, se utilizó una ecuación estadística propuesta por Kemerer [103], la cual fue utilizada exitosamente por Abrahão [53]. Esta ecuación fue calculada tomando la diferencia entre el valor del tamaño producido por un sujeto  $i$  y el valor promedio producido por los otros “ $n-1$ ” sujetos de la muestra, dividida por este valor promedio. Cada resultado obtenido, el cual es representado por “REPi”, fue interpretado de la siguiente manera: “*mientras el valor sea más cercano a cero, existirá menos variabilidad en la medida obtenida por el sujeto  $i$* ”. La ecuación aplicada se muestra a continuación:

$$REP_i = \left| \frac{\sum_{k=1, k \neq i}^n \frac{Tamaño_k}{n-1} - Tamaño_i}{\sum_{k=1, k \neq i}^n \frac{Tamaño_k}{n-1}} \right| \quad (1)$$

Estos valores REPi obtenidos para cada sujeto son presentados en la Tabla B2 del Apéndice B. El promedio de variabilidad (REPi) para las treinta y tres observaciones fue de 0,051 (Vea Tabla 17). Este valor al ser comparado

con el resultado reportado por Kemerer (26,53%) en un estudio acerca de la fiabilidad de IFPUG FPA, permite concluir que la reproducibilidad es mayor usando el procedimiento RmFFP ya que su variabilidad fue mucho menor.

**Tabla 17.** Estadística descriptiva para la variabilidad de RmFFP

Variable	Promedio	Desviación estándar	Mínimo	Máximo
Variabilidad	0,051	0,04164	0,00	0,15

Para probar la hipótesis nula  $H_{10}$ , previamente se comprobó la normalidad de los valores REPi utilizando la prueba de Shapiro-Wilk. Como los datos fueron normales, se utilizó la técnica estadística t-student, considerándose al valor de cero como valor de referencia para ser comparado con cada uno de los valores REPi obtenidos.

Los resultados de la prueba, mostrados en la Tabla 18, permiten el rechazo de la hipótesis nula debido a que el valor t obtenido está fuera de los valores del intervalo. Por lo tanto, podemos afirmar con un 95% de confianza que los datos obtenidos satisfacen la hipótesis que RmFFP es reproducible.

**Tabla 18.** Prueba de T-student para la variabilidad de RmFFP

Variable	Diferencia de medias	95% Intervalo de confianza para la diferencia	t	p-valor
Variabilidad	.05091	0,0361 (inferior) 0,0657 (superior)	7,024	0,000

Con el fin de profundizar en el análisis de la reproducibilidad de RmFFP, se llevó a cabo también un análisis a nivel de tipo de movimientos de datos, la que es explicado a continuación.

***Nivel 2: Por tipo de movimiento de dato***

Este nivel de análisis permitió analizar la fiabilidad de la identificación de movimientos de datos usando el procedimiento de medición RmFFP.

Aplicando la ecuación 1, se cuantificó la variabilidad de los tamaños funcionales por tipo de movimiento de dato. La Tabla 19 muestra el análisis descriptivo para estos valores cuantificados.

**Tabla 19.** Variabilidad de los tipos de movimientos de dato

<b>Variabilidad</b>	<b>Entrada</b>	<b>Lectura</b>	<b>Escritura</b>	<b>Salida</b>
Mínimo	0,018	0,008	0,004	0,000
Máximo	0,410	0,149	0,081	0,000
Promedio	0,117	0,046	0,020	0,000
Desviación estándar	0,095	0,048	0,020	0,000

Tal como se observa en la Tabla 19, la variabilidad obtenida para el movimiento de datos de tipo entrada es ligeramente mayor con respecto a la variabilidad de los otros tipos de movimientos de datos. Sin embargo, a pesar de esta pequeña diferencia obtenida, una revisión de la aplicación de la Regla 8 (identificación de movimiento de datos de tipo entrada) fue llevada a cabo. Como consecuencia de esta revisión, se identificó algunas consideraciones adicionales para la identificación de este tipo de movimiento de dato en los mensajes con el estereotipo <<signal>>. Estas consideraciones adicionales fueron presentadas en el capítulo 4.

#### **7.4.4.2. Análisis de la productividad con RmFFP**

Para calcular la productividad de medición de un sujeto usando RmFFP, se dividió el tamaño funcional por el tiempo de medición expresado en horas. Este tiempo de medición comprende del tiempo requerido para la aplicación de las representación, duplicidad y agregación. En la Tabla B2 del Apéndice B se presenta estos valores de tamaño y tiempo registrado por cada sujeto.

Tal como se observa en la Figura 42, la productividad obtenida por cada sujeto oscila entre 90-190 Cfsu/hora, siendo la productividad media de 131.48 Cfsu/hora.

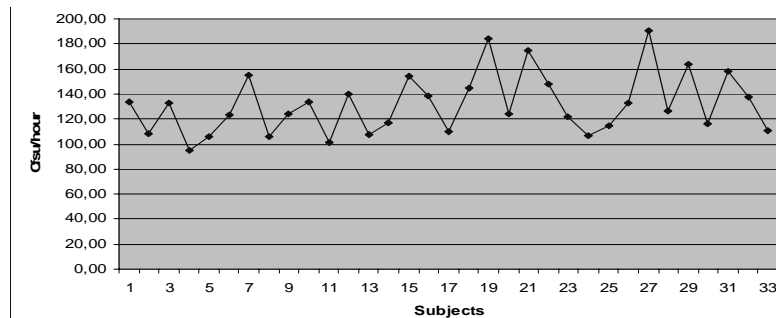


Figura 42. Productividad de medición para sujeto usando RmFFP.

La evaluación de esta productividad media con RmFFP fue llevado a cabo comparando con los valores reportados en la literatura. Para esta evaluación inicial, se consideró un informe publicado por Nagano [125], sobre la productividad de medición con COSMIC-FFP en 13 casos de estudio. El promedio de medición obtenida fue de 45 Cfsu/hora. Este valor es tres veces menor, comparado a la productividad obtenida con el procedimiento RmFFP. Sin embargo, varios factores pudieron afectar a esta diferencia, tales como.

- *Especificación de los requisitos funcionales proveída a los sujetos para que pueda ser medida*, Nagano especificó la funcionalidad de los sistemas de tiempo real en lenguaje natural; mientras que para RmFFP la funcionalidad fue especificada de manera semi-formal, usando el Modelo de Requisitos de OO-Method.
- *Reglas de representación*, Nagano aplicó directamente las reglas genéricas de COSMIC-FFP; mientras que con RmFFP, la aplicación de sus reglas de representación permitió la reducción de la generalidad de COSMIC-FFP.



- *Experiencia de los sujetos*, Nagano entrenó al personal en el uso de COSMIC-FFP; mientras que los usuarios de RmFFP fueron entrenados tanto en modelado de requisitos con OO-Method, como en el mismo procedimiento de medición.

Dado que sólo se encontró el trabajo de Nagano relacionado a la productividad con COSMIC-FFP, en el presente análisis se consideró también los datos reportados con el método IFPUG FPA [2]. Esta comparación fue factible, ya que tanto COSMIC-FFP como IFPUG FPA obedecen al estándar ISO 14143 [1].

Según experiencia divulgada por la industria, la compañía “Total Metrics” publicó diversos niveles de producción de medición [124]. Según estos niveles, la productividad de un estimador puede variar entre 200-750 PF/día. Como un día se asume que tiene 8 horas de trabajo, la tasa de productividad oscilaría entre 25-93.75 PF/hora, dependiendo de la experiencia de la persona. Teniendo en cuenta que la correlación entre los PF y Cfsu es lineal y que el factor de convertibilidad es cercano a 1 [24], la productividad obtenida con RmFFP fue cerca de cuatro veces más que los resultados reportados por Total Metrics. Aunque este análisis proporciona una cierta evidencia empírica de la buena productividad con RmFFP en este grupo de sujetos, aún no es posible generalizar sobre esta productividad de medición obtenida. A continuación se presenta las amenazas que han sido identificadas en el presente estudio empírico.

#### **7.4.5. Evaluación de la validación**

Existen varias amenazas en la validez de los resultados de un experimento, las cuales han sido identificadas por Wohlin et al. [79]. A continuación, se describirá las principales amenazas que afectan a la validez interna y externa del presente estudio empírico.

#### **7.4.5.1. Validez interna**

Los riesgos para este tipo de validez son las influencias que pueden afectar al procedimiento de medición con respecto a su causalidad, sin el conocimiento del investigador. Las siguientes amenazas fueron consideradas:

- *Selección.* Este es el efecto de la variación natural en el desenvolvimiento del ser humano. Dependiendo de cómo los sujetos son seleccionados para un grupo, los efectos de selección pueden variar. En este experimento, los sujetos fueron seleccionados por conveniencia. Es decir, se eligió estudiantes de último año que cursaban la asignatura de “Ambientes de Desarrollo del Software”. Este curso fue seleccionado porque incluía unidades especializadas, lo que facilitó que los estudiantes tuvieran la preparación necesaria dentro del alcance de este curso.
- *Maduración.* Este es el efecto que se puede producir cuando los sujetos reaccionan diferentemente con el transcurrir del tiempo; por ejemplo, el efecto de la fatiga. Este efecto se considera en nuestro caso no relevante, ya que el tiempo estimado para la medición fue de menos de dos horas.

#### **7.4.5.2. Validez externa**

La validez externa se refiere a la generalización de los resultados a la práctica. Las amenazas que fueron consideradas en el presente estudio son las siguientes:

- *La interacción de sujetos con el procedimiento.* Este es el efecto de no tener una población representativa en el experimento con el cual generalizar. En nuestro caso, aceptamos que más experimentos con sujetos profesionales de la industria es necesario.

- *La interacción del material con el procedimiento.* Este es el efecto asociado a no tener el material representativo para generalizar. En el experimento, intentamos utilizar una especificación de requisitos representativa de un caso real en el dominio de los sistemas de información de gestión. Sin embargo, más estudios empíricos son necesarios con más especificaciones del mismo dominio funcional.

### **7.5. Conclusiones**

En este capítulo se llevó a cabo la evaluación de la aplicación de RmFFP, mediante la verificación de la calidad de los resultados obtenidos, la evaluación de la productividad de los usuarios usando RmFFP, y un análisis metrológico del proceso de medición.

En cuanto a la verificación de la calidad de los resultados en términos de su exactitud y reproducibilidad, se demostró empíricamente que RmFFP es reproducible. Sin embargo, en cuanto a la exactitud de este procedimiento, no pudo ser corroborado de manera empírica, debido a varias razones, tales como la falta de determinación de un valor de incertidumbre para la medición de tamaño funcional software. Este valor de incertidumbre es comúnmente calculado en otras áreas, en base a factores tanto externos como internos que afectan un proceso de medición, los mismos que son luego modelizados matemáticamente. Otra razón fue la falta de “valores de referencia” con los que se pueda comparar. Sin embargo esta verificación inicial de la exactitud fue llevada a cabo en base a la cuarta parte del estándar ISO/IEC 14143, informe que define un conjunto de estándares de requisitos de referencia. Finalmente, con respecto al análisis metrológico se concluyó que RmFFP basado en el método estándar COSMIC-FFP si considera todos los conceptos de metrología relacionados con la categoría de “medición”.



# *Adopción en la práctica*

*“la experimentación nos permite adquirir conocimiento sobre qué hace al software bueno y cómo hacer bien el software”.*

*- Pfleeger-*

Actualmente, IFPUG FPA [25] se ha constituido en el método de medición de tamaño funcional con mayor aceptación en la industria del software. Sin embargo, en estos últimos años con la aparición del nuevo método estándar COSMIC-FFP [22], una variedad de procedimientos de medición basados en este método han ido surgiendo en un contexto académico con una posible proyección futura de ser utilizados en la práctica.

Este capítulo tiene como finalidad evaluar las percepciones de los usuarios cuando usan el procedimiento de medición RmFFP y predecir la intención a ser usado en un futuro. De este modo, el presente capítulo complementa la evaluación de las propiedades de RmFFP descritos en el anterior capítulo (reproducibilidad y exactitud).

El Modelo de Adopción de Métodos (MAM) propuesto por Moody [75] ha sido utilizado para llevar a cabo esta evaluación. El MAM está basado en el Modelo de Aceptación de la Tecnología de Davis (TAM) [76], un modelo para explicar y predecir la aceptación de la tecnología de la

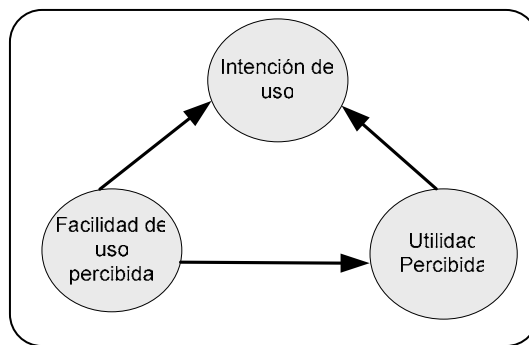
información por parte del usuario en base a un conjunto de constructores, los cuales fueron adaptados para explicar y predecir la adopción de métodos. Estos constructores son:

**Facilidad del Uso Percibida:** Grado de esfuerzo que una persona percibe al usar un método en particular.

**Utilidad Percibida:** Grado de eficacia de un método en particular que es percibido por una persona para el logro de sus objetivos propuestos.

**Intención de Uso:** Grado de intencionalidad que tiene una persona para usar un método particular.

La Figura 43 muestra cada uno de estos constructores y las relaciones existentes entre ellos.



**Figura 43.** Modelo de Adopción de métodos [75]

Este Modelo reconoce que las percepciones de eficiencia (facilidad de uso) y efectividad (utilidad) de un método juegan un rol importante para que dicho método sea adoptado en la práctica.

A continuación se presenta la descripción general de la evaluación llevada a cabo sobre la intención de uso de RmFFP.

### 8.1. Descripción general

El proceso de evaluación llevado a cabo se inició con la selección de participantes, quienes formaron parte de un proceso de entrenamiento. Este entrenamiento tiene como propósito desarrollar cierto nivel de experticia requerido en los sujetos para medir especificaciones de requisitos usando el procedimiento de medición RmFFP. Tal como se observa en la Figura 1, un conjunto de materiales de entrenamiento fue utilizado, el cual fue previamente diseñado y elaborado. Este material consta de casos de estudio especificados con el Modelo de Requisitos de OO-Method [77], ejemplos del uso de RmFFP y guía de medición.

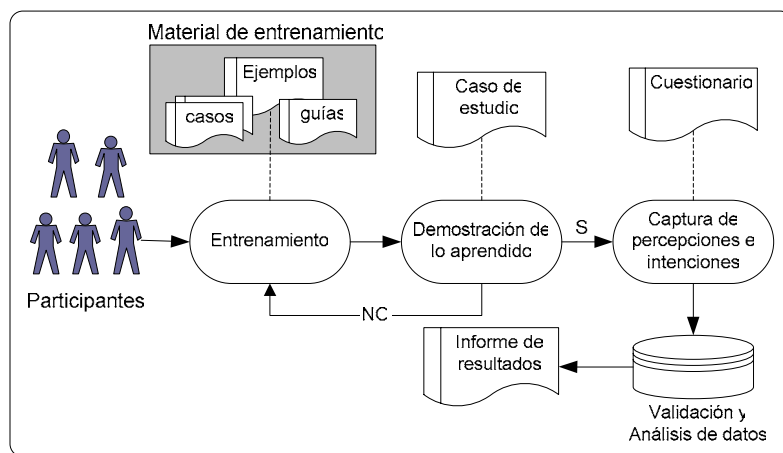


Figura 44. Proceso de evaluación de adopción en la práctica de RmFFP

Finalizado el proceso de entrenamiento, los participantes inician la demostración de lo aprendido, mediante la medición de un caso de estudio que fue previamente seleccionado. Si esta demostración no es satisfactoria, se reforzará nuevamente con la sesión de entrenamiento. En caso contrario, se procederá a la captura de percepciones e intenciones que tienen los participantes sobre el uso de RmFFP mediante la aplicación de un cuestionario, el cual es descrito con mayor detalle en la sección 8.2.2.5.

Luego, cada de una de las respuestas de los participantes es registrada, validada y analizada para finalmente ser interpretadas y presentadas en un informe.

La fiabilidad de estos resultados obtenidos debiera ser asegurada. Para esto, es necesario considerar ciertas variables que podrían afectar a los resultados. Juristo y Moreno [78] recomiendan dos perspectivas diferentes a tomar en cuenta para la identificación de tipos de variables en un experimento de software:

Desde una *perspectiva externa* se tiene a aquellas variables que no pueden ser modificadas o ajustadas dentro del proyecto del software. Por ejemplo, las características de los usuarios que pueden afectar al proceso de desarrollo, las características del problema que se está intentando resolver, las características de organización donde el software es desarrollado, algunas fuentes de información y restricciones de clientes.

Desde una *perspectiva interna* se tiene a aquellas variables que son configuradas al inicio o durante el proyecto. Por ejemplo: los procesos, métodos, herramientas, personal y productos.

Considerando estas dos perspectivas, en el proceso de evaluación (Figura 44) surgen algunas preguntas importantes, tales como: ¿qué nivel de familiaridad en el uso del método OO-Method será necesario?, ¿qué nivel de experiencia de los participantes será requerido?, ¿qué casos de estudio serían los más representativos?, etc. Estas preguntas son respondidas en la siguiente sección.

## **8.2. Adopción de RmFFP en la práctica**

Con el fin de llevar a cabo un estudio más riguroso y sistemático sobre la posible adopción de RmFFP en la práctica, un estudio empírico fue llevado a cabo en base a los marcos de experimentación propuestos por



Nuristo et al. [78] y Wholin et al. [79]. A continuación se describe la definición del presente estudio empírico

### 8.2.1. *Definición*

Usando la plantilla de Goal/Question/Metric (GQM) [80], el objetivo del presente estudio empírico es:

*Analizar* el procedimiento de medición RmFFP *para* evaluar y predecir *su* posible adopción en la práctica *desde el punto de vista* de los medidores<sup>3</sup> *en el contexto de* estudiantes de informática que miden especificaciones de requisitos OO-Method.

Teniendo en cuenta el MAM reconoce que las percepciones de los usuarios sobre un método juegan un rol importante para predecir su adopción en la práctica, las interrogantes de investigación identificadas a partir de la meta definida son las siguientes:

¿RmFFP es percibido como fácil de usar?

¿RmFFP es percibido como útil?

¿Existe la intención en usar RmFFP en el futuro?

¿Es la intención de uso realmente un resultado de las percepciones experimentadas por los sujetos al usar RmFFP?

### 8.2.2. *Diseño*

#### 8.2.2.1. *Selección de sujetos*

Los sujetos seleccionados fueron estudiantes de la Facultad de Informática de la Universidad Politécnica de Valencia con conocimiento similar en el uso del Modelo de Requisitos de OO-Method. Estos estudiantes

---

<sup>3</sup> El término “medidores” se refiere a las personas que realizan una medición del software; por ejemplo: un gestor de proyectos

estuvieron matriculados en el curso “Ambientes de desarrollo del software”, cursado desde febrero hasta junio del 2005. El experimento fue organizado como parte obligatoria de este curso. Además se formaron dos grupos de estudiantes debido a que algunos no podían asistir con regularidad a clases por diferentes motivos. El primer grupo estuvo formado por 18 estudiantes y el segundo grupo estuvo formado por 17 estudiantes. De este modo, se contó con la asistencia de todos los estudiantes. Por lo tanto, el presente estudio experimental estuvo conformado por 35 estudiantes, número total de matriculados en el curso seleccionado.

#### **8.2.2.2. Objeto experimental**

Los objetos, sobre el cual el presente estudio empírico es llevado a cabo, vienen a ser las especificaciones de requisitos usando OO-Method para los casos de estudio “Alquiler de Vehículos” y “Gestión de Servicios de Mantenimiento”.

#### **8.2.2.3. Selección de variables**

Teniendo en cuenta la clasificación de variables propuesta por Juristo y Moreno [78], identificamos en primer lugar las *variables respuesta* definidas como el resultado del experimento, el cual debe ser cuantificable. Es decir, estas variables reflejan los datos que son recogidos a partir del experimento. Tal como se observa en la Figura 44, la captura de percepciones e intenciones es llevada a cabo mediante la aplicación de un cuestionario. Este cuestionario incluye una serie de preguntas sobre la facilidad de uso percibida, la utilidad percibida y la intención de uso. Por consiguiente, las variables respuesta para el presente estudio fueron la *facilidad de uso y utilidad percibida, y la intención de uso*, las cuales fueron cuantificadas mediante puntajes en una escala de 1 a 5.

En segundo lugar identificamos los *factores* definidos como las características que afectan a la variable respuesta. Un factor que afecta a las variables respuesta es el *procedimiento de medición* que usan los sujetos para llevar a cabo las mediciones. En este estudio, el procedimiento de medición utilizado fue RmFFP.

En tercer lugar identificamos los *parámetros* definidos como las características que no influyen o no se desea que influyan en las variables respuesta. De este modo, los resultados serán resultados obtenidos bajo las condiciones definidas como parámetros. En el presente estudio se consideró como parámetros a la calidad de la especificación de requisitos de los casos de estudio, la similitud de conocimientos en medición de tamaño funcional y la familiaridad de los sujetos usando el método OO-Method y la herramienta de especificación RETO.

#### 8.2.2.4. Formulación de hipótesis

Las hipótesis formuladas a partir de las preguntas de investigación definidas previamente son las siguientes:

**Hipótesis 1:** A la pregunta si RmFFP es percibido como fácil de usar, se formularon los siguientes tipos de hipótesis:

*Hipótesis nula, H1<sub>0</sub>:* RmFFP no es percibido como fácil usar.

*Hipótesis alternativa, H1<sub>1</sub>:* RmFFP es percibido como fácil usar.

**Hipótesis 2:** A la pregunta si RmFFP es percibido como útil, se formularon los siguientes tipos de hipótesis:

*Hipótesis nula, H2<sub>0</sub>:* RmFFP no es percibido como útil.

*Hipótesis alternativa, H2<sub>1</sub>:* RmFFP es percibido como útil.

**Hipótesis 3:** A la pregunta si existe la intención de usar RmFFP en el futuro, se formularon los siguientes tipos de hipótesis:

*Hipótesis nula, H3<sub>0</sub>:* No hay intención de usar RmFFP.

*Hipótesis alternativa, H3<sub>1</sub>:* Hay intención de usar RmFFP.

**Hipótesis 4:** A la pregunta si la intención de uso es realmente un resultado de las percepciones experimentadas por los sujetos al usar RmFFP, se formularon los siguientes tipos de hipótesis:

*Hipótesis nula, H4<sub>0</sub>:* La intención de uso no es influenciada por la utilidad percibida ni por la facilidad de uso percibida.

*Hipótesis alternativa, H4<sub>1</sub>:* La intención de uso es influenciada por la utilidad percibida y facilidad de uso percibida.

Además, teniendo en cuenta las relaciones establecidas en el MAM, las siguientes hipótesis fueron definidas:

**Hipótesis 5:** MAM mantiene la facilidad de uso percibida del TAM como una determinante directa de la utilidad percibida.

- *Hipótesis Nula (H5<sub>0</sub>):* La utilidad percibida de RmFFP no es determinada por su facilidad de uso percibida.
- *Hipótesis Alternativa (H5<sub>1</sub>):* La utilidad percibida de RmFFP es determinada por su facilidad de uso percibida.

**Hipótesis 6:** MAM mantiene la facilidad de uso percibida de TAM como una determinante directa de la intención de uso.

- *Hipótesis Nula (H6<sub>0</sub>):* La intención a usar RmFFP no es determinada por su facilidad de uso percibida.

- *Hipótesis Alternativa (H6<sub>a</sub>)*: La intención a usar RmFFP es determinada por la facilidad de uso percibida.

**Hipótesis 7:** MAM mantiene la utilidad percibida de TAM como una determinante directa de la intención de uso.

- *Hipótesis Nula (H7<sub>0</sub>)*: La intención a usar RmFFP no es determinada por la utilidad percibida.
- *Hipótesis Alternativa (H7<sub>a</sub>)*: La intención a usar RmFFP es determinada por la utilidad percibida.

#### 8.2.2.5. Instrumento experimental:

Como instrumento experimental, un cuestionario fue adaptado de [53] para evaluar las “variables respuesta” identificadas. Este cuestionario incluyó trece preguntas de tipo cerradas-valorativas. La escala de valoración utilizada fue la *escala Likert* de 5 puntos. Cada pregunta fue formulada en formato “Afirmativo-Negativo”. Sin embargo, algunas preguntas fueron iniciadas en formato opuesto “Negativo-Positivo”. Dichas preguntas fueron ubicadas al azar para evitar respuestas monótonas en los sujetos.

La “variable respuesta” *facilidad de uso percibida* fue evaluada usando las 5 preguntas las que son presentadas a continuación:

P1	El proceso a seguir para aplicar el procedimiento de medición es simple y fácil de seguir
P3	De manera general, el procedimiento de medición es fácil de utilizar
P4	Las reglas de medición del procedimiento son claras y fáciles de entender
P6	El procedimiento de medición es fácil de aprender
P9	Me pareció fácil de aplicar el procedimiento de medición al caso de estudio

La “variable respuesta” *utilidad percibida* fue evaluada usando las 5 preguntas las que son presentadas a continuación:

P2	Creo que este procedimiento reduciría el tiempo requerido para medir el tamaño funcional de las aplicaciones orientado a objetos
P5	De manera general, el procedimiento de medición es útil
P8	Pienso que este procedimiento mejoraría la precisión de las mediciones de las aplicaciones orientado a objetos
P10	De manera general, pienso que este procedimiento proporcionó una manera eficaz de medir el tamaño funcional de aplicaciones orientado a objetos durante la fase de especificación de requisitos
P11	El uso de este procedimiento mejoraría mi rendimiento en la medición del tamaño funcional de aplicaciones orientado a objetos

Finalmente, la “variable respuesta *intención de uso* fue evaluada usando las 3 preguntas las que son presentadas a continuación:

P7	Utilizaré este procedimiento si tengo que medir el tamaño funcional de las aplicaciones orientadas a objetos en el futuro
P12	Pienso que sería fácil ser hábil usando este procedimiento
P13	Tengo la intención de utilizar este procedimiento de medición en el futuro

El diseño de este cuestionario es presentado en el Apéndice B.

#### 8.2.2.6. Tareas experimentales

Tal como se ilustró en la Figura 44, tres grupos de tareas fueron llevadas a cabo con todos los sujetos del estudio empírico: entrenamiento, demostración de lo aprendido y captura de las percepciones e intenciones.

- En cuanto al entrenamiento llevado a cabo, se siguió el método de entrenamiento llamado *demostración/práctica* [102], las tareas relacionadas a la parte de “*demostración*” fueron las mismas que se describieron en la sección 7.4.2.6 (Página - 170 -). En cuanto a la parte de “*práctica*” se realizó una aplicación guiada de RmFFP a módulos de

los casos de estudio “Gestión de Servicios de Mantenimiento” y Gestión de Partidos de Golf”.

El tiempo utilizado para esta sesión de entrenamiento fue de ocho horas distribuidas en cuatro días.

En cuanto a la demostración de lo aprendido, los sujetos aplicaron RmFFP en el caso de estudio “Alquiler de Vehículos”. La especificación de este caso de estudio fue la misma para todos los sujetos, la cual fue revisada por el autor del Modelo de Requisitos de OO-Method en [81].

En cuanto a la captura de percepciones e intenciones, cada sujeto respondió a un cuestionario. Esta tarea fue llevada a cabo para recoger los datos sobre las “variables respuesta” previamente identificadas (facilidad de uso percibida, utilidad percibida e intención de uso).

Antes de que los sujetos procediesen a la ejecución de las tareas, se condujo una de prueba del cuestionario para mejorar la claridad de las preguntas. Como resultado de esta prueba, no fue necesario realizar ningún cambio.

### **8.2.3. Ejecución**

Para que el diseño del estudio empírico sea ejecutado adecuadamente, la interacción entre los sujetos fue controlada para evitar que las respuestas sean influenciadas por opinión de otros sujetos. Sin embargo, ellos podían consultar al profesor del curso cualquier duda con respecto al cuestionario. No hubo límite en el tiempo para la ejecución de la encuesta.

### **8.2.4. Análisis e interpretación**

Las hipótesis H1, H2 y H3, formuladas en la sección 9.2.4, han sido probadas formalmente con la intención de verificar si los puntajes asignados por los sujetos a cada una de las preguntas son perceptiblemente mejores al

puntaje medio de la escala likert (3). Es decir que sus hipótesis nulas ( $H_0$ ) y alternativas ( $H_a$ ) fueron definidas formalmente de la siguiente manera:

$H_0: \mu \leq 3, \quad \alpha = 0.05$ $H_a: \mu > 3$
---

Donde  $\alpha$  representa el 5% de significancia, es decir que existe un 0,05 de probabilidad de aceptar la hipótesis alternativa cuando la hipótesis nula es verdadera (Error tipo I).

El análisis de los datos fue llevado a cabo con la herramienta estadística SPSS. A continuación, se describe este análisis e interpretación llevado a cabo para evaluar la eficiencia y utilidad percibida, así como la intención a usar RmFFP.

**8.2.4.1. Evaluación de la facilidad de uso percibida**

En primer lugar para evaluar la facilidad de uso percibida por cada uno de los sujetos al usar el procedimiento de medición RmFFP, se aplicó la siguiente fórmula:

$$FUP_i = \frac{\sum_{j=1}^n P_j}{n} \quad \begin{matrix} 1 \leq P \leq 5 & n = 5 \\ 1 \leq i \leq 33 & i, j \in N \end{matrix} \quad (1)$$

Donde:  $P_j$  representa el puntaje que oscila entre 1 y 5 para la j-ésima pregunta; n representa el número de preguntas del cuestionario que fueron consideradas (P1, P3, P4, P6 y P9) y  $FUP_i$  representa el valor promedio de la facilidad de uso percibida por cada i-ésimo sujeto. En la Tabla B1 del Apéndice B se presenta los puntajes asignados por cada sujeto.



En segundo lugar, un análisis descriptivo es llevado a cabo sobre estos valores FUP, la cual es presentada en la Tabla 20. Obsérvese que la media aritmética fue de 4,24 un valor claramente superior al valor 3 (puntaje medio). El rango de valores osciló entre un mínimo de 3,2 y un máximo de 5. Además un desvío de sólo 0,519 fue obtenido con respecto a la media, lo que significa que es una distribución homogénea y por consiguiente la media aritmética obtenida es representativa.

**Tabla 20.** Valores descriptivos para la Facilidad de Uso Percibida

Variable	Media	Desviación estándar	Mínimo	Máximo
FUP	4,2424	0,51903	3,20	5,0

Finalmente, con el fin de poder generalizar estos resultados obtenidos, la prueba de la hipótesis H1 fue llevada a cabo. Para esto, aplicando la prueba de Kolmogorov-Smirnov se observó que los datos presentaban una distribución normal. Por consiguiente, la prueba de t-student fue utilizada para verificar H1<sub>0</sub>:  $\mu \leq 3$ .

Los resultados obtenidos (vea Tabla 21) permiten el rechazo de la hipótesis nula H1<sub>0</sub>, ya que el valor de t está fuera del intervalo de confianza, siendo el nivel de significancia muy alto (p-valor<0.001). Por lo tanto, podemos afirmar con un 95% de confianza que RmFFP es fácil de usar.

**Tabla 21.** Prueba de T-Student para la Facilidad de Uso Percibida

Variable	Diferencia de medias	95% Intervalo de confianza	t	p-valor
FUP	1,24242	1,0584 (inferior) 1,4265 (superior)	13,751	.000

### 8.2.4.2. Evaluación de la utilidad percibida

Para evaluar la utilidad percibida, se procedió en primer lugar a calcular el promedio de los puntajes asignados por cada sujeto, aplicando la misma fórmula (1) vista anteriormente. Cinco preguntas -P2, P5, P8, P10 y P11- fueron consideradas para evaluar la utilidad percibida.

Luego, un análisis descriptivo fue también llevado a cabo sobre estos valores promedio (Vea Tabla 22)

**Tabla 22.** Valores descriptivos para la Utilidad Percibida

Variable	Media	Desviación estándar	Mínimo	Máximo
UP	3,94	0,411	3,00	4,60

La media aritmética obtenida fue de 3,94, valor que supera también el puntaje medio. No hubo valor alguno por debajo de este puntaje, ya que el rango osciló entre 3,0 y 4,6. La desviación obtenida con respecto a la media fue de sólo 0,411, por consiguiente el valor de la media obtenido con los 33 sujetos es representativo. Sin embargo, este valor no es generalizable. Para esto, se llevó a cabo una prueba de la hipótesis H2. Con el fin de verificar la normalidad de los datos, se aplicó la prueba de Kolmogorov-Smirnov. Como se observó que los datos presentaban una distribución normal, la prueba de t-student fue utilizada para verificar H2<sub>0</sub>:  $\mu \leq 3$ .

**Tabla 23.** Prueba de T-Student para la Utilidad Percibida

Variable	Diferencia de medias	95% Intervalo de confianza	t	p-valor
UP	0,93939	0,7937 (inferior) 1,0850 (superior)	13,138	0,000

Los resultados obtenidos (vea Tabla 21) permiten el rechazo de la hipótesis nula  $H_{2_0}$ , ya que el valor de  $t$  está fuera del intervalo de confianza, siendo el muy alto nivel de significancia ( $p\text{-valor} < 0.001$ ). Por lo tanto, podemos afirmar con un 95% de confianza que RmFFP es percibido como útil.

#### 8.2.4.3. Evaluación de la intención de uso

Al igual que en las anteriores evaluaciones, con el fin de representar un valor representativo de los puntajes asignados por cada uno de los sujetos que usaron el procedimiento de medición RmFFP, se calculó el promedio aplicando la misma fórmula (1), con la excepción de que  $n$  para este caso es igual a 3 preguntas (P7, P12 y P13). Un análisis descriptivo de estos valores fue llevado a cabo, el cual es presentado en la Tabla 24. La media aritmética fue de 3,86 un valor perceptiblemente mejor al puntaje medio (3). Sin embargo, aunque esta media aritmética es aceptable, el rango de valores inicia desde un valor menor a 3, tal como es de 2,67. No obstante, analizando la desviación estándar obtenida, podemos afirmar que la media aritmética obtenida para los 33 sujetos es representativa.

**Tabla 24.** Valores descriptivos para la Intención de Uso

Variable	Media	Desviación estándar	Mínimo	Máximo
IU	3,86	0,493	2,67	4,67

Para poder generalizar estos resultados iniciales, una prueba de la hipótesis  $H_3$  fue llevada a cabo. Para esto, aplicando la prueba de Kolmogorov-Smirnov se observó que los datos también presentaban una distribución normal. Por esta razón, la prueba de  $t$ -student fue utilizada para verificar  $H_{3_0}: \mu \leq 3$ .

**Tabla 25.** Prueba de T-Student para la Intención de Uso

Variable	Diferencia de medias	95% Intervalo de confianza	t	p-valor
IU	0,86	0,6837 (inferior) 1,0335 (superior)	10,000	0,000

Los resultados obtenidos (vea Tabla 25) permiten el rechazo de la hipótesis nula  $H_{3_0}$ , ya que el valor de t está fuera del intervalo de confianza. Por lo tanto, podemos afirmar con un 95% de confianza que existe una intención a usar el procedimiento de medición RmFFP.

Con el fin de verificar si esta intención de uso (IU) es influenciada por las percepciones de facilidad de uso y utilidad, la técnica de análisis de regresión fue utilizada para probar la hipótesis **H4: FUP + UP → IU**

Donde la Facilidad de Uso Percibida y la Utilidad Percibida fueron utilizadas como variables predictoras y la intención de uso como variable predecible. La ecuación de regresión múltiple resultante del análisis fue: **IU = 1,21 + 0,39 \* UP + 0,26 \* FUP**

Los detalles del análisis de regresión múltiple son:

R cuadrado ( $r^2$ ) = 0,21

F-statistic = 3,914

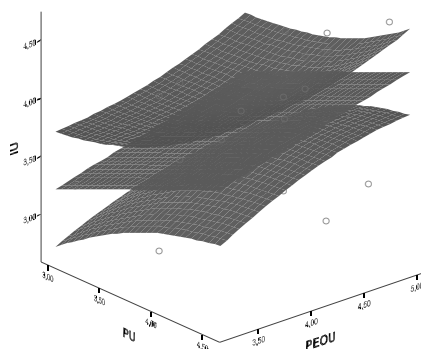
Nivel de significancia (p) = 0,031

Tal como se observa en la Tabla 26, el resultado de la regresión permite el rechazo de las hipótesis nula ( $H_{4_0}$ ) lo que significa que empíricamente podemos afirmar con un 95% de confianza que la intención de uso es influenciada por la facilidad de uso percibida y la utilidad percibida.

**Tabla 26.** Modelo de Regresión entre la FUP, UP y la IU

Modelo	Coeficientes no estandarizados		t	Sig.	95% Interv. confianza	
	B	Error típ.			Límite inf.	Límite sup.
Constante	1,21	,95	1,27	,213	-,731	3,150
FUP	,26	,16	1,65	,110	-,062	,576
UP	,39	,19	2,01	,054	-,007	,798

El coeficiente de determinación ( $r^2$ ) del análisis de regresión indica que el 21% de la variación en la intención de uso puede ser explicada por la variación que ocurre tanto en la facilidad de uso percibida, como en la utilidad percibida. La ecuación de regresión múltiple fue obtenida con un nivel de significancia media, ya que  $p < 0.05$ . La Figura 45 muestra este análisis de regresión para las variables en estudio.



**Figura 45.** Intención de Uso vs. Utilidad y Facilidad de Uso Percibida

#### 8.2.4.4. Evaluando las relaciones del MAM

En la presente sección, las hipótesis H5, H6 y H7 fueron probadas con el fin de evaluar si las relaciones del MAM (vea Figura 43) pueden ser comprobadas para explicar la adopción de los procedimientos de medición de tamaño funcional en la práctica. A continuación, la evaluación de cada una de estas relaciones es presentada:

***Facilidad de uso percibida vs. Utilidad percibida***

Para probar la hipótesis  $H5_0$ , la facilidad de uso percibida fue usada como variable predictora y la utilidad percibida como variable predecible. La ecuación de regresión lineal simple resultante del análisis fue:  **$UP = 3.46 + 0.11 * FUP$**

Los detalles del análisis de regresión son:

- R cuadrado ( $r^2$ ) = 0.02
- F-statistic = 0.633
- Nivel de significancia ( $p$ ) = 0.432

El coeficiente de determinación ( $r^2$ ) del análisis de regresión indica que sólo el 2% de la variación en la utilidad percibida podría ser explicada por la variación en la facilidad de uso percibida. Sin embargo, como la ecuación de regresión fue obtenida con un nivel de significancia nula ( $p > 0.1$ ), no podemos rechazar la hipótesis nula  $H5_0$ .

***Facilidad de uso percibida vs. Intención de uso***

Para probar la hipótesis nula  $H6_0$ , la facilidad de uso percibida fue usada como variable predictora y la utilidad percibida como variable predecible. La ecuación de regresión lineal resultante fue:  **$IU = 2.58 + 0.301 * FUP$**

Los detalles de este análisis de regresión son:

- R cuadrado ( $r^2$ ) = 0.101
- F-statistic = 3.467
- Nivel de significancia ( $p$ ) = 0.072

El coeficiente de determinación ( $r^2$ ) del análisis de regresión indica que el 10% de la variación en la intención de uso puede ser explicada por la variación en la facilidad de uso percibida. Sin embargo, como la ecuación de regresión fue obtenida con un bajo nivel de significancia ( $p < 0.1$ ), no podemos corroborar empíricamente que exista una relación entre la facilidad de uso percibida y la intención de uso.

#### ***Intención de uso vs. Utilidad Percibida***

Para probar la hipótesis nula  $H7_0$ , la utilidad percibida fue usada como variable predictora y la intención de uso como variable predecible. La ecuación de regresión lineal resultante del análisis llevado a cabo fue:  **$IU = 2.119 + 0.442 * UP$**

Los detalles del análisis de regresión son:

- R cuadrado ( $r^2$ ) = 0.14
- F-statistic = 4.847
- Nivel de significancia ( $p$ ) = 0.035

El coeficiente de determinación ( $r^2$ ) del análisis de regresión indica que el 14% de la variación en la intención de uso puede ser explicada por la variación en la utilidad percibida. Además, la ecuación de regresión fue obtenida con un nivel de significancia medio ( $p < 0.05$ ). Por lo tanto, existe un indicio de probar la relación existente entre la Utilidad Percibida y la Intención de Uso. Para corroborarlo empíricamente, una prueba de hipótesis fue llevada a cabo.

Tal como se observa en la Tabla 27, el resultado de la regresión permite el rechazo de la hipótesis nula ( $H7_0$ ), lo que significa que la utilidad percibida de RmFFP influye en la intención de uso.

Tabla 27. Regresión simple entre la UP y la IU

Modelo	Coeficientes no estandarizados		t	Sig.	95% Interv. confianza	
	B	Error típ.			Límite inf.	Límite sup.
Constante	2,119	0,794	2,668	0,012	0,499	3,739
FUP	0,442	0,201	2,202	0,035	0,033	0,851

Variable dependiente: IU

#### 8.2.5. Evaluación de la validez

Esta sección tiene como fin asegurar que los resultados experimentales sean válidos para la población. Existen varias amenazas en la validez de los resultados de un experimento, los cuales han sido identificados por Wohlin et al. [79]. A continuación, se describirá las principales amenazas para la validez de la conclusión, validez interna, validez de la construcción y validez externa del presente estudio empírico.

##### 8.2.5.1. Validez de la conclusión

Las amenazas para este tipo de validez se refieren a los aspectos que afectan a la capacidad de derivar correctas conclusiones acerca de las relaciones entre los factores, parámetros y los resultados obtenidos. Las siguientes amenazas fueron consideradas:

- *Fiabilidad de la aplicación del procedimiento a los sujetos.* Hay un riesgo en la variabilidad de la aplicación del procedimiento a diversas personas y en diversas ocasiones. En nuestro estudio, RmFFP fue aplicado siguiendo un mismo procedimiento diseñado para los dos grupos de sujetos. De este modo, el riesgo de obtener aplicaciones diferentes por diversos sujetos en distintas ocasiones disminuye.
- *Heterogeneidad de sujetos.* Todos los sujetos tenían aproximadamente el mismo nivel de experiencia para trabajar con el modelo de requisitos



de OO-Method. Sin embargo, esta homogeneidad conseguida reduce la validez externa del experimento.

#### 8.2.5.2. *Validez interna*

Las amenazas para este tipo de validez son las influencias que pueden afectar al factor con respecto a su causalidad, sin el conocimiento del investigador. Las siguientes amenazas fueron consideradas:

- *Instrumentación*. Este es el efecto causado por los instrumentos utilizados durante la ejecución del estudio. Con el fin de evitar efectos negativos, tales como la doble interpretación por parte de los sujetos al utilizar el cuestionario, la guía y los casos de estudio, se llevó a cabo una prueba de cada uno de estos instrumentos. Primero, las especificaciones de los casos de estudio (objetos a medir) fueron revisadas por el autor del modelo de requisitos de OO-Method. Segundo, la guía de medición y el cuestionario fueron probadas por un pequeño grupo de estudiantes con el fin de mejorar su comprensibilidad.

#### 8.2.5.3. *Validez de la construcción*

Las amenazas para este tipo de validez se refieren a las influencias que podrían afectar a la generalización de los resultados. La siguiente amenaza fue considerada:

- *Explicación inadecuada de los constructores*. Esta amenaza significa que los constructores no estén definidos adecuadamente y por lo tanto el experimento pueda no estar suficientemente claro. Para evitar esta amenaza, un *análisis de correlación de inter-ítems* fue llevado a cabo para evaluar la validez de los ítems que corresponden a los constructores del MAM. Esta evaluación fue basada en dos criterios propuestos por Campbell y Fiske [82]:

- La *validez convergente* (VC) se refiere a la convergencia entre los diversos indicadores utilizados para medir un constructor particular. La VC de un indicador es medido por la correlación promedio entre el indicador y los otros indicadores, los cuales se utilizan para medir el mismo constructor.
- La *validez divergente* (VD) se refiere a la divergencia de los indicadores utilizados para medir diversos constructores. La VD de un indicador es medida por la correlación promedio entre el indicador y los indicadores que se utilizan para medir un constructor diferente.

El promedio de la VD debe ser más bajo que el promedio de la VC. Los resultados del análisis de la validez para cada constructor se muestran en la

Tabla 28. Este análisis demuestra que el valor del VC es mayor al valor de la VD para todos los ítems de la facilidad de uso percibida, utilidad percibida, e intención de uso.

Además, también se realizó un análisis de fiabilidad del cuestionario diseñado. La fiabilidad del instrumento describe la consistencia en medir el mismo objeto en un cierto tiempo por diversas personas. Para esto, la fiabilidad fue llevada a cabo usando la técnica de Cronbach. La fiabilidad para cada constructor del MAM es presentada en la Tabla 29.

Tabla 28. Análisis de la validez constructiva del cuestionario

	Facilidad de Uso Percibida					Utilidad Percibida					Intención de Uso			Promedio		Válido
	P1	P3	P4	P6	P9	P2	P5	P8	P10	P11	P7	P12	P13	VC	VD	
<b>FUP</b> P1	1,00	0,52	0,45	0,61	0,31	0,17	0,26	-0,08	0,12	0,15	0,27	0,33	-0,02	0,58	0,15	<b>SI</b>
P3	0,52	1,00	0,22	0,39	0,20	-0,03	0,06	-0,17	-0,12	0,06	0,11	0,25	0,10	0,46	0,03	<b>SI</b>
P4	0,45	0,22	1,00	0,49	0,25	0,01	0,10	0,01	-0,12	0,32	0,19	0,22	0,09	0,48	0,10	<b>SI</b>
P6	0,61	0,39	0,49	1,00	0,23	-0,06	-0,09	-0,29	-0,18	0,06	0,04	0,06	-0,03	0,54	-0,06	<b>SI</b>
P9	0,31	0,20	0,25	0,23	1,00	0,04	0,22	0,13	0,27	0,20	0,25	0,15	0,18	0,40	0,18	<b>SI</b>
<b>UP</b> P2	0,17	-0,03	0,01	-0,06	0,04	1,00	0,22	-0,09	-0,02	0,12	0,30	0,18	-0,02	0,25	0,07	<b>SI</b>
P5	0,26	0,06	0,10	-0,09	0,22	0,22	1,00	0,26	0,26	0,44	0,21	0,00	0,05	0,44	0,10	<b>SI</b>
P8	-0,08	-0,17	0,01	-0,29	0,13	-0,09	0,26	1,00	0,47	0,13	0,31	-0,15	0,38	0,35	0,02	<b>SI</b>
P10	0,12	-0,12	-0,12	-0,18	0,27	-0,02	0,26	0,47	1,00	0,07	0,19	0,13	0,08	0,36	0,05	<b>SI</b>
P11	0,15	0,06	0,32	0,06	0,20	0,12	0,44	0,13	0,07	1,00	0,09	0,21	0,26	0,35	0,17	<b>SI</b>
<b>IU</b> P7	0,27	0,11	0,19	0,04	0,25	0,30	0,21	0,31	0,19	0,09	1,00	0,22	0,43	0,55	0,20	<b>SI</b>
P12	0,33	0,25	0,22	0,06	0,15	0,18	0,00	-0,15	0,13	0,21	0,22	1,00	0,02	0,41	0,14	<b>SI</b>
P13	-0,02	0,10	0,09	-0,03	0,18	-0,02	0,05	0,38	0,08	0,26	0,43	0,02	1,00	0,49	0,11	<b>SI</b>

**Tabla 29.** Fiabilidad para los constructores del MAM

<i>Constructor</i>	<i>Cronbach (<math>\alpha</math>)</i>
<i>Facilidad de uso percibida</i>	<i>0,7</i>
<i>Utilidad percibida</i>	<i>0,5</i>
<i>Intención de uso</i>	<i>0,5</i>

Sin embargo, el valor de alfa para todo el cuestionario fue de 0,7.

Según Nunally [83] podemos afirmar que un constructor es fiable si el valor de alfa es mayor o igual a 0,7. Por consiguiente, a pesar de obtener un alfa general de 0,7, es importante destacar que los ítems de utilidad percibida e intención requieren de algunos ajustes para mejorar su fiabilidad. Por esta razón, este estudio sobre la adopción de RmFFP en la práctica usando MAM fue repetido una vez más.

### **8.3. Replicación**

La replicación tiene como objetivo incrementar la fiabilidad de los resultados obtenidos. Para esto, se alteraron algunas cuestiones del diseño presentado en la sección 8.2.2, las cuales son descritas a continuación:

- En cuanto a los sujetos seleccionados, fue alterado el nivel de estudios y el número de sujetos. Se consideró once estudiantes de doctorado, quienes estuvieron matriculados en el curso de “Tecnologías de Software” de la Universidad Politécnica de Valencia durante el periodo de Febrero a Junio de 2006.
- En cuanto al objeto experimental, se adicionó un caso de estudio más, el cual trata sobre la gestión de un campo de Golf.
- En cuanto a la selección de variables, se mantuvo el mismo factor, pero además de RmFFP se introdujo el método Use Case Point como otra alternativa. Sin embargo, ello fue únicamente con fines informativos

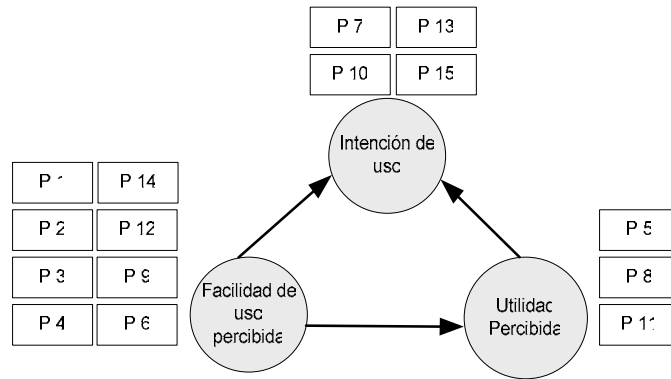
para que los sujetos puedan conocer otras formas de medir especificaciones de alto nivel.

- En cuanto el instrumento utilizado, el cuestionario fue ajustado (vea Apéndice C). En concreto, los cambios que se hicieron fueron los siguientes:
  - En la pregunta 7 se cambió el modo del tiempo: “utilizaré” por “utilizaría”. De esta forma se da un énfasis de posibilidad.
  - En la pregunta 8 se cambió el término “precisión” por el término “exactitud”.
  - Las preguntas 2 y 11 fueron reubicadas de la Utilidad Percibida a la Facilidad de Uso Percibida, debido a que las expresiones de “reducción de tiempo” y “rendimiento” están más relacionadas con la facilidad de uso que con la utilidad. Lo mismo se hizo para la pregunta 12 que fue también reubicada de la Intención de Uso a la Facilidad de Uso Percibida. De este modo tres preguntas fueron adicionadas para evaluar la Facilidad de Uso Percibida.
  - Se formularon dos preguntas más para evaluar la Intención de Uso, las cuales son :

P: Si tuviera algún proyecto de software a mi cargo, utilizaría este procedimiento de medición para apoyar a la gestión de dicho proyecto.

P: Si mi compañía necesitara algún procedimiento de medición de tamaño funcional, recomendaría el uso de RmFFP.

Por consiguiente la operacionalización del MAM para evaluar la adopción de RmFFP en la práctica fue la siguiente:



**Figura 46.** Operacionalización el Modelo MAM

El proceso seguido para ejecutar la repetición del presente estudio empírico fue el mismo que es ilustrado en la Figura 43. A continuación se resume la evaluación de la validez llevada a cabo.

### 8.3.1. Evaluación de la validez

Se tomaron en cuenta las mismas amenazas para cada uno de los tipos de validación anteriormente descritos. Sin embargo, nuevos resultados fueron obtenidos con respecto a la *validez de la construcción*, siendo estos favorables. A continuación se resume los dos tipos análisis llevados a cabo:

En cuanto al *análisis de correlación de inter-ítems*, los resultados son presentados en la Tabla 30.

El promedio de la VD es más bajo que el promedio de la VC para todos los ítems a excepción de las preguntas 2 y 12 que corresponden a la facilidad de uso percibida. Por consiguiente, eliminamos estas dos preguntas del cuestionario (Ver Tabla 31).

En cuanto al *análisis de fiabilidad* del cuestionario diseñado, usando alfa Cronbach para analizar las 15 preguntas y luego sin las preguntas 2 y 12, los resultados obtenidos fueron favorables. Dichos resultados son presentados en la Tabla 32.

**Tabla 30.** Análisis de la validez constructiva del cuestionario

	Facilidad de Uso Percibida								Utilidad Percibida			Intención de uso				Promedio		Válid
	P1	P2	P3	P4	P6	P9	P12	P14	P5	P8	P11	P7	P10	P13	P15	VC	VD	
FUP P1	1,00	-0,07	0,63	0,36	0,28	0,61	0,56	0,52	0,26	0,32	0,33	0,69	0,30	0,71	0,47	0,49	0,44	SI
P02	-0,07	1,00	-0,31	-0,44	-0,13	-0,26	0,27	0,09	0,38	0,15	0,09	0,31	0,13	0,24	0,08	0,02	0,20	NO
P03	0,63	-0,31	1,00	0,79	0,07	0,63	0,01	0,72	0,31	0,03	0,18	0,18	-0,11	0,29	0,39	0,44	0,18	SI
P04	0,36	-0,44	0,79	1,00	0,48	0,25	-0,12	0,49	0,28	-0,11	-0,12	-0,12	0,07	-0,07	0,27	0,35	0,03	SI
P06	0,28	-0,13	0,07	0,48	1,00	0,03	0,23	0,12	0,13	0,29	0,04	0,23	0,65	-0,08	0,20	0,26	0,21	SI
P09	0,61	-0,26	0,63	0,25	0,03	1,00	0,18	0,74	0,26	0,43	0,60	0,50	-0,11	0,44	0,37	0,40	0,35	SI
P12	0,56	0,27	0,01	-0,12	0,23	0,18	1,00	-0,02	0,47	0,79	0,70	0,82	0,72	0,83	0,68	0,27	0,71	NO
P14	0,52	0,09	0,72	0,49	0,12	0,74	-0,02	1,00	0,57	0,16	0,22	0,25	-0,21	0,33	0,20	0,46	0,22	SI
UP P05	0,26	0,38	0,31	0,28	0,13	0,26	0,47	0,57	1,00	0,49	0,42	0,38	0,31	0,56	0,52	0,64	0,37	SI
P08	0,32	0,15	0,03	-0,11	0,29	0,43	0,79	0,16	0,49	1,00	0,91	0,71	0,66	0,64	0,58	0,80	0,39	SI
P11	0,33	0,09	0,18	-0,12	0,04	0,60	0,70	0,22	0,42	0,91	1,00	0,74	0,43	0,62	0,70	0,78	0,38	SI
IU P07	0,69	0,31	0,18	-0,12	0,23	0,50	0,82	0,25	0,38	0,71	0,74	1,00	0,57	0,73	0,76	0,77	0,43	SI
P10	0,30	0,13	-0,11	0,07	0,65	-0,11	0,72	-0,21	0,31	0,66	0,43	0,57	1,00	0,38	0,53	0,62	0,26	SI
P13	0,71	0,24	0,29	-0,07	-0,08	0,44	0,83	0,33	0,56	0,64	0,62	0,73	0,38	1,00	0,52	0,66	0,41	SI
P15	0,47	0,08	0,39	0,27	0,20	0,37	0,68	0,20	0,52	0,58	0,70	0,76	0,53	0,52	1,00	0,70	0,41	SI

**Tabla 31.** Análisis de la validez constructiva del cuestionario (sin P2 y P12)

	Facilidad de Uso Percibida						Utilidad Percibida			Intención de uso				Promedio		Válida	
	P1	P3	P4	P6	P9	P14	P5	P8	P11	P7	P10	P13	P15	VC	VD		
FUP	P1	1,00	0,63	0,36	0,28	0,61	0,52	0,26	0,32	0,33	0,69	0,30	0,71	0,47	0,57	0,44	SI
	P03	0,63	1,00	0,79	0,07	0,63	0,72	0,31	0,03	0,18	0,18	-0,11	0,29	0,39	0,64	0,18	SI
	P04	0,36	0,79	1,00	0,48	0,25	0,49	0,28	-0,11	-0,12	-0,12	0,07	-0,07	0,27	0,56	0,03	SI
	P06	0,28	0,07	0,48	1,00	0,03	0,12	0,13	0,29	0,04	0,23	0,65	-0,08	0,20	0,33	0,21	SI
	P09	0,61	0,63	0,25	0,03	1,00	0,74	0,26	0,43	0,60	0,50	-0,11	0,44	0,37	0,54	0,35	SI
UP	P14	0,52	0,72	0,49	0,12	0,74	1,00	0,57	0,16	0,22	0,25	-0,21	0,33	0,20	0,60	0,22	SI
	P05	0,26	0,31	0,28	0,13	0,26	0,57	1,00	0,49	0,42	0,38	0,31	0,56	0,52	0,64	0,36	SI
	P08	0,32	0,03	-0,11	0,29	0,43	0,16	0,49	1,00	0,91	0,71	0,66	0,64	0,58	0,80	0,37	SI
	P11	0,33	0,18	-0,12	0,04	0,60	0,22	0,42	0,91	1,00	0,74	0,43	0,62	0,70	0,78	0,37	SI
IU	P07	0,69	0,18	-0,12	0,23	0,50	0,25	0,38	0,71	0,74	1,00	0,57	0,73	0,76	0,77	0,39	SI
	P10	0,30	-0,11	0,07	0,65	-0,11	-0,21	0,31	0,66	0,43	0,57	1,00	0,38	0,53	0,62	0,22	SI
	P13	0,71	0,29	-0,07	-0,08	0,44	0,33	0,56	0,64	0,62	0,73	0,38	1,00	0,52	0,66	0,38	SI
	P15	0,47	0,39	0,27	0,20	0,37	0,20	0,52	0,58	0,70	0,76	0,53	0,52	1,00	0,70	0,41	SI



**Tabla 32.** Fiabilidad para los constructores de MAM

<i>Constructor</i>	<i>Cronbach (<math>\alpha</math>)</i>	<i><math>\alpha</math> sin P2 y P12</i>
<i>Facilidad de uso percibida</i>	<i>0,71</i>	<i>0,802</i>
<i>Utilidad percibida</i>	<i>0,818</i>	<i>0,818</i>
<i>Intención de uso</i>	<i>0,846</i>	<i>0,846</i>

A continuación el análisis e interpretación de los datos para la prueba de las hipótesis definidas es descrito a continuación.

### 8.3.2. *Análisis e interpretación de resultados*

Los puntajes asignados por cada sujeto fueron promediados sobre las diferentes preguntas del cuestionario que son relevantes para cada constructor (véase Tabla C1 del Apéndice C). De este modo, obtuvimos tres valores promedios para cada sujeto. El análisis descriptivo para las tres variables de interés se presenta en la Tabla 33. Es de destacar que la media más alta obtenida en la replicación del presente estudio correspondió a facilidad de uso percibida. Además, un desvío de sólo 0,59 fue obtenido, por lo que esta media es representativa. El resto de promedios fueron superiores al puntaje medio. Sin embargo, en la Intención de Uso se observa que los datos están ligeramente más dispersos en un mayor rango. Esto es confirmado con los respectivos valores de la desviación estándar.

**Tabla 33.** Valores descriptivos para los constructores del MAM

<b>Estadística</b>	<b>FUP</b>	<b>UP</b>	<b>IU</b>
Media aritmética	3,98	3,67	3,61
Desviación estándar	0,59	0,87	0,89
Mínimo	2,83	2,33	1,75
Máximo	5,00	5,00	5,00

Con el fin de poder generalizar estos resultados, la prueba de las hipótesis H1, H2 y H3 fueron llevadas a cabo. En primer lugar, la prueba de Kolmogorov-Smirnov fue llevada a cabo. Como la distribución de estos datos fue normal, la prueba de t-student fue utilizada para comprobar la diferencia

entre los valores promedio y el valor 3 (puntaje medio). La prueba estadística fue aplicada con un nivel de significación de 5 % ( $\alpha= 0,05$ ). Los resultados obtenidos (ver Tabla 34) permiten el rechazo de las hipótesis nulas. Por consiguiente verificamos empíricamente con un 95% de confiabilidad que RmFFP es percibido como fácil de usar y útil. Además existe la intención de ser usado en un futuro.

**Tabla 34.** Prueba de T-Student para los constructores del MAM

<b>Estadística</b>	<b>FUP</b>	<b>UP</b>	<b>IU</b>
Diferencia de medias	0,9849	0,6667	0,614
95% Intervalo de confianza	1,381 (Sup.) 0,5891 (Inf.)	1,2516 (Sup.) 0,0827 (Inf.)	1,212 (Sup) 0,0157 (Inf)
t	5,545	2,544	2,29
p-valor	0,000	0,029	0,045

#### **8.4. Discusión**

La aplicación del Modelo de Adopción de Métodos para evaluar y predecir la adopción de RmFFP en la práctica fue positiva y cuestionable a la vez.

Fue positiva porque nos permitió evaluar las percepciones de los sujetos al usar RmFFP, confirmando de manera empírica que este procedimiento de medición fue percibido como fácil de usar y útil a la vez. Además, la intencionalidad de ser usado en el futuro pudo ser también verificada con un alto nivel de significancia.

Sin embargo, fue cuestionable porque nos permitió detectar indicios sobre algunas relaciones del MAM que no pudieron ser verificadas. A continuación estas relaciones son explicadas.

*Con respecto a la Utilidad y Facilidad de Uso Percibida.*- A pesar de que existen trabajos donde se verifica la relación entre estos dos constructores [84], [85], en el presente trabajo de investigación se observó que no existe una relación lineal entre ambas variables. Una explicación puede ser que la percepción sobre la utilidad de un procedimiento de medición puede ser determinada por otros factores en el campo de la medición del software. Por ejemplo, Moore y Benbasat afirman que la tangibilidad de los resultados es una influencia directa a la utilidad percibida [86]. Esta tangibilidad en nuestro caso se concretaría por ejemplo en la estimación de otros indicadores en base a las medidas de tamaño obtenidas.

*Con respecto a la Facilidad de Uso Percibida e Intención de Uso.*- Esta relación tampoco pudo ser verificada de manera empírica. Sin embargo, la intención de uso de un procedimiento si fue influenciada por su Utilidad Percibida. Esta evidencia encontrada resulta ser comprensible, ya que un procedimiento tendrá mayor posibilidad de ser usado si este procedimiento es útil en términos de exactitud y fiabilidad para la estimación de otros indicadores (costo, productividad, esfuerzo, etc.). La percepción de su facilidad de uso no sería un factor determinante, más aún si este procedimiento de medición es automatizado. Asimismo, cabe mencionar que la habilidad adquirida con el entrenamiento continuo, podría afectar tanto a la facilidad de uso percibida, como a la utilidad percibida. Existen estudios al respecto, donde la habilidad con el tiempo puede afectar de manera positiva o negativa a la utilidad percibida, ya que el descubrimiento de ventajas o desventajas del procedimiento ocasionarán una percepción favorable o desfavorable de su utilidad [87].



En este capítulo se discute si el objetivo general de la presente tesis ha sido alcanzado por el trabajo de investigación desarrollado. Además se presenta la relación de publicaciones que han sido derivadas del presente trabajo de investigación. Finalmente una serie de trabajos futuros también son presentados.

### ***9.1. Conclusiones***

Las conclusiones de esta tesis son presentadas a continuación:

**Primero:** Un procedimiento de medición fue diseñado para medir el tamaño funcional de las aplicaciones generadas con OO-Method a partir de una especificación de requisitos. Este procedimiento incluye la definición de un conjunto de reglas de representación que permiten la obtención de un “modelo de software”, abstracción de las primitivas del modelo de requisitos de OO-Method que contribuyen al tamaño funcional, según el método estándar COSMIC-FFP. Así mismo, con el fin de contribuir a la correcta construcción de dicho modelo de software a medir, se definió otro conjunto de reglas que permiten la eliminación de movimientos de datos duplicados. Además, un conjunto de reglas adicionales fueron definidas para obtener un tamaño funcional a nivel de caso de uso y de todo el sistema.

**Segundo:** Para facilitar la aplicación de RmFFP, se definió una secuencia de pasos para permitir la definición del contexto de medición, la construcción del modelo de software y su cuantificación para la obtención del tamaño funcional.

**Tercero:** Se definió el alcance de la validación del diseño de un procedimiento de medición que es comúnmente confundido como validación teórica. Por consiguiente, el diseño de RmFFP fue validado en base a 1) su conformidad con el método estándar ISO/IEC 19761, para esto se adaptó la parte 2 del estándar ISO/IEC 14143; 2) que la mayoría de conceptos del modelo de metrología fueron contemplados por RmFFP; 3) que fue validado teóricamente de acuerdo al marco formal DISTANCE.

**Cuarto:** La evaluación de la calidad de los resultados de medición fue llevada a cabo en términos de su exactitud y precisión. Así mismo, un estudio sobre la productividad de los usuarios cuando usan RmFFP fue también llevado a cabo. En cuanto a la evaluación de su exactitud, al comparar las mediciones llevadas a cabo sobre una especificación de requisitos de referencia, seleccionada de la ISO 14143-4 [19], se observó que la diferencia entre el valor de referencia obtenido con COSMIC-FFP y el valor obtenido con RmFFP era mínima (1 cfsu). Sin embargo, a pesar de observar esta evidencia es necesario tener más valores de referencia sobre diferentes especificaciones de requisitos para poder mostrar con mayor evidencia la exactitud de RmFFP. En cuanto a la evaluación de su precisión, en términos de su reproducibilidad, se demostró que RmFFP es un procedimiento de medición reproducible. En cuanto a la evaluación de la productividad de los usuarios -cuando estos miden manualmente con RmFFP-, se observó que esta productividad es mayor que cuando es medida con COSMIC-FFP, según estudios reportados por Nagano, ó con IFPUG FPA según Total Metrics. Sin

embargo, a pesar de que la productividad de los usuarios, usando un procedimiento, es mayor que cuando usan un método de medición, esta conclusión no puede ser generalizada dado que existen factores externos que pueden estar afectando a los resultados obtenidos, tales como: la complejidad del caso de estudio, la experiencia de las personas que miden, etc.

**Quinto:** Un estudio empírico fue llevado a cabo para evaluar la intención de adoptar RmFFP en la práctica en base a las percepciones de utilidad y facilidad de uso que tienen los usuarios al medir con RmFFP. Este estudio fue replicado con el fin de aumentar la fiabilidad de los resultados obtenidos, los cuales demostraron empíricamente que si hay una intención a usar RmFFP, puesto que este procedimiento de medición es percibido como fácil de usar y útil.

## **9.2. Publicaciones relacionadas**

Las contribuciones de esta tesis están respaldadas por las siguientes publicaciones obtenidas a lo largo del presente trabajo desarrollado.

### **9.2.1. Revistas internacionales**

- **Condori-Fernández N.**, Abrahão S., and Pastor O., On the Estimation of Software Functional Size from Requirements Specifications. *Journal of Computer Science and Technology (JCST)*, Special Issue on Advances in Software Metrics and Software Processes, Springer, May-2007.

### **9.2.2. Revistas iberoamericanas y nacionales**

- **Condori-Fernández N.**, Pastor O., Un Análisis sobre la Adopción en la Industria de un Procedimiento de Medición conforme a la ISO/IEC 19761, *Revista de Gerencia Tecnológica*

Informática, Instituto Tecnológico Ibero-americano de Informática de Colombia, diciembre 2006.

- **Condori-Fernández N.**, Pastor O. Re-Evaluando la Intención de Uso de un Procedimiento de Medición basado en COSMIC-FFP. (Artículo invitado) 2006.
- **Condori-Fernández N.**, Abrahão S., Pastor O., y Martí S. Un Procedimiento de Medición de Tamaño Funcional: Diseño y Aplicación, Revista de Procesos y Métricas, Asociación Española de Métricas del Software, Número 3, Diciembre 2004.

### 9.2.3. *Conferencias internacionales*

- **Condori-Fernández N.**, Pastor O., Analyzing the Applicability of a Theoretical Model in the Evaluation of Functional Size Measurement Procedures. 19th International Conference on Software Engineering and Knowledge Engineering. Quality Software (SEKE 2007), Boston, USA, June 2007 (accepted for publication).
- **Condori-Fernández N.**, Pastor O., An Empirical Study on the Likelihood of Adoption in Practice of a Size Measurement Procedure for Requirements Specification 6th IEEE International Conference on Quality Software (QSIC 2006), Beijing, China, October 2006.
- **Condori-Fernández N.**, Pastor O. Re-Assessing the Intention to Use a Measurement Procedure based on COSMIC-FFP, International Conference on Software Process and Product Measurement, Cadiz, Spain, November 2006.



- **Condori-Fernández N.**, Abrahão S., and Pastor O., Towards a Functional Size Measure for Object-Oriented Systems from Requirements Specifications, 4th IEEE International Conference on Quality Software (QSIC 2004), Braunschweig-Germany, September 2004.

#### 9.2.4. *Workshops Internacionales e Iberoamericanas*

- **Condori-Fernández N.**, **Pastor O.**, Evaluating the Productivity and Reproducibility of a Measurement Procedure, ER International Workshop on Quality of Information Systems, Springer LNCS, (QoIS), Tucson-Arizona, USA, November 2006.
- **Condori-Fernández N.**, **Pastor O.**, Verifying the Construction of a Software Model from a Requirements Model, IX Workshop on Requirements Engineering (WER 2006), Río de Janeiro, Brazil, July 2006.
- **Condori-Fernández N.**, Abrahão S., Pastor O. El Problema de la Duplicidad de Movimientos de Datos en un Procedimiento de Medición. 9° Ibero-american Workshop on Requirements Engineering and Software Environments (IDEAS'2006), La Plata, Argentina, April 2006.
- **Condori-Fernández N.**, Abrahão S., Pastor O., Validación del Diseño de un Procedimiento de Medición de Tamaño Funcional, 8° Ibero-american Workshop on Requirements Engineering and Software Environments (IDEAS'2005), Valparaíso-Chile, May 2005.
- **Condori-Fernández N.**, Abrahão S., and Pastor O., Una Experiencia en la Validación Teórica de Métricas de Software, 7°

Ibero-american Workshop on Requirements Engineering and Software Environments (IDEAS'2004), Arequipa, Perú, May 2004.

#### **9.2.5. Conferencias Nacionales**

- **Condori-Fernández N.**, Pastor O. Abrahão S., Martí S., Procedimiento de Medición del Tamaño Funcional para Aplicaciones Software Basado en un Modelo de Requisitos, V Congreso Nacional de Procesos y Métricas de Sistemas de Información, Madrid, 17-18 Noviembre 2004.

### **9.3. Trabajos futuros**

Los resultados de I+D obtenidos en el ámbito de este trabajo de Tesis van a tener continuidad en un conjunto de líneas específicas de mejora y enriquecimiento que están ya siendo abordadas. Concretamente, los siguientes trabajos futuros han sido derivados a partir del trabajo realizado en la presente tesis.

- Estudio sobre el factor de convertibilidad a otras unidades de medición de tamaño funcional.
- Estudio sobre técnicas de determinación de valores de incertidumbre para el campo de la medición del software, con el fin de facilitar una mejor evaluación de la exactitud de las mediciones llevadas a cabo.
- Estudio experimental sobre la exactitud de RmFFP, adaptando el estándar ISO 5725.
- Replicación del estudio experimental -con profesionales de la industria- para analizar si las percepciones e intenciones son realmente un resultado del rendimiento usando RmFFP.

- Análisis sobre la validez del Modelo de Evaluación de Métodos para la evaluación de procedimientos de medición de tamaño funcional.
- Implementación y evaluación de un instrumento de medición, mediante la automatización del procedimiento de medición propuesto, para ser incorporado en la herramienta RETO y OLIVANOVA.

Una de las líneas de investigación más prometedoras es la de llevar las técnicas derivadas de COSMIC-FFP a todos los ámbitos de medición de la funcionalidad asociada a entornos de producción de software basados en OO-Method y soportados por la herramienta comercial OlivaNova. Demostrada su aplicación en el ámbito del Modelado de Requisitos, el paso siguiente es aplicarla al nivel de Modelado Conceptual, y compararla con propuestas existentes en la actualidad basadas en IFPUG FPA, con el objetivo de mejorar las prestaciones actuales del ámbito metodológico proporcionado por OO-Method.



# Referencias

- [1] ISO/IEC, International Vocabulary of Basic and General Terms in Metrology (VIM), International Organization for Standardization, Geneva, Switzerland, 1993.
- [2] BIPM: <http://www.bipm.org/> última visita 31 de Julio de 2006
- [3] OIML: <http://www.oiml.org> última visita 1 de Agosto de 2006.
- [4] Carnahan L., Carver G., Gray M., Hogan M., Hopp T., Horlick J., Lyon G., and Messina E., Metrology for Information Technology, Standard View, 5(3):103-109, 1997.
- [5] Abran A., Software Metrics Need to Mature into Software Metrology (Recommendations)", presented in the NIST Workshop on Advancing Measurements and Testing for Information Technology (IT), Maryland, USA, 1998.
- [6] Peng W., Wallace D., Software Error Analysis, NIST Special Publication, National Institute of Standards and Technology, Gaithersburg, MD, March 1993.
- [7] Brito F. and Carapua R., Candidate metrics for object oriented software within a taxonomy framework. Journal of Systems and Software, 26(1):87-96, July 1994.
- [8] Woodings, T L, A Taxonomy of Software Metrics, Software Process Improvement Network (SPIN), available from Comast Consulting, Perth, 1995.
- [9] Mendes E., Cuncell S., Mosley N., Towards a Taxonomy of Hypermedia and Web Application Size Metrics, International Conference on Web Engineering, Springer LNCS, Sydney, Australia, July , 2005, pp. 110-123.
- [10] Abran A., Sellami A., Suryn W., Metrology, Measurement and Metrics in Software Engineering, Proceedings of the Ninth International Software Metrics Symposium, Sydney, Australia, 2003.
- [11] Abran A. and Sellami A., Initial Modeling of the Measurement Concepts in the ISO Vocabulary of Terms in Metrology, in Software Measurement and Estimation-Proceedings of the 12th International Workshop on Software Measurement, Magdeburg-Germany, October 2002, Shaker Verlag, pp. 315.
- [12] Gray M., Applicability of Metrology to Information Technology, Journal of Research of the National Institute of Standards and Technology, 4(3):103-109, 1999.

- [13] Abran A. Sellami A., Analysis of Software Measures Using Metrology Concepts – ISO 19761 Case Study, International Workshop on Software Audits and Metrics, Porto, Portugal 2004.
- [14] Abran A., E. Al-Quatish, J. Cuadrado, Investigation of the Metrology Concepts in ISO 9126 on Software Product Quality Evaluation, In Proceedings of the 10th International Conference on Computers, Athens-Greece, July 2006, pp. 864-872.
- [15] ISO/IEC, ISO/IEC TR 9126-4: Software Engineering – Product Quality – Part 4: Quality in Use Metrics, International Organization for Standardization, Geneva, Switzerland, 2004.
- [16] ISO, ISO/IEC 14143-1- Information Technology - Software measurement-Functional Size Measurement. Part 1: Definition of Concepts, International Organization for Standardization, Geneva, Switzerland, 1998.
- [17] ISO, "ISO/IEC 14143-2 - Information Technology - Software measurement -Functional Size Measurement. Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1:1998", International Organization for Standardization, Geneva, Switzerland, 2002.
- [18] ISO, "ISO/IEC TR 14143-3 - Information technology - Software measurement --Functional size measurement - Part 3: Verification of functional size measurement methods", International Organization for Standardization, Geneva, Switzerland, 2003.
- [19] ISO, "ISO/IEC TR 14143-4: Information technology-Software measurement-Functional size measurement - Part 4: Reference model", International Organization for Standardization, Geneva, Switzerland, 2002.
- [20] ISO, "ISO/IEC TR 14143-5:2004, Information technology – Software measurement -- Functional size measurement - Part 5: Determination of functional domains for use with functional size measurement", International Organization for Standardization, Geneva, Switzerland, 2004.
- [21] ISO, "ISO/IEC 14143-6:2006, Information technology – Software measurement -- Functional size measurement - Part 6: Guide for use of ISO/IEC 14143 series and related International Standards", International Organization for Standardization, Geneva, Switzerland, 2006.
- [22] ISO, "ISO/IEC 19761: 2003, Software Engineering: COSMIC-FFP-A Functional Size Measurement Method", International Organization for Standardization-ISO, Geneva, 2003.
- [23] ISO, ISO/IEC 15939:2002, Software Engineering – Software Measurement Process, International Organization for Standardization, Geneva, Switzerland, 2002.
- [24] Abran A., J. M. Desharnais, S. Oigny, D. St-Pierre, and C. Symons, "COSMIC-FFP Measurement Manual – Version 2.2, The COSMIC Implementation Guide for ISO/IEC 19761:2003" École de technologie

supérieure- ETS, Montreal (Canada) 2003. Available free at: [www.lrgl.uqam.ca/cosmic-ffp](http://www.lrgl.uqam.ca/cosmic-ffp)

[25] ISO, ISO/IEC 20926:2003, Software engineering - IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual, International Organization for Standardization, Geneva, 2003.

[26] ISO, ISO/IEC 20968: 2002, Software engineering - Mk II Function Point Analysis - Counting Practices Manual, International Organization for Standardization, Geneva, 2002.

[27] ISO, ISO/IEC 24570: 2004, Software engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization, Geneva, 2004.

[28] ISO, ISO 5725-1:1994, Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions, International Organization for Standardization, Geneva, 1994.

[29] ISO, ISO 5725-2:1994, Accuracy (trueness and precision) of measurement methods and results - Part 2: Basic method for the determination of repeatability and reproducibility of a standard measurement method, International Organization for Standardization, Geneva, 1994.

[30] ISO, ISO 5725-3:1994, Accuracy (trueness and precision) of measurement methods and results - Part 3: Intermediate measures of the precision of a standard measurement method, International Organization for Standardization, Geneva, 1994.

[31] ISO, ISO 5725-4:1994, Accuracy (trueness and precision) of measurement methods and results - Part 4: Basic methods for the determination of the trueness of a standard measurement method, International Organization for Standardization, Geneva, 1994.

[32] ISO, ISO 5725-5:1998, Accuracy (trueness and precision) of measurement methods and results - Part 5: Alternative methods for the determination of the precision of a standard measurement method, International Organization for Standardization, Geneva, 1998.

[33] ISO, ISO 5725-6:1994, Accuracy (trueness and precision) of measurement methods and results - Part 6: Use in practice of accuracy values, International Organization for Standardization, Geneva, 1994.

[34] Jacquet J. P. and Abran A., "From Software Metrics to Software Measurement Methods: A Process Model", in International Software Engineering Standards Symposium and Forum, ISESS 97: IEEE-Computer Society Press, 1997, pp. 128-135.

[35] Conte, S.D., Dunsmore, H.E., "Software Engineering Metrics and Models", Benjamin/Cummings, Menlo Park, CA, 1986.

- [36] Tran-Cao D., Levesque G., Abran A., "Measuring Software Functional Size: Towards an Effective Measurement of Complexity", Proceedings of the International Conference on Software Maintenance, Montreal-Canada, October 2002.
- [37] Albrecht A.J., "Measuring application development productivity". In IBM Application Development Symposium, 1979, pp. 83-92.
- [38] Jones C., "The SPR Feature Point Method", Software Productivity Research Inc., 1986.
- [39] Whitmire S.A., "3-D Function Points: Scientific and Real-Time Extensions to Function Points", Proceedings of the Pacific Northwest Software Quality Conference, Portland, 1992.
- [40] St-Pierre, D., Maya, M., Abran, A., Desharnais, J.-M., and Bourque, P., "Full Function Points: Function Points Extension for Real-Time Software - Counting Practices Manual", Université du Québec à Montréal, Montréal, Technical Report no. 1997-04.
- [41] NESMA, "Definitions and Counting Guidelines for the Application of Function Point Analysis. Nov 1997, (Disponible en [www.nesma.org](http://www.nesma.org) ).
- [42] UKSMA, "MKII Function Point Analysis Counting Practices Manual. Version 1.3.1", United Kingdom Software Metrics Association 1998.
- [43] IFPUG, "Function Point Counting Practices Manual, Release 4.1", International Function Point Users Group, Westerville, Ohio, USA 1999.
- [44] Abran A., Desharnais J.-M, Oligny S., St-Pierre D., Symons C., "COSMIC-FFP Measurement Manual, version 2.0", Ed. S. Oligny, Software Engineering Management, Research Laboratory, Université du Québec à Montréal, Montreal, Canada, 1999.
- [45] Abran A., Desharnais J.-M, Oligny S., St-Pierre D., Symons C., "COSMIC-FFP Measurement Manual, Version 2.1", The Common Software Measurement International Consortium, May 2001.
- [46] Lothar M., Dumke R., "Point Metrics-Comparison and Analysis", Comparison and Analysis. In: Dumke/Abran: Current Trends in Software Measurement, Shaker Publ., Aachen, Germany, 2001, pp. 228-267.
- [47] Fetcke T., Abran A., and Nguyen T. Mapping the OO-Jacobson approach to function point analysis. In Proceedings. TOOLS Conference, Santa Barbara, USA, 28 July- 1 August 1997, pp. 1-12.
- [48] Tavares H., Carvalho A., Castro J. Function Points Measurement from Requirement Specification. In Proceedings. 5th Workshop Engineering Requirements, Valencia-Spain, November 11-12, 2002, pp. 278-298.
- [49] Bertolami M., Oliveros A. Functionality Measurement Process on the Requirements Elicitation, In Proceedings. Workshop Ibero American



Requirements Engineering and Software Environment, Arequipa-Perú, May 3-7, 2004.

[50] IFPUG, "Function Point Counting Practices Manual, Release 4.0", International Function Point Users Group, Westerville, Ohio 1994.

[51] Jacobson I., Christerson M., Johnson P., and Overgaard G., Object-Oriented Software Engineering: A Use-Case Driven Approach, Addison Wesley Longman, Inc., 1992.

[52] J. Leite, J. Hadad, G. Doorn, J. Kaplan, "A Scenario Construction Process", Requirements Engineering Journal, Springer Verlag, 5(1):38-61, 2000.

[53] Abrahão S., PhD Thesis: "On The Functional Size Measurement of Object-Oriented Conceptual Schemas: Design and Evaluation Issues", Valencian University of Technology, Spain, July 2004.

[54] Pastor O., Gomez J., Insfran E., Pelechano V. The OO-Method approach for information systems modelling: from object-oriented conceptual modelling to automated programming. Information Systems, 2001, 26(7):507-534.

[55] Uemura T., Kusumoto S., and Inoue K., "Function Point Measurement Tool for UML Design Specification", 5th International Software Metrics Symposium (METRICS'99), Florida, USA, 1999.

[56] Lehne A., "Experience Report: Function Points Counting of Object Oriented Analysis and Design based on the OOram method", Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'97), 1997.

[57] Reenskaug T., Wold O., and Lehne A., Working with Objects: the OOram Software Engineering Method. England, 1996.

[58] Bévo V., Lévesque G., and Abran A. UML Notation for Functional Size Measurement Method. In Proc. 9th International Workshop on Software Measurement, Canada, September 8-10, 1999, pp. 230-242.

[59] Jenner M.S. COSMIC-FFP and UML: Estimation of the Size of a System Specified in UML-Problems of Granularity. In Proc. Fourth European Conf. Soft. Measurement and ICT Control, Germany, May 2001, pp. 173-184.

[60] Azzouz S., Abran A. A Proposed Measurement Role in the Rational Unified Process and its Implementation with ISO 19761: COSMIC-FFP. In Proc. Software Measurement European Forum, Rome, Italy, January 28-30, 2004.

[61] Habela P., Glowacki E., Serafinski T. Adapting Use Case Model for COSMIC-FFP based Measurement. In Proc. 15th International Workshop on

- Software Measurement, Montreal, Canada, Shaker-Verlag, September 12-14, 2005.
- [62] Poels G. Functional Size Measurement of Multi-Layer Object-Oriented Conceptual Models. In Proc. 9th International Object-Oriented Information Systems Conference, Geneva, Switzerland, September 2-5, 2003, pp. 334-345.
- [63] Diab H., Koukane F., Frappier M., St-Denis R.  $\mu$ ROSE: Automated Measurement of COS-MIC-FFP for Rational Rose Real Time. *Information and Software Technology*, 2005, 47(3): 151-166.
- [64] Nagano S., Ajisaka T., Functional metrics using COSMIC-FFP for object-oriented real-time systems. In Proc. 13th International Workshop on Software Measurement, Montreal, Canada, September 23-25, 2003.
- [65] Object Management Group - OMG. Unified Modelling Language-UML v 1.5, March 2003. <http://www.omg.org/uml>.
- [66] Jenner M.S., "Automation of Counting of Functional Size Using COSMIC-FFP in UML," Proceedings. 12th Int'l Workshop Soft. Measurement, pp. 43-51, 2002.
- [67] Dedene G. and Snoeck M. M.E.R.O.DE.: A Model-driven Entity-Relationship Object-oriented Development Method. *ACM SIGSOFT Software Engineering Notes*, 1994, 19(3): 51-61.
- [68] Poels G., Definition and Validation of a COSMIC-FFP Functional Size Measure for Object-Oriented Systems. In: Proceedings. 7th Int. ECOOP Workshop Quantitative Approaches OO Software Eng. Darmstadt, 2003.
- [69] Poels G., Dedene G.: Distance-based software measurement: necessary and sufficient properties for software measures. *Information and Software Technology*, 2000, 42(1): 35-46.
- [70] Selic B. Real-Time Object-Oriented Modeling (ROOM). In Proc. 2nd IEEE Real-Time Technology and Applications Symposium, Boston, USA, June 10-12, 1996, pp. 214.
- [71] Shlaer S., & Mellor S., *Object Lifecycles, Modelling the World in States*, Yourdon Press, Prentice-Hall, 1992.
- [72] Mellor S. Balcer J., *Executable UML: A Foundation for Model-Driven Architecture*, Addison Wesley, 2002.
- [73] Bevo V., *Analyse et Formalisation Ontologique des Procédures de Mesure Associées aux Méthodes de Mesure de la Taille Fonctionnelle des Logiciels: de Nouvelles Perspectives Pour la Mesure*, Université du Québec à Montréal - UQAM, Montréal, Québec, PhD Thesis, April 2005.
- [74] Uemura T., Kusumoto O., Katsuro I., Function-point analysis using design specifications based on the unified modelling language, *Journal of*

Software Maintenance and Evolution: Research and Practice, John Wiley & Sons, Ltd., 2001, 13 (1): 223-243.

[75] Moody D. L., "Dealing with Complexity: A Practical Method for Representing Large Entity Relationship Models", PhD. Thesis, Department of Information Systems, University of Melbourne, Australia, 2001.

[76] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology", MIS Quarterly, 1989, pp. 319-340.

[77] Insfrán E., Pastor O. and Wieringa R., 2002. Requirements Engineering-Based Conceptual Modelling. Journal Requirements Engineering (RE), Springer-Verlag, pp. 61-72.

[78] Juristo N. and Moreno A.M<sup>a</sup>. Basics of Software Engineering Experimentation, Kluwer Academic Publishers, 2001.

[79] Wohlin C., Runeson P., Höst M., M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in Software Engineering: An Introduction, 2000.

[80] Basili V. R. and Rombach H. D., 1988. The TAME Project: Towards Improvement-Oriented Software Environments, IEEE Transactions on Software Engineering, pp. 758-773.

[81] Insfran E., 2003. A Requirements Engineering Approach for Object-Oriented Conceptual Modeling, PhD Thesis, Valencia Polytechnic University.

[82] Campbell D. T. and Fiske D. W., "Convergent and Discriminant Validation by the Multitrait-Multimethod Matrix", in Psychological Bulletin, vol. 56, 1959, pp. 81-105.

[83] Nunally J., Psychometric Theory, McGraw-Hill, 2nd ed., New York, NY1978.

[84] Neville K., Fitzgerald B., An Innovative Training for an Organization Embracing Technology, Journal of Information Technology Education, 2002, 1(3):193-200.

[85] Davis, F., Bagozzi, R. and Warshaw, P. User acceptance of computer technology: comparison of two theoretical models, Management Science, 1989, 13(3): 982-1003.

[86] Moore, G. C., Benbasat, I. Development of an instrument to measure the perceptions of adopting an information technology innovation. Journal Information Systems Research, 1991, 2(3):192-222.

[87] Venkatesh, V. and Davis, F. D., Theoretical extension of the technology acceptance model: Four longitudinal field studies, Journal Management Science, February 2000, 46(2):186-204.

[88] Sellami A., Abran A., The contribution of metrology concepts to understanding and clarifying a proposed framework for software

- measurement validation, in International Workshop on Software Measurement, Montreal, Shaker-Verlag, 2003, pp.18-40.
- [89] Whitmire, S.: Object Oriented Design Measurement. Wiley Computer Publishing, New York (1997).
- [90] Baker, A.L.; Bieman, J.M.; Fenton, N.; Gustavson, D.A.; Melton, A.; Whitty, R.: A Philosophy for Software Measurement. The Journal of Systems and Software, 199012 (3): 277-281.
- [91] Zuse, H.: A Framework for Software Measurement, Walter de Gruyter, Berlin, (1998).
- [92] Fetcke T., Abran A., and Dumke R., "A Generalized Representation for Selected Functional Size Measurement Methods", 11th International Workshop on Software Measurement, Montréal, Canada, 2001.
- [93] Abrahão S., Poels G., Pastor O., "Validation Issues in Functional Size Measurement of Object-Oriented Conceptual Schemas: The Case of OOmFP", 8th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, Oslo-Norway, June 2004.
- [94] COSMIC: <http://www.cosmicon.com/> última visita 15 de septiembre de 2006.
- [95] Kitchenham, B.: Towards a Framework for Software Measurement Validation, IEEE Transactions of Software Engineering, 1995, 21(12): 929-944.
- [96] N. Fenton "Software Metrics: A Rigorous Approach", Chapman & Hall, 1991.
- [97] Fetcke T, Abran A., and Dumke R., "A Generalized Representation for Selected Functional Size Measurement Methods", 11th International Workshop on Software Measurement, Montréal, Canada, 2001, pp 1-12.
- [98] Pastor O., Abrahão S., and Fons J., "An Object-Oriented Approach to Automate Web Applications Development", 2nd International Conference on Electronic Commerce and Web Technologies (EC-Web'01), Munich, Germany, 2001.
- [99] Suppes, P. Krantz D. M., Luce, R. D., and Tversky, A. Foundations of Measurement: Geometrical, Threshold, and Probabilistic Representations, Vol. 2, Academic Press, San Diego, California, 1989, pp 493.
- [100] Desharnais J.-M., Abran A., Applying a Functional Measurement Method: Cognitive Issues, International Workshop on Software Measurement, Montréal-Canada, Shaker-Verlag, 2001, pp.13-25.
- [101] Pendrill L., Swedish National Testing and Research Institute: <http://www.sp.se/Metrology/eng/terminology.htm>, last visit October 3, 2006.

- [102] DOE Handbook, Alternative Systematic Approaches to Training, 1074-95, January 1995.
- [103] Kemerer C. F. Reliability of Function Points Measurement: A Field Experiment. *Communications of the ACM*, February 1993, 36(2): 85-97
- [104] Software Functional Size with ISO 19761: 2003, Case Study: Valve Control System, Université du Québec, École de Technologie Supérieure, January 2006, (available at <http://www.lrgl.uqam.ca/cosmic-ffp/>).
- [105] ISO: Guide to the Expression of Uncertainty in Measurement, First edition 1995, (available at <http://www.gum.dk/>).
- [106] Diaz I., Moreno L., Pastor O., Matteo A., Interaction Transformation Patterns Based on Semantic Roles, International Conference on Applications of Natural Language to Information Systems, Alicante-Spain, Springer, June 2005, pp. 239-250.
- [107] IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.
- [108] Object Management Group (OMG), última visita 27 de Noviembre de 2006, <http://www.omg.org/mda>.
- [109] Molina J.C., Pastor O., MDA, OO-Method y la Tecnología OLIVANOVA Model Execution, Actas del I Taller sobre el desarrollo dirigido por modelos, MDA y Aplicaciones, Universidad de Málaga, Noviembre 2004, Málaga-España, pp. 78-87.
- [110] Diaz I., Sanchez J. y Pastor O., Metamorfosis: Un marco para el análisis de requisitos funcionales, Workshop on Requirements Engineering (WER), Porto-Portugal, 2005, pp. 233-244.
- [111] Molina, P. J. Meliá, S. and Pastor, O. (2002), Just-UI: A User Interface Specification Model. In Ch. Kolski and J. Vanderdonckt (Eds.), *Computer-Aided Design of User Inter-faces III*, Kluwer Academics Publisher, pp. 63–74.
- [112] Molina P., Especificación de interfaz de usuario: De los requisitos a la generación automática, PhD Thesis, Universidad Politécnica de Valencia-DSIC, Marzo de 2003.
- [113] Pastor O., Hayes F., Bear S., OASIS: An Object Oriented specification language, *Procs in CAISE*, Springer-Verlag, Berlin-Germany, 1992, pp. 348-363.
- [114] Gómez J., Insfrán E., Pelechano V. and Pastor O., The Execution Model: a component-based architecture to generate software components from conceptual models, Workshop on Component-based Information Systems Engineering, Pisa-Italia, June 1998.
- [115] Wieringa R., Postmodern Software Design with NYAM: not yet another Method, International Workshop on Requirements Targeting

- Software and Systems Engineering, Springer, Bernried-Germany, 1997, pp. 69-94.
- [116] Wirfs-Brock R. Designing Scenarios: Making the Case for a Use Case Framework, Smalltalk Report, Nov-Dec, 1993: SIGS Publications.
- [117] Leite J.C.S., Hadad G., Doorn J. and Kaplan G., A Scenario Construction Process. Requirements Engineering Journal, 2000, 5(1): p. 38-61.
- [118] Diaz I., Pastor O., Modeling Interactions using Role-Driven Patterns, 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society, 2005, Paris, France, pp. 209-220.
- [119] CARE Technologies, última visita 30 de Noviembre 2006, <http://www.care-t.com>
- [120] CARE Technologies S.A, "Software Engineering Process", 2nd Scientific Advisory Borrada Meeting, Denia-Spain, November 2003, <http://www.care-t.com>.
- [121] Insfran E., Requirements Tools (RETO), última visita setiembre 2006, <http://reto.dsic.upv.es/reto/>.
- [122] Paulk C., Weber V., Curtis B., Chrissis B., The Capability Maturity Model: Guidelines for improving the software process, Addison Wesley Professional, June 1995, pp. 464.
- [123] Condori-Fernández N., Modelo de Evaluación de Especificación de Requisitos de Software basado en un sistema difuso, 5º Workshop Ibero-Americano sobre Ingeniería de Requisitos y Ambientes de Software, IDEAS, Universidad de la Habana, La Habana-Cuba, Abril 2002.
- [124] Total Metrics, "Levels of Counting", Australia, August 2001.
- [125] Nagano, S.; Mase, K., Watanabe Y., Watahiki T., Nishiyama S., Validation of Application Results of COSMIC-FFP, in Australian Software Conference on Measurement (ASCOM), Australia, 2001.
- [126] Moody D., The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods, Proceedings of the 11th European Conference on Information Systems, ECIS 2003, Naples, Italy 16-21 June 2003.
- [127] Garcia F., Bertoa M., Calero C., Vallecillo A., Ruiz F., Piattini M., Genero M.: Towards a consistent terminology for software measurement. Information & Software Technology 48(8): 631-644 (2006).
- [128] N. Condori-Fernández, S. Abrahao, O. Pastor. Reporte Técnico: "Validación Teórica de Métodos de Medición de Tamaño Funcional, Departamento de Sistemas Informáticos y Computación", Universidad Politécnica de Valencia, 2004.



# Lista de Figuras

<i>Figura 1. Evolución de los métodos estándares.....</i>	<i>- 32 -</i>
<i>Figura 2. Proceso de desarrollo de OO-Method .....</i>	<i>57</i>
<b>Figura 3. Ingeniería de Requisitos de OO-Method .....</b>	<b>60</b>
<i>Figura 4: Relaciones entre casos de uso [81].....</i>	<i>63</i>
<i>Figura 5. Plantilla de descripción de caso de uso [81]......</i>	<i>64</i>
<i>Figura 6. Estructura y notación de un diagrama de secuencia [81] .....</i>	<i>65</i>
<i>Figura 7. Metamodelo del Modelo de Requisitos de OO-Method.....</i>	<i>69</i>
<i>Figura 8. Proceso de desarrollo con ONME .....</i>	<i>71</i>
<i>Figura 9. Metamodelo de COSMIC-FFP.....</i>	<i>76</i>
<i>Figura 10. Correspondencia entre procesos funcionales y casos de uso</i>	<i>79</i>
<i>Figura 11. Correspondencia entre grupo de datos y clases.....</i>	<i>80</i>
<i>Figura 12. Correspondencia entre tipos de movimientos de datos y tipos de mensajes de interacción .....</i>	<i>81</i>
<b>Figura 13. Relación entre movimiento de dato y grupo de dato.....</b>	<b>82</b>
<i>Figura 14. Fragmento de interacción tipo I.....</i>	<i>83</i>
<i>Figura 15. Fragmento de interacción tipo II .....</i>	<i>83</i>
<i>Figura 16. Identificación de movimientos de dato tipo entrada .....</i>	<i>84</i>
<i>Figura 17. Ejemplo de duplicidad de movimientos de datos .....</i>	<i>87</i>
<i>Figura 18. Ejemplo de ocurrencia de movimientos de datos .....</i>	<i>88</i>
<i>Figura 19. Ejemplo de ocurrencia, relación incluye .....</i>	<i>89</i>
<i>Figura 20. Aplicación de un procedimiento de medición .....</i>	<i>96</i>
<i>Figura 21. Proceso general de la aplicación de RmFFP.....</i>	<i>97</i>
<i>Figura 22. Modos de documentación de requisitos funcionales [24] .....</i>	<i>98</i>
<i>Figura 23. Proceso del procedimiento de medición RmFFP.....</i>	<i>101</i>
<i>Figura 24. Diagrama de casos de uso general .....</i>	<i>106</i>
<i>Figura 25. Sub grupo funcional: Gestión de Operaciones .....</i>	<i>106</i>
<i>Figura 26. Diagrama de secuencia para “Crear operación” .....</i>	<i>108</i>
<i>Figura 27. Tamaño funcional de los casos de uso Gestión de vehículos</i>	<i>112</i>



<i>Figura 28. Alcance de la validación del diseño .....</i>	<i>120</i>
<i>Figura 29. Proceso de análisis para cada pregunta, Adaptado de [17].</i>	<i>122</i>
<i>Figura 30. Modelo de proceso de las categorías de metrología [11]...</i>	<i>125</i>
<i>Figura 31. Meta-modelo de los conceptos de alto nivel del VIM.....</i>	<i>126</i>
<i>Figura 32. Topología de alto nivel de la categoría “cantidades y unidades” [11]. .....</i>	<i>128</i>
<i>Figura 33. Aplicación del fundamento de la medición .....</i>	<i>133</i>
<i>Figura 34. Validación teórica de RmFFP usando DISTANCE.....</i>	<i>143</i>
<b><i>Figura 35. Analogía entre exactitud y precisión .....</i></b>	<b><i>152</i></b>
<i>Figura 36. Alcance de la evaluación de la aplicación de RmFFP.....</i>	<i>153</i>
<i>Figura 37. Sub-conceptos de la categoría “Medición” [11].....</i>	<i>154</i>
<i>Figura 38. Estrategia para evaluar la exactitud, adaptado de [19].....</i>	<i>158</i>
<i>Figura 39. Diagrama de casos de uso de SCV.....</i>	<i>160</i>
<i>Figura 40. Diagrama de secuencia: Control de tiempo bajo condiciones normales.....</i>	<i>- 162 -</i>
<i>Figura 41. Diagrama de secuencia: Control de tiempo en cambio de marcha .....</i>	<i>- 163 -</i>
<i>Figura 42. Productividad de medición para sujeto usando RmFFP.-</i>	<i>175 -</i>
<i>Figura 43. Modelo de Adopción de métodos [75].....</i>	<i>- 181 -</i>
<i>Figura 44. Proceso de evaluación de adopción en la práctica de RmFFP-</i>	<i>182 -</i>
<i>Figura 45. Intención de Uso vs. Utilidad y Facilidad de Uso Percibida-</i>	<i>196 -</i>
<b><i>Figura 46. Operacionalización el Modelo MAM.....</i></b>	<b><i>- 205 -</i></b>

## Listado de Tablas

<i>Tabla 1. Estándares de medición</i> .....	- 27 -
<i>Tabla 2. Procedimientos de medición de tamaño funcional</i> .....	52
<i>Tabla 3. Propiedades del mensaje tipo consulta [81]</i> .....	68
<i>Tabla 4. Propiedades del mensaje tipo conexión [81]</i> .....	68
<i>Tabla 5. Medición del Sistema de Alquiler de vehículos</i> .....	113
<i>Tabla 6. Respuestas de la evaluación de conformidad</i> .....	123
<i>Tabla 7. Categorías VIM y el Modelo de Proceso de Medición, ajustada de [88].</i> .....	127
<i>Tabla 8. Estándares de medición VIM [1]</i> .....	128
<i>Tabla 9. Cantidades y unidades”: conceptos para el análisis de las medidas de tamaño</i> .....	129
<i>Tabla 10. Análisis de “cantidades y unidades” en RmFFP</i> .....	132
<i>Tabla 11. Resumen de trabajos sobre validación teórica</i> .....	138
<i>Tabla 12. Medidas del procedimiento de medición RmFFP</i> .....	143
<i>Tabla 13. Correspondencia entre conceptos de metrología y el proceso de RmFFP</i> .....	155
<i>Tabla 14. Resultados del SCV usando COSMIC-FFP [104].</i> .....	159
<i>Tabla 15. Movimientos de datos identificados con RMFFP</i> .....	- 164 -
<i>Tabla 16. Comparación de movimientos de datos identificados con COSMIC-FFP y RmFFP</i> .....	- 166 -
<i>Tabla 17. Estadística descriptiva para la variabilidad de RmFFP..</i> -	173 -
<i>Tabla 18. Prueba de T-student para la variabilidad de RmFFP</i> .....	- 173 -
<i>Tabla 19. Variabilidad de los tipos de movimientos de dato</i> .....	- 174 -
<i>Tabla 20. Valores descriptivos para la Facilidad de Uso Percibida</i> -	192 -
<i>Tabla 21. Prueba de T-Student para la Facilidad de Uso Percibida</i> -	192 -
<i>Tabla 22. Valores descriptivos para la Utilidad Percibida</i> .....	- 193 -
<i>Tabla 23. Prueba de T-Student para la Utilidad Percibida</i> .....	- 193 -

<i>Tabla 24. Valores descriptivos para la Intención de Uso .....</i>	<i>- 194 -</i>
<i>Tabla 25. Prueba de T-Student para la Intención de Uso.....</i>	<i>- 195 -</i>
<i>Tabla 26. Modelo de Regresión entre la FUP, UP y la IU.....</i>	<i>- 196 -</i>
<i>Tabla 27. Regresión simple entre la UP y la IU .....</i>	<i>- 199 -</i>
<i>Tabla 28. Análisis de la validez constructiva del cuestionario .....</i>	<i>202</i>
<i>Tabla 29. Fiabilidad para los constructores del MAM.....</i>	<i>- 203 -</i>
<i>Tabla 30. Análisis de la validez constructiva del cuestionario .....</i>	<i>- 206 -</i>
<i>Tabla 31. Análisis de la validez constructiva del cuestionario (sin P2 y P12)</i> <i>.....</i>	<i>- 207 -</i>
<i>Tabla 32. Fiabilidad para los constructores de MAM.....</i>	<i>- 208 -</i>
<i>Tabla 33. Valores descriptivos para los constructores del MAM....</i>	<i>- 208 -</i>
<i>Tabla 34. Prueba de T-Student para los constructores del MAM....</i>	<i>- 209 -</i>

## *Lista de Abreviaturas*

<b>ARF</b>	<i>Árbol de refinamiento de funciones</i>
<b>BDR</b>	<i>Base de Datos Relacional</i>
<b>CFSU</b>	<i>Cosmic Functional Size Unit</i>
<b>COSMIC</b>	<i>Consortio Internacional de Métricas de Software Común</i>
<b>CS</b>	<i>Capa de Software</i>
<b>CU</b>	<i>Caso de Uso</i>
<b>DET</b>	<i>Tipos de Elementos de Dato</i>
<b>EI</b>	<i>Entradas Externas</i>
<b>EO</b>	<i>Salidas Externas</i>
<b>EQ</b>	<i>Consultas Externas</i>
<b>FFP</b>	<i>Full Function Point</i>
<b>FPA</b>	<i>Function Point Analysis</i>
<b>FSM</b>	<i>Function Size Measurement</i>
<b>FTR</b>	<i>File Type Referenced</i>
<b>FUR</b>	<i>Functional User Requirements</i>
<b>GQM</b>	<i>Goal/Question/Metric</i>
<b>IFPUG</b>	<i>International Function Point User Group</i>

<b>ILF</b>	<i>Archivo Lógico Interno</i>
<b>MAM</b>	<i>Modelo de Adopción del Método</i>
<b>MD</b>	<i>Movimientos de Datos</i>
<b>MERODE</b>	<i>Model Driven Existence-dependency Related Object Oriented Development</i>
<b>MROO</b>	<i>Modelos de Requisitos OO Method</i>
<b>MSAS</b>	<i>Espacio de métrica con segmentos aditivos</i>
<b>NCM</b>	<i>Número de movimientos de datos tipo lectura identificados en cada Condición de Mensaje</i>
<b>NCRE</b>	<i>Número de movimientos de datos tipo lectura identificados en cada Condición de Relación de Extend</i>
<b>NMC</b>	<i>Número de movimientos de datos tipo escritura identificados en el mensaje Connect</i>
<b>NMQ</b>	<i>Número de movimientos de datos tipo lectura identificados en cada Mensaje Query</i>
<b>NMSD</b>	<i>Número de movimientos de datos tipo escritura identificados en cada mensaje Service/Destroy</i>
<b>NMSI</b>	<i>Número de movimientos de datos tipo entrada identificados en cada Mensaje Signal con el valor Input</i>
<b>NMSN</b>	<i>Número de movimientos de datos tipo escritura identificados en cada Mensaje Service/New</i>

<b>NMSO</b>	<i>Número de movimientos de datos tipo salida identificados en cada Mensaje Signal con valor Output</i>
<b>NMSU</b>	<i>Número de movimientos de datos tipo escritura identificados en cada Mensaje Service/Update</i>
<b>NPCU</b>	<i>Número de movimientos de datos tipo lectura identificados en cada precondition de casos de uso</i>
<b>PAD</b>	<i>Pesos de Actores Desajustados</i>
<b>PAR</b>	<i>Proceso de Análisis de Requisitos</i>
<b>PCUD</b>	<i>Pesos de Casos de Uso Desajustados</i>
<b>PF</b>	<i>Procesos Funcionales</i>
<b>RET</b>	<i>Tipos de Elementos de Registro</i>
<b>RETO</b>	<i>Requirements Tools</i>
<b>ROOM</b>	<i>Lenguaje de modelado formal para el desarrollo de sistemas de tiempo real</i>
<b>RUP</b>	<i>Rational Unified Process</i>
<b>SAPS</b>	<i>Estructura de proximidad segmentalmente aditiva</i>
<b>AM</b>	<i>Modelo de Aceptación de la Tecnología de Davis</i>
<b>TCF</b>	<i>Factor de Complejidad Técnico</i>
<b>UKSMA</b>	<i>Asociación de Métricas del Reino Unido</i>
<b>UML</b>	<i>Unified Modeling Language</i>

<b>UMDE</b>	<i>Universo de Movimientos de Datos tipo Entrada</i>
<b>UMROO</b>	<i>Universo de Modelos de Requisitos OO-Method</i>
<b>VC</b>	<i>Validez Convergente</i>
<b>VD</b>	<i>Validez Divergente</i>

Anexo	<b><i>Caso de estudio:</i></b>
<b>A</b>	<b><i>Alquiler de vehículos</i></b>

El presente caso de estudio trata sobre una empresa que tiene por actividad principal el alquiler de vehículos. Sin embargo, esta actividad conlleva a otra serie de actividades derivadas como el mantenimiento y reparación de vehículos, y alquiler de extras (teléfono, chofer,...). Además, normalmente, los vehículos a alquilar son comprados al comienzo de la temporada y suelen venderse al finalizar la misma. Las operaciones de compra de los vehículos suelen contemplar la venta de los mismos al cabo de cierto periodo de tiempo (seis meses) que también se necesita automatizar.

### ***1. Descripción del sistema***

Este sistema realizará la gestión de la compra, alquiler, mantenimiento y venta de vehículos, así como la gestión de los distintos tipos de clientes que operan con la empresa. Dichas funcionalidades se describen a continuación:

#### ***Vehículos***

La flota de vehículos de la empresa varía aproximadamente de 250 unidades en temporada alta (meses de abril a septiembre) a 50 unidades en temporada baja (resto del año). Los datos de interés de un vehículo son:



<b>Registro de vehículo</b>	
<i>Código</i>	124111
<i>Matricula</i>	V-4512-EC
<i>Marca</i>	Ford
<i>Clase</i>	Turismo
<i>Grupo</i>	B
<i>Fecha de compra</i>	01/03/1999
<i>Kilometraje</i>	125

Los vehículos se clasifican de acuerdo con ciertos criterios:

- **Clase:** atiende al uso al que esta destinado, y puede ser: Turismo, Furgoneta, Industriales.
- **Grupo:** en función del tamaño o prestaciones del mismo y a efectos de facturación. Por ejemplo, los vehículos del Grupo A son los vehículos más sencillos y pequeños de una clase, y los que corresponden a un menor precio de alquiler.

El estado de un vehículo representa la situación en que este se encuentra, es decir:

- **Alquilado:** El vehículo esta asignado a un contrato.
- **A preparar:** el coche acaba de devolverse por algún cliente que lo había alquilado, por lo que debe prepararse antes de utilizarlo (limpieza, mantenimiento, etc.). También cuando el vehículo es nuevo debe pasar a este estado para ser puesto a punto antes de utilizarse por primera vez.
- **Listo:** El vehículo se encuentra en condiciones de ser alquilado.

Otro aspecto a considerar es la disponibilidad del vehículo. Siempre que el vehículo no este en el estado alquilado, puede ser deshabilitado para

alguna actividad especial (por ejemplo: ser reservado por algún gerente de la empresa) o para la realización de tareas como:

- **Preparación:** el vehículo pasa a limpieza, verificación del combustible, agua, aceite, etc.
- **Mantenimiento:** el vehículo se encuentra en operaciones de mantenimiento. Deben registrarse los datos de la operación (código de vehículo, fecha de entrega, código, enviado por, y descripción).
- **Reparación:** el vehículo debe ser reparado por alguna avería. En este caso, la reparación puede llevarse a cabo en un taller externo o en la propia base (taller propio de la empresa). En cualquiera de los dos casos deben registrarse los datos relativos a la reparación (fecha, código de vehículo, precio, enviado por, descripción).

Un punto importante a considerar es que cuando un vehículo esta deshabilitado no puede ser alquilado, pero sí vendido.

Observación: por política de la empresa, aquellos vehículos que sobrepasen los 150000 Km., automáticamente deben ser deshabilitados y no pueden ser habilitados nuevamente para ser alquilados.

Otros aspectos a considerar respecto a los vehículos son:

- **Compras:** al momento de la compra del vehículo se registran sus datos: marca, modelo, matrícula, etc., pudiéndose concertar además la posterior venta del vehículo, registrando un precio y una fecha de entrega (normalmente al finalizar la temporada). Posteriormente a la compra y automáticamente se procede a deshabilitar el vehículo de forma que pueda ser puesto a punto por el personal de la empresa (preparación) antes de alquilarlo.

- **Venta:** el vehículo puede ser vendido concertando un precio y una fecha de entrega. Llegada la fecha de entrega, si el vehículo no estuviera alquilado deberá de ser deshabilitado automáticamente, en espera de ser retirado o enviado a su nuevo propietario. A continuación mostramos un registro de la venta de un vehículo.
- **Seguro de alquiler:** Todos los vehículos de la flota deben estar amparados por un seguro. Debe registrarse información como: nombre de la empresa aseguradora, tipo de cobertura y periodo de cobertura. Si finaliza el periodo de cobertura del seguro de un vehículo y el mismo no está en alquiler, deberá ser automáticamente deshabilitado y no podrá volver a habilitarse a menos que se amplíe nuevamente la cobertura.

### ***Cliente***

Atendiendo al origen que lo referencia, el cliente puede ser de uno de los dos tipos siguientes:

- **Directa** cliente fijo u ocasional al cual se le aplica la tarifa general de acuerdo al contrato. Se recogen todos los datos personales del cliente para el contrato: nombre, dni, dirección, teléfono, tipo de tarjeta de crédito, número de tarjeta de crédito, número de carné de conducir, etc.
- **Agencia o intermediaria:** en este caso el cliente es alguna agencia sucursal u otra empresa intermediaria, por tanto no se recogen los datos personales del conductor.

## **Tarifa**

Las tarifas representan la cantidad a cobrar al cliente por el alquiler de un vehículo y por el seguro por unidad de tiempo, y depende principalmente de las características del contrato y de la clase y grupo del vehículo. Un ejemplo de registros se muestra a continuación:

<i>Tarifas de alquiler/seguros (por día)</i>			
	<i>Turismo</i>	<i>Furgoneta</i>	<i>...</i>
<i>Grupo A</i>	<i>600/100</i>	<i>700/100</i>	
<i>Grupo B</i>	<i>750/200</i>	<i>800/150</i>	
<i>...</i>			

## **Contrato**

Una vez pactado con el cliente las condiciones de alquiler se elabora un contrato. Este contrato debe de ser único para el alquiler de cada coche. Los datos principales a registrar se muestran a continuación:

Otros aspectos importantes a considerar con respecto al contrato son:

- **Seguro:** los vehículos cuando se alquilan deben de estar amparados por un seguro, que se cobra al cliente por un importe por unidad de tiempo que va asociado a clase y grupo al que pertenece. Puede registrarse también un monto que expresa el importe que ha de cubrir el cliente en caso de siniestro.
- **Extras:** en ocasiones, además, se prestan otros servicios adicionales que se denominan extras, entre los que podemos mencionar: Aeropuerto (entrega o recogida), combustible, accesorios en el vehículo (baca, asiento bebé, teléfono). A cada extra se le asigna un código de extra y un precio por día por unidad independientemente de las características del vehículo o del cliente.

A cualquier contrato (coche en alquiler), es posible agregarle estos extras durante los días deseados y en la cantidad deseada (por ejemplo: a un contrato de una semana dos teléfonos móviles durante tres días)

### ***Empleados***

En la empresa trabajan varios empleados que pueden tener un cargo de Usuario o Administrador. Los usuarios pueden realizar todas las tareas de gestión habituales (alquilar y devolver vehículos, cambiar datos de un alquiler en curso, mantener los datos de los clientes y mantener los datos de las operaciones sobre los vehículos), exceptuando las operaciones de compra, venta y eliminación de los vehículos de la flota, el mantenimiento de las distintas tarifas (alquiler, seguro y extras), mantener el registro de seguros de los coches y la gestión de los empleados de la empresa. Entre los datos a registrar de un empleado se tiene: número de registro, nombres y apellidos, fecha de ingreso, observaciones.

### ***Alquiler***

El cliente va a una de las oficinas del Alquiler de coches, y luego de consultar sobre los coches disponibles y los precios respectivos, solicita el alquiler. Si se trata de un cliente nuevo se le abre un Registro de Cliente, asignándole un código de cliente. Se recogen los datos del alquiler, pudiendo anticipar una Fianza a cuenta del importe definitivo de la liquidación en concepto de garantía del buen fin del contrato.

Por último, se imprime el contrato, lo firma el cliente y se le entrega una copia del mismo junto con las llaves del vehículo alquilado.

Mientras el contrato esta abierto, con una llamada telefónica o personalmente, el cliente podrá cambiar algunos datos del contrato, como son: fecha de devolución, algún extra adicional o el lugar de devolución. Al

devolver el vehículo se verifica el estado del vehículo, comprobando el combustible utilizado y realizando la liquidación definitiva del contrato, devolviendo la fianza (si la hubiere) o pagar el saldo correspondiente.

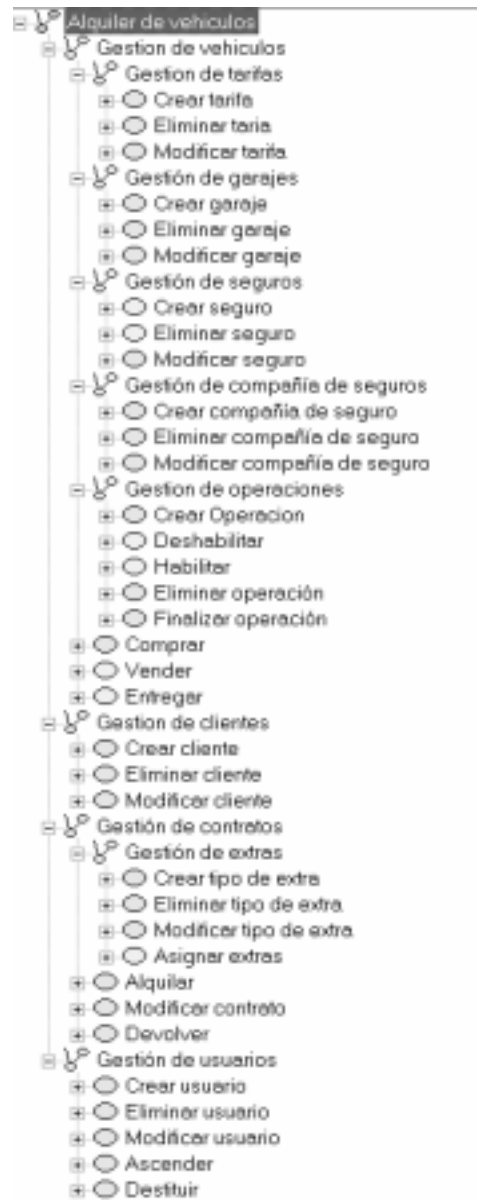
A continuación se presenta la especificación del caso de estudio presentado.

## **2. Especificación de los requisitos del sistema**

Esta especificación se ha basado en la aplicación del Modelo de Requisitos de OO-Method: elaboración del árbol de refinamiento de funciones, modelo de casos de uso y diagramas de secuencia.

### ***Árbol de refinamiento de funciones***

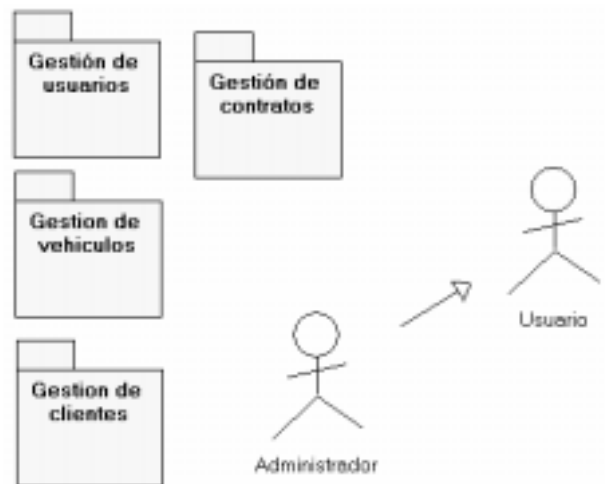
La Figura 1 muestra una lista de funcionalidades identificadas para el presente caso de estudio. Este árbol comprende 4 grupos funcionales: Gestión de vehículos, Gestión de clientes, Gestión de contratos, y Gestión de usuarios. Cada uno de estos grupos funcionales se descompone en sub-grupos funcionales ó directamente en funciones del sistema, los mismos que representan a los casos de uso. Este caso de estudio está conformado por 28 casos de uso primario y 7 casos de uso secundario.



**Figura 1.** Árbol de refinamiento de funciones de Alquiler de vehículos

## ***Diagrama de casos de uso***

A partir del árbol de refinamiento de funciones, se identifica los actores y su interacción con el sistema. La Figura 2 representa el diagrama de casos de uso general del sistema. Los cuatro grupos funcionales son representados por paquetes de UML, cada paquete está conformado por su respectivo diagrama de casos de uso, los cuales son presentados a continuación.

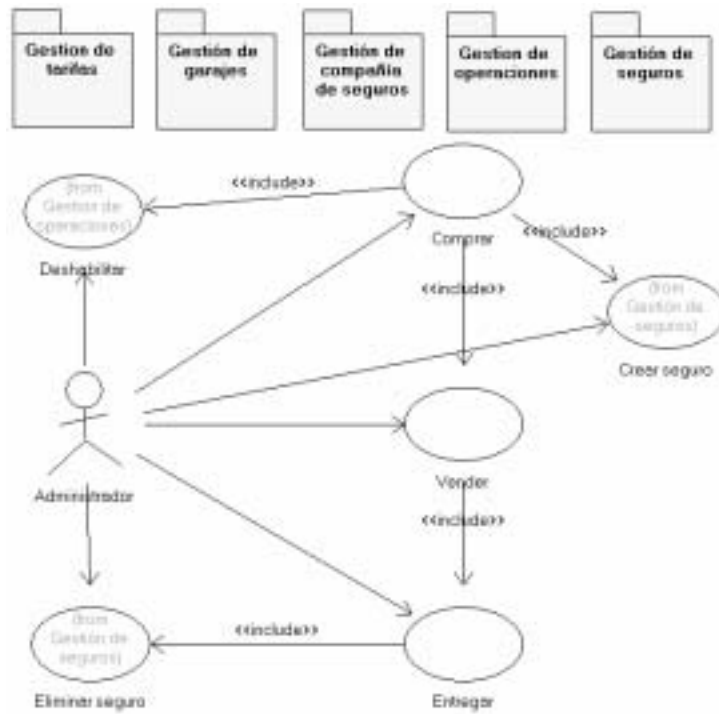


**Figura 2.** Diagrama de casos de uso general

### ***Gestión de vehículos***

En la Figura 3 se puede observar las operaciones de compra y venta de vehículos, así como de entrega del vehículo, cuando éste es retirado definitivamente de la flota de la empresa. Al mismo tiempo también se puede apreciar los diferentes subpaquetes de los que consta, los que son presentados a continuación.

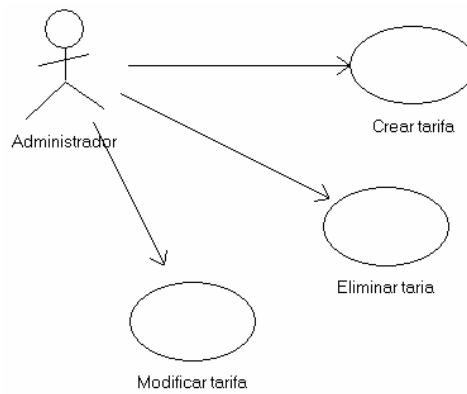




**Figura 3.** Diagrama de caso de uso de Gestión de vehículos

### ***Gestión de Tarifas***

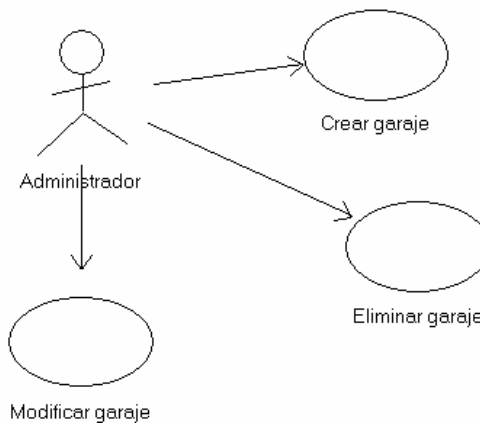
Como su propio nombre indica es la gestión de Tarifas. El administrador, encargado de realizar la gestión de tarifas, es el que directamente pondrá en funcionamiento los casos de uso que se observan en el diagrama de casos de uso (Vea Figura 4).



**Figura 4.** Diagrama de casos de uso de Gestión de tarifas

***Gestión de garajes***

Es similar a la anterior. En este caso nos referimos a la gestión de talleres o garajes. Tal como se observa en la Figura 5, vuelve a ser el administrador el actor encargado de realizar estas gestiones.

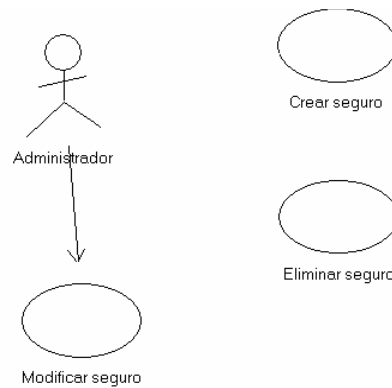


**Figura 5.** Diagrama de casos de uso de Gestión de garaje

***Gestión de seguros***

La Figura 6 muestra el diagrama de casos de uso referido con la gestión de pólizas de seguro. Como podemos observar, el administrador tan solo puede modificar los datos de una póliza, pero no puede crearla ni

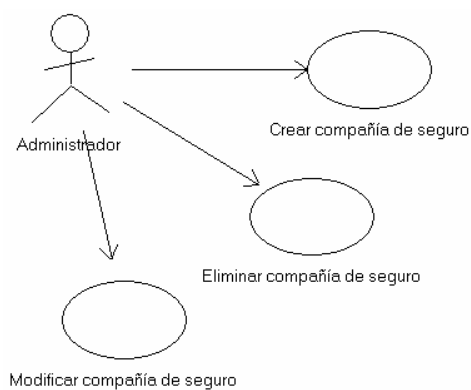
destruirla directamente. Esto se realiza en el momento de adquirir o vender un vehículo (Vea Figura 3).



**Figura 6.** Diagrama de casos de uso para la gestión de seguros

### ***Gestión de compañía de seguros***

Gestión de compañías de seguro. Es el administrador el encargado de realizar esta gestión. Como podemos ver en la Figura 7, una compañía de seguros puede ser tanto creada como modificada y eliminada.

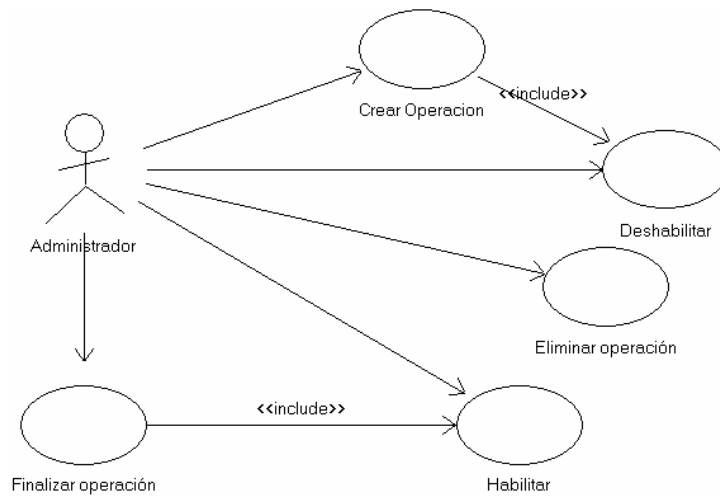


**Figura 7.** Diagrama de casos de uso sobre la gestión de compañía de seguro.

### ***Gestión de operaciones***

La Figura 8 muestra la gestión de operaciones realizadas sobre los vehículos. Hay operaciones que implican la deshabilitación del vehículo hasta

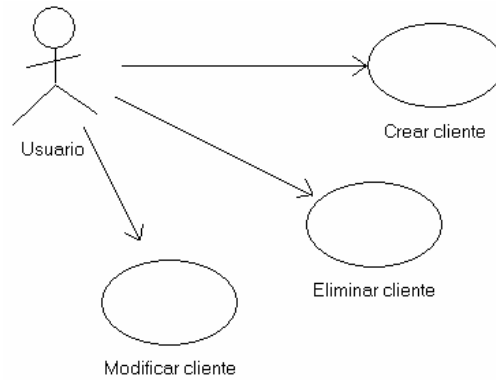
que a éste se le termina de realizar una operación, momento en el que se vuelve a habilitar. El encargado de realizar esta gestión es el usuario.



**Figura 8.** Diagrama de casos de uso de Operaciones

### ***Gestión de clientes***

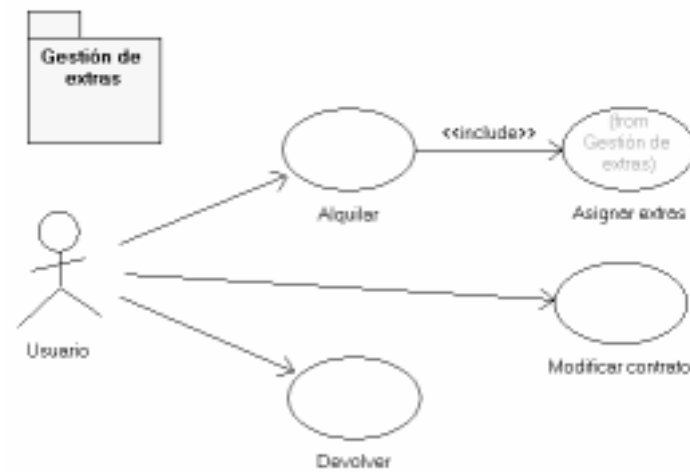
El usuario podrá crear clientes, modificar los datos de los que ya existen y eliminarlos, siempre y cuando se cumplan con una serie de restricciones (Vea Figura 9). A continuación mostramos el diagrama de casos de uso correspondiente a la gestión de clientes:



**Figura 9.** Diagrama de casos de uso de gestión de clientes

### ***Gestión de contrato***

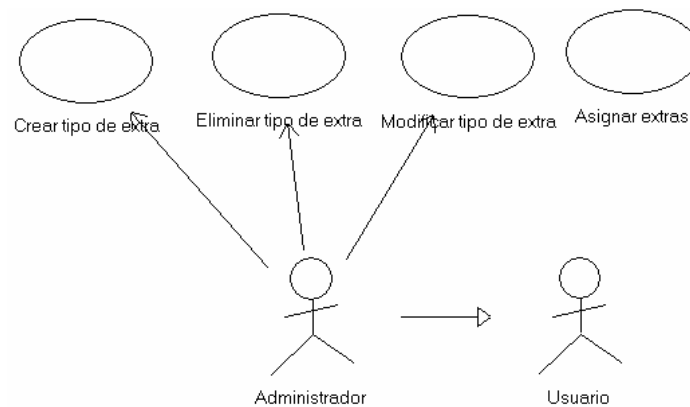
Este diagrama de casos de uso representa la gestión de contratos de alquiler. El usuario podrá alquilar, devolver o modificar un contrato. La operación de alquiler puede conllevar la asignación de algún tipo de extra, lo cual es representado en la Figura 10 como un paquete llamado Gestión de extras.



**Figura 10.** Diagrama de casos de uso de gestión de contratos

### ***Gestión de extras***

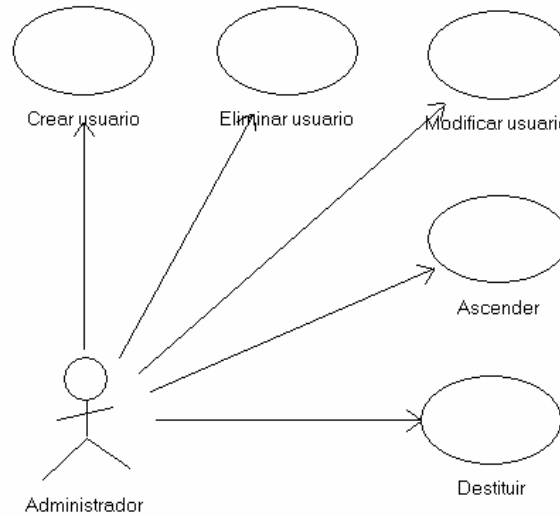
La Gestión de extras es llevada a cabo por el administrador y el usuario. Podremos realizar las operaciones básicas de creación, modificación y eliminación, así como la operación de asignación de extras a contratos. Esta asignación de extras no podrá ser accedida directamente.



**Figura 11.** Diagrama de casos de uso sobre la gestión de extras

### ***Gestión de usuarios***

El administrador se encargará de llevar a cabo esta gestión. Como podemos observar en la Figura 12, el administrador puede crear, modificar, eliminar usuarios así como elevarlos y bajarlos de categoría.



**Figura 12.** Diagrama de casos de uso sobre gestión de usuarios

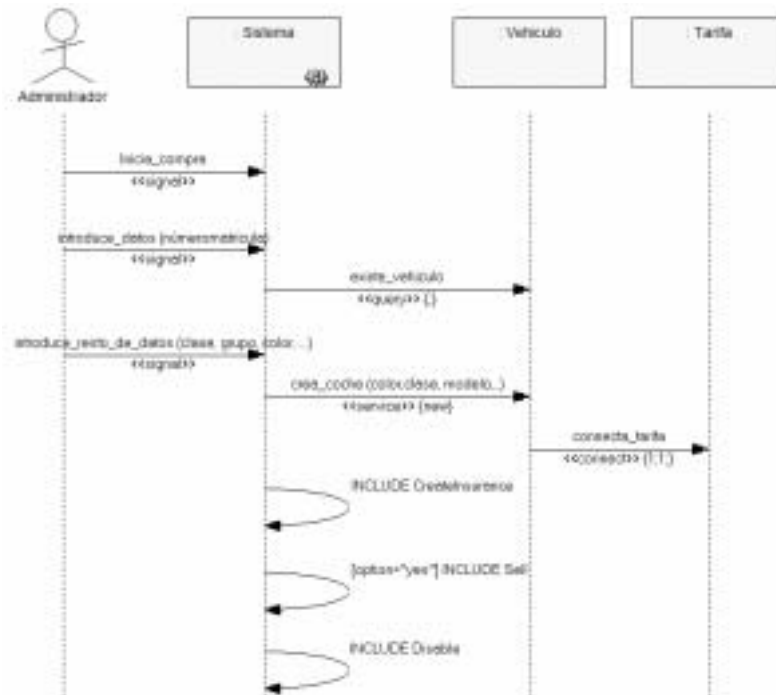
### ***Diagrama de secuencias***

Estos diagramas de secuencia son obtenidos a partir de la especificación de los casos de uso. A continuación estos diagramas son presentados a continuación.

#### ***Comprar***

Vemos que se trata de la creación de un objeto de la clase Car, el cual necesita de un objeto de la clase Rate, con el cual se asocia, y de un objeto de la clase Insurance, el cual crea. Por último, en el escenario de compra de un vehículo, éste debía ser deshabilitado para una puesta a punto antes de su primer uso (mensaje de TR\_Disable). Previamente, si se había pactado una

futura venta del vehículo se ha realizado la anotación de esa futura venta (mensaje TR\_Sell).

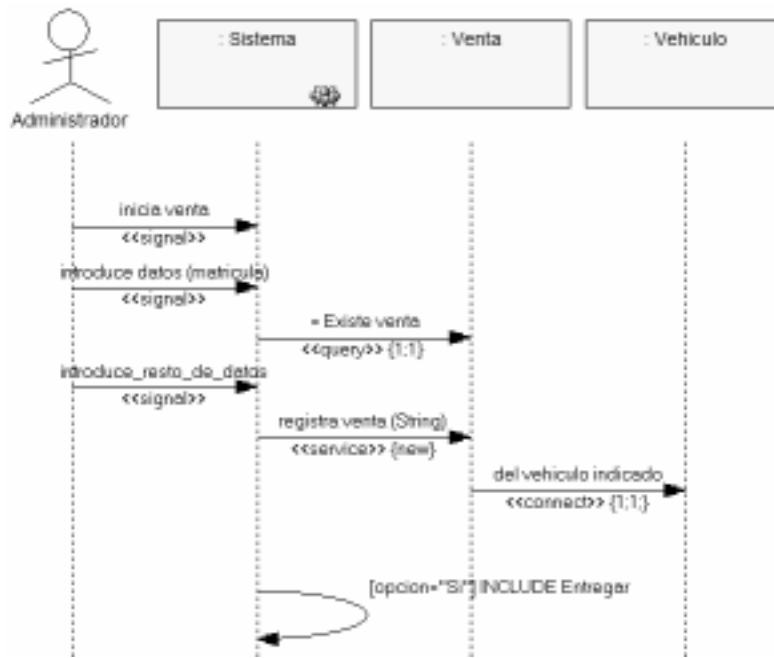


**Figura 13.** Escenario de Comprar de vehículo

### ***Vender***

Vemos que se trata de la creación de un objeto de la clase Sale, el cual se asocia con un objeto de la clase Car (objeto que va a ser vendido). Posteriormente a anotar la venta, se puede entregar al cliente de forma inmediata o no (mensaje TR\_Deliver).

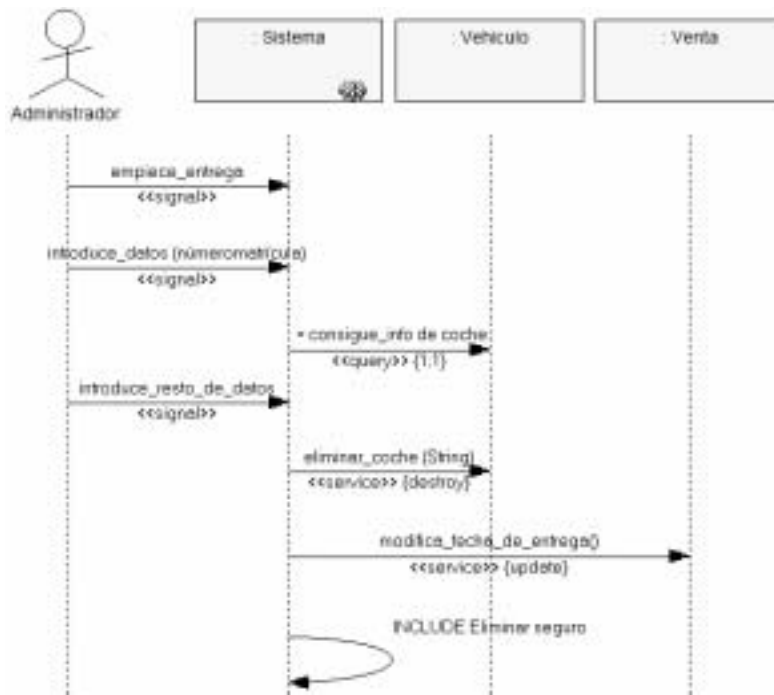




**Figura 14.** Escenario de venta de vehículo

### ***Entregar***

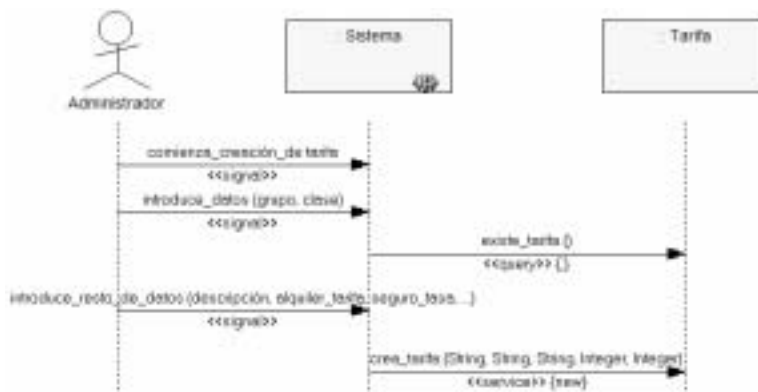
Supone la entrega del vehículo vendido al comprador. Supone la eliminación del objeto de la clase Car y la eliminación del objeto de la clase Insurance con el que estaba asociado el objeto de la clase Car.



**Figura 15.** Escenario de entrega de vehículo

### ***Crear tarifa***

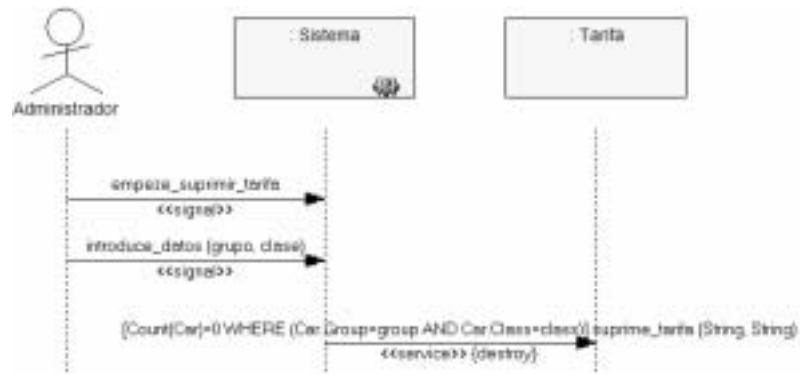
Se trata de la creación de objetos de la clase Rate, de acuerdo a unos valores que el actor proporciona.



**Figura 16.** Escenario de creación de tarifa

### ***Eliminar tarifa***

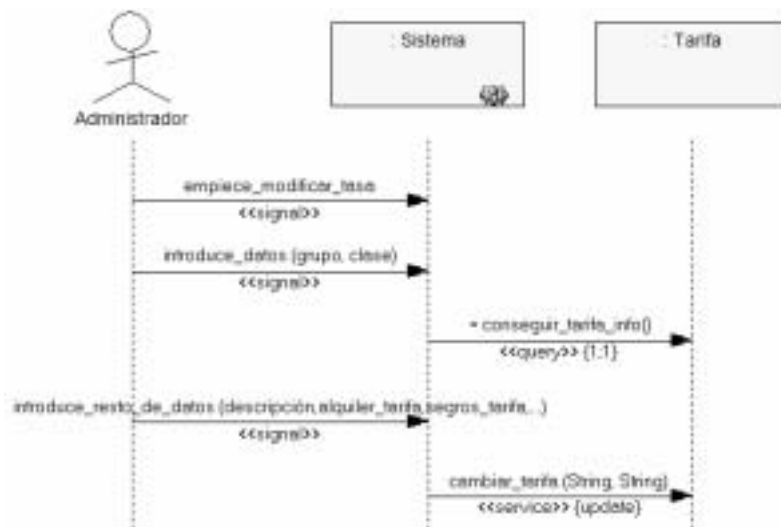
Se trata de la eliminación de los objetos de la clase Rate, aunque sujetos, como podemos ver en el último mensaje, a unas condiciones.



**Figura 17.** Escenario de eliminación de tarifa

### ***Modificar tarifa***

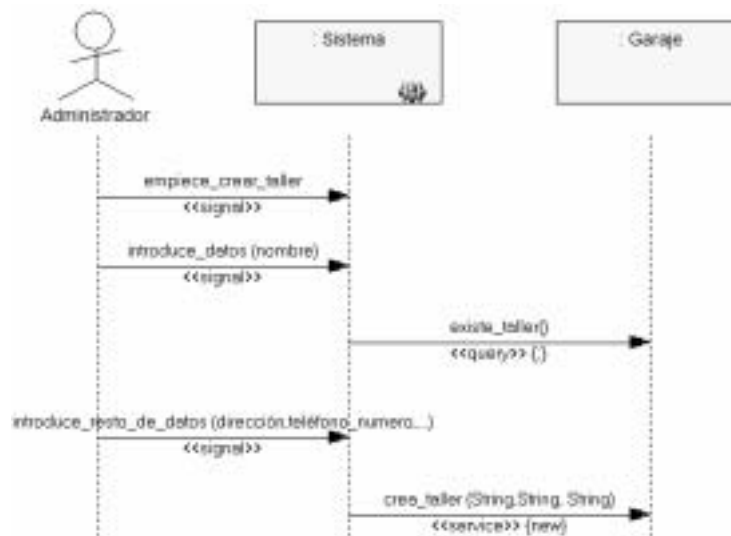
Modificación de los atributos de los objetos de la clase Rate. Se utilizará en posibles cambios de precio de alquiler o de seguro, etc.



**Figura 18.** Escenario de modificación de tarifa

### ***Crear garaje***

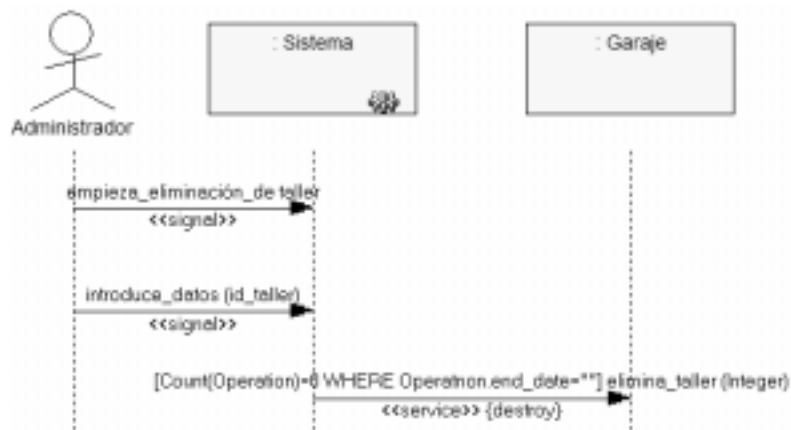
Creación de nuevos objetos de la clase Garage. Supone el alta de un nuevo taller al que la empresa llevará sus vehículos a reparar.



**Figura 19.** Escenario de creación de garaje

### ***Eliminar garaje***

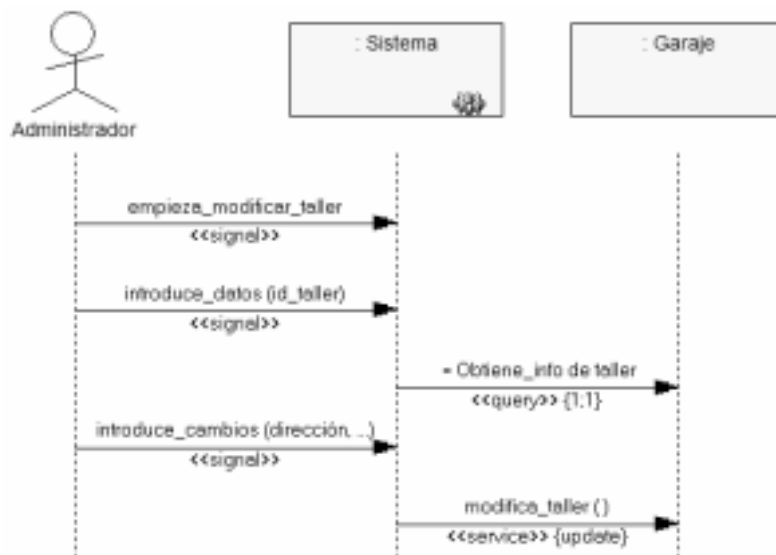
La eliminación de los objetos de la clase Garage, es decir, eliminación de talleres. Dicha eliminación esta sujeta a una serie de condiciones, expresadas en el último de los mensajes que se intercambian.



**Figura 20.** Escenario de eliminación de garaje

***Modificar garaje***

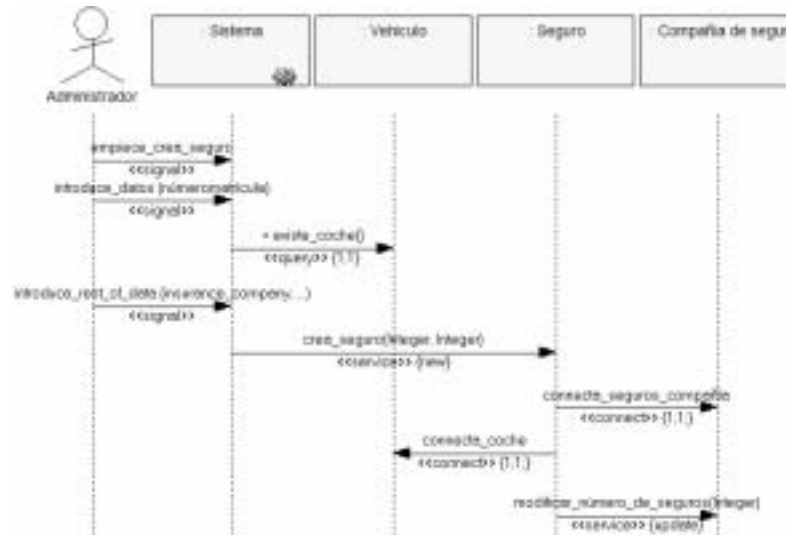
Función dedicada a la modificación de los datos de los talleres con los que la empresa de alquiler trabaja. A continuación podemos el diagrama de secuencia correspondiente a la modificación:



**Figura 21.** Escenario de modificación de garaje

### ***Crear seguro***

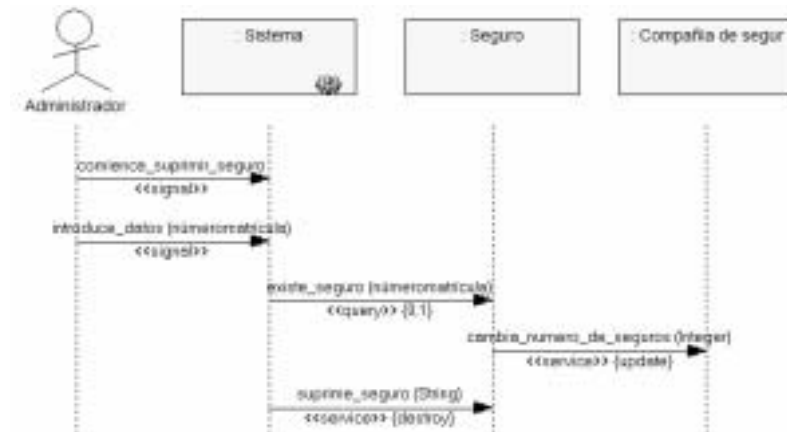
La creación de la póliza para un vehículo. Como podemos observar en el diagrama de secuencia, la creación del objeto Insurance, supone la asociación con un objeto de la clase Car y un objeto de la clase InsuranceCompany, que representan al vehículo que se quiere asegurar y a la compañía que lo va a asegurar, respectivamente.



**Figura 22.** Escenario de creación de seguro

### ***Eliminar seguro***

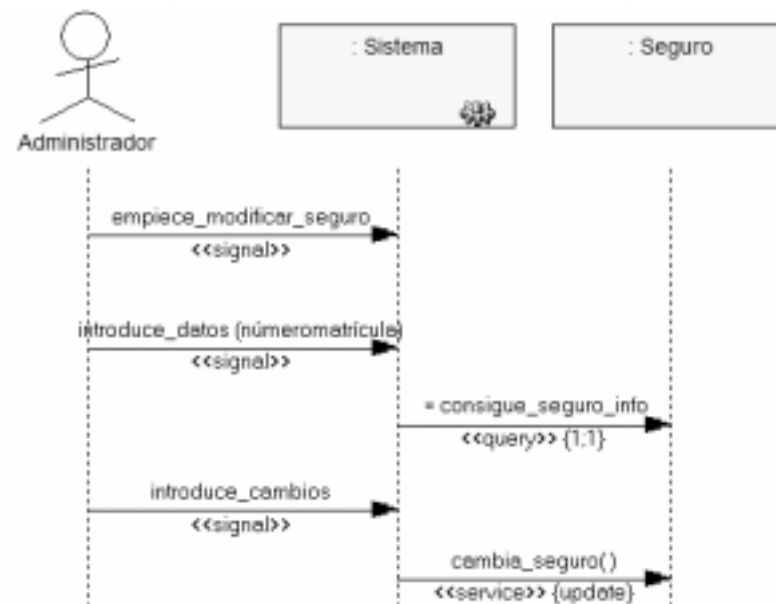
Este proceso tiene lugar en el momento en el que el objeto del tipo Car con el que estaba asociada la póliza es eliminado del sistema. Supone la eliminación del objeto Insurance.



**Figura 23.** Escenario de eliminación de seguro

### ***Modificar seguro***

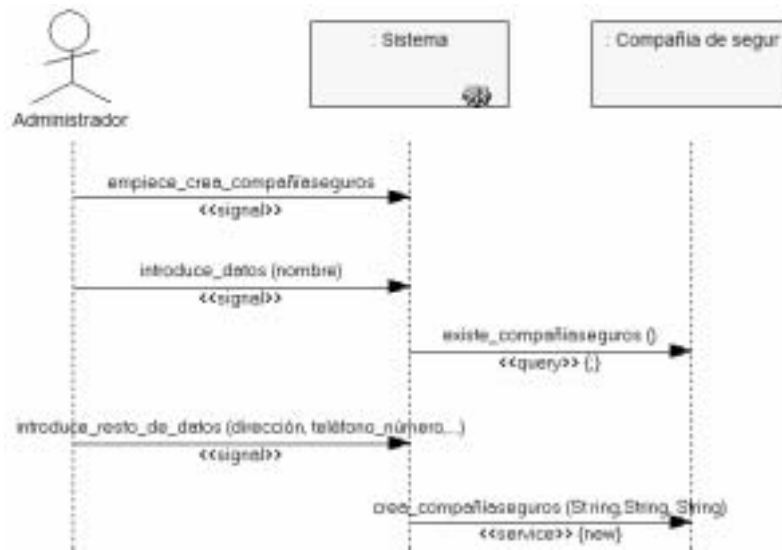
Tienen lugar cuando las condiciones de la póliza de seguro cambian, ya sea el periodo de cobertura como en el precio, etc.



**Figura 24.** Escenario de modificación de seguro

### ***Crear compañía de seguro***

Supone la creación de compañías de seguro, las cuales van a albergar y respaldar las pólizas que se creen y se asignen a dichas compañía.

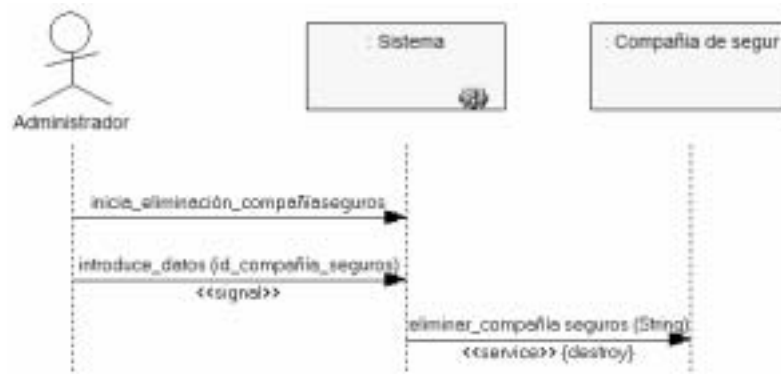


**Figura 25.** Escenario de creación de compañía de seguro

### ***Eliminar compañía de seguro***

Supone la eliminación de una compañía de seguros debido a diversos motivos (no alberga ninguna póliza, sus condiciones de cobertura no nos interesan, etc.).

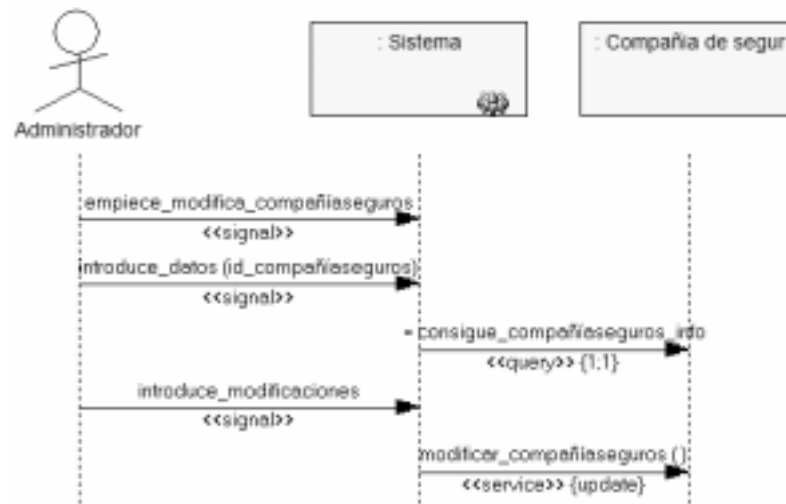




**Figura 26.** Escenario de eliminación de compañía de seguro

### ***Modificar la compañía de seguro***

Supone la modificación de los datos de la compañía de seguros, tales como su dirección, teléfono, etc.

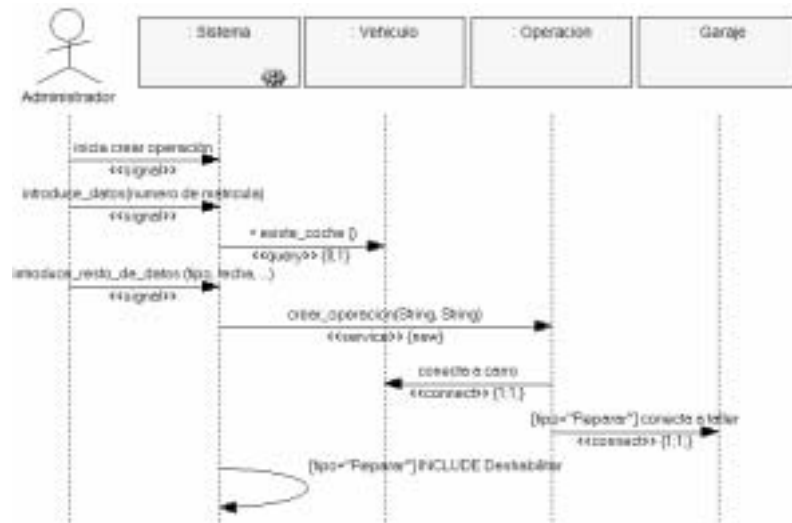


**Figura 27.** Escenario de modificación de compañía de seguro

### ***Crear operación***

Función llevada a cabo para realizar alguna operación en algún vehículo. Como podemos observar en la siguiente figura, el objeto creado de la clase Operation se asocia con un objeto de la clase Car y con un objeto de la

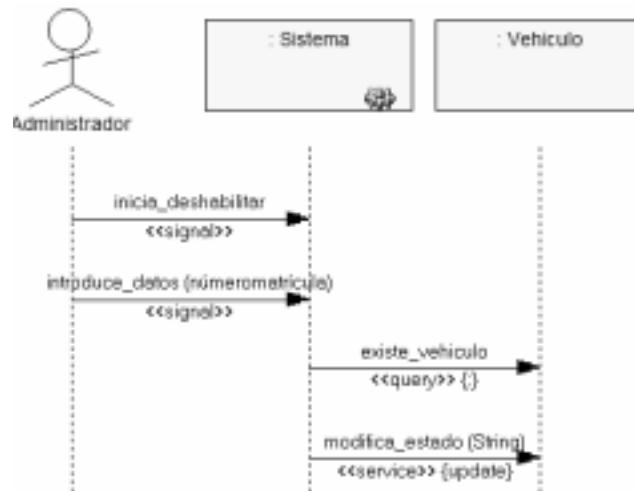
clase Garage, siempre y cuando la operación a realizar sea de reparación, con la consecuente deshabilitación del objeto Car.



**Figura 28.** Escenario de creación de operación

### ***Deshabilitar***

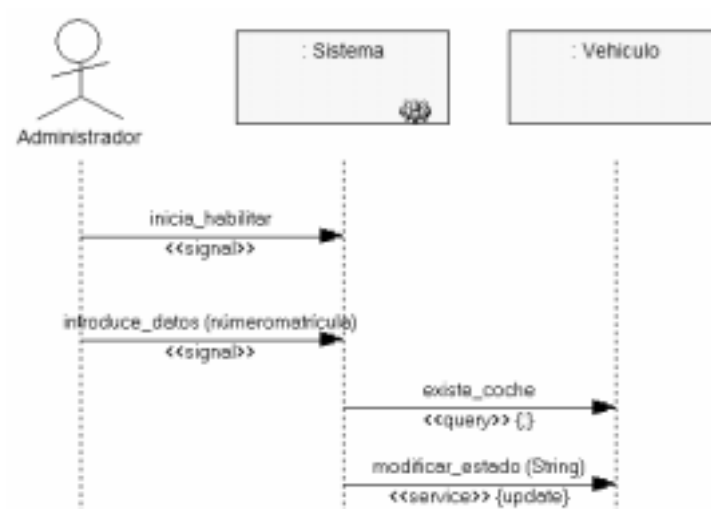
El diagrama de secuencia muestra la operación de deshabilitación llevado a cabo sobre un vehículo de la flota. Esta deshabilitación es ejecutada cuando el vehículo alquilado sobrepasa los 150000 Km.



**Figura 29.** Escenario de deshabilitación de vehículo

### ***Habilitar***

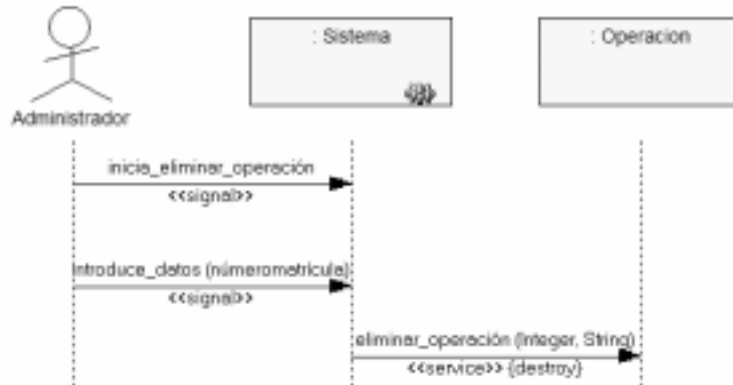
El diagrama de secuencia muestra la operación de habilitación llevado a cabo sobre un vehículo de la flota. Esta habilitación se ejecuta cuando alguna operación de mantenimiento o reparación ha sido finalizada.



**Figura 30.** Escenario de habilitación de vehículo

### ***Eliminar operación***

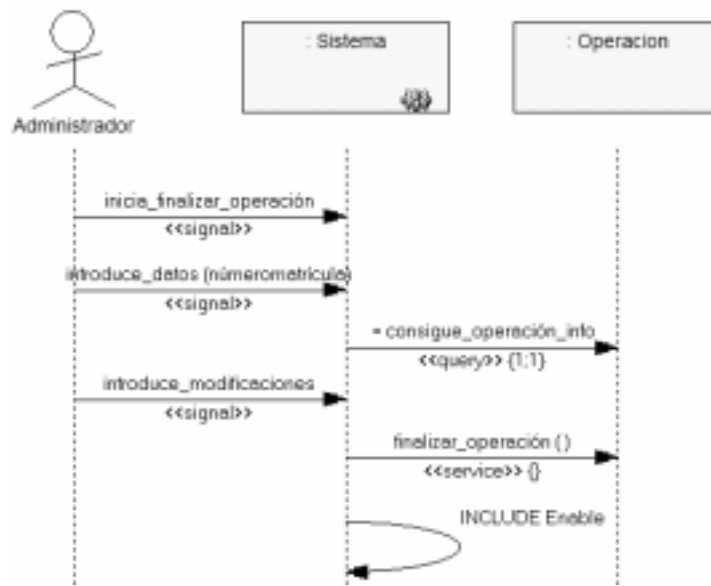
Función destinada a enmendar posibles errores en la creación de operaciones. Si creamos una operación y luego nos damos cuenta de que no es correcta, la podremos eliminar mediante el uso de esta función. No debe llevar a confusión que el borrar una operación no implica que la operación ya se haya realizado y que por tanto el vehículo puede ser habilitado. Para dicho efecto se dispone de la operación de FinaliceOperation.



**Figura 31.** Escenario de eliminación de operación

### ***Finalizar operación***

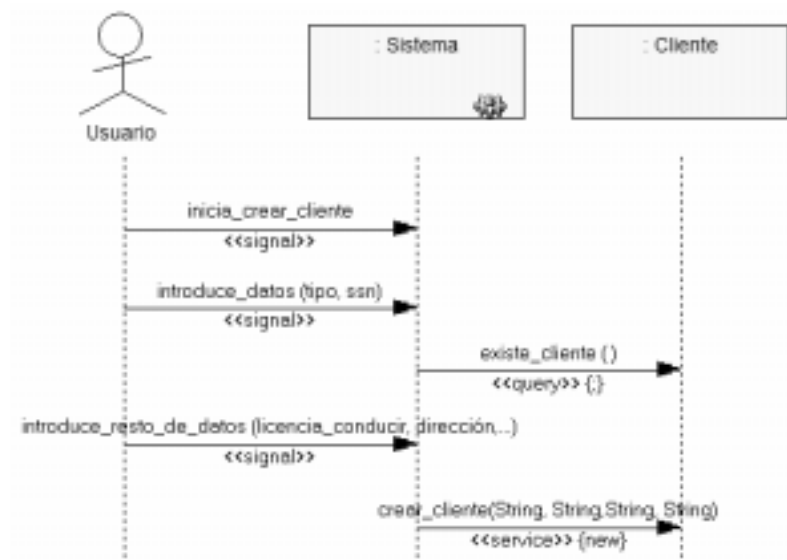
Función que nos indica que una operación ha llegado a su fin y que por tanto debemos habilitar el vehículo para su posterior alquiler.



**Figura 32.** Escenario de finalización de escenario

### ***Crear cliente***

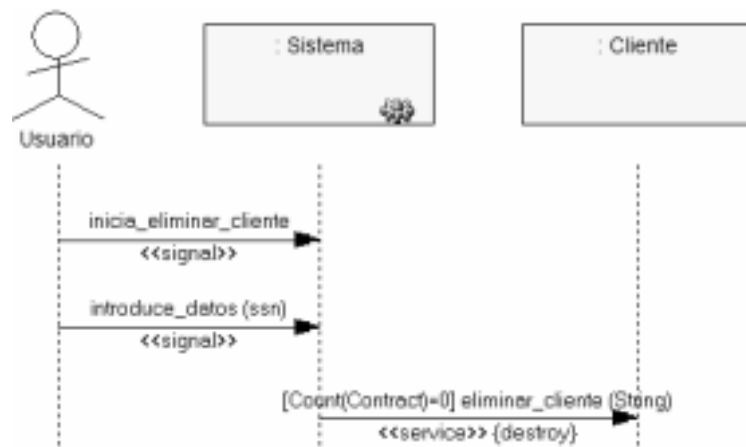
Función destinada a la creación de nuevos clientes, sean del tipo que sean. En la siguiente figura podemos ver los pasos que se siguen para llevar a cabo esta función.



**Figura 33.** Escenario de creación de cliente

***Eliminar cliente***

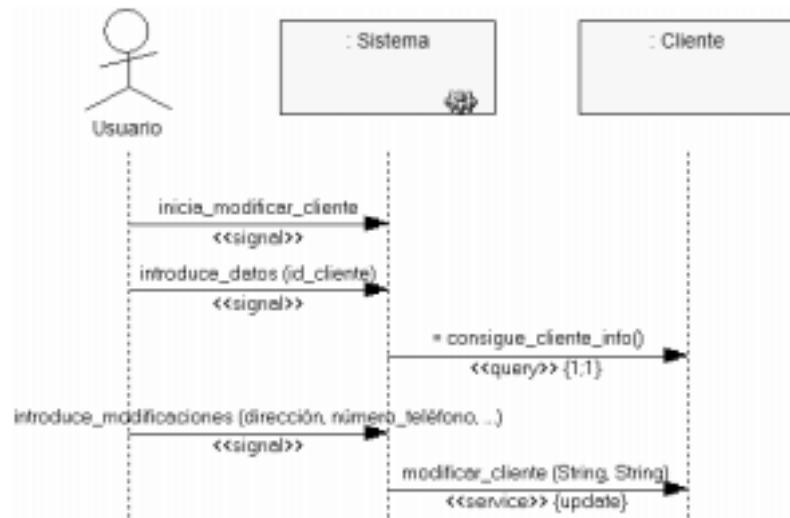
Operación destinada para eliminar del sistema a un cliente. Obviamente estará sujeta a unas restricciones, expresadas en el diagrama de secuencia en los mensajes pertinentes.



**Figura 34.** Escenario de eliminación de cliente

### ***Modificar cliente***

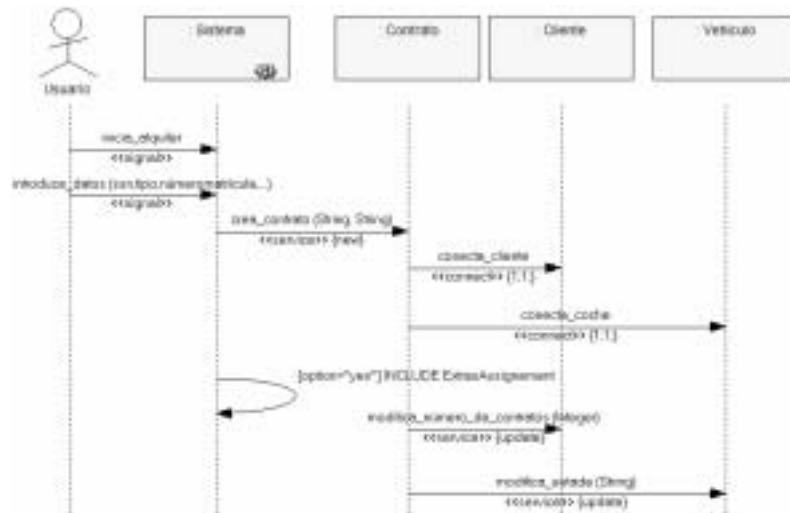
Función destinada a la modificación de los atributos de los objetos de la clase Customer, tales como dirección, teléfono, etc.



**Figura 35.** Escenario de modificación de cliente

### ***Alquilar***

Quizás la función más importante del sistema, por ser el objeto de negocio de la misma. Como podemos observar en el diagrama de secuencia, se crea el objeto de la clase Contract, el cual es asociado con un vehículo y un cliente. Después de crear el contrato, se pueden asignar los extras que se determinen, los cuales estarán asociados al contrato en cuestión:

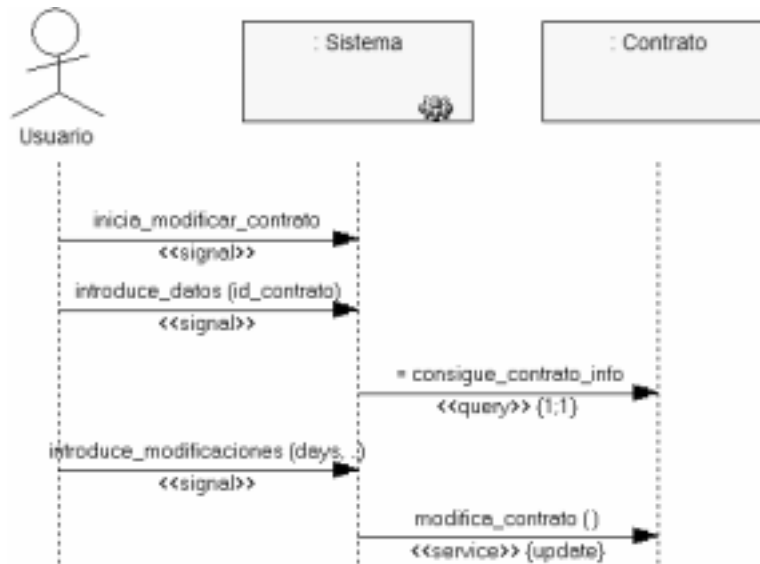


**Figura 36.** Escenario de alquilar un vehículo

### ***Modificar contrato***

Función destinada a modificar de forma puntual algunos de los aspectos del contrato, como pueden ser la fecha de devolución, el lugar de devolución, la forma de entrega, etc. Todo esto puede ser solicitado por teléfono o personalmente, indicando el número del contrato a modificar.

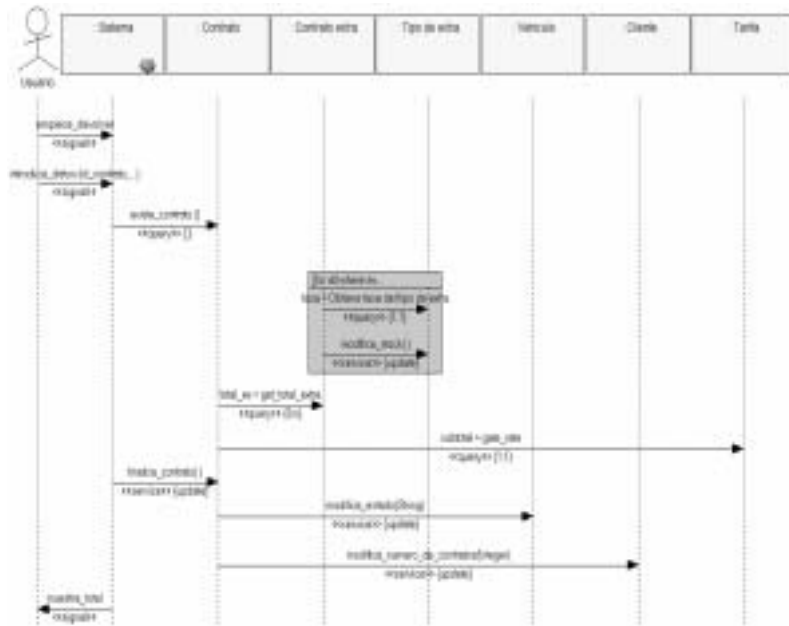




**Figura 37.** Escenario de modificación de contrato

***Devolver***

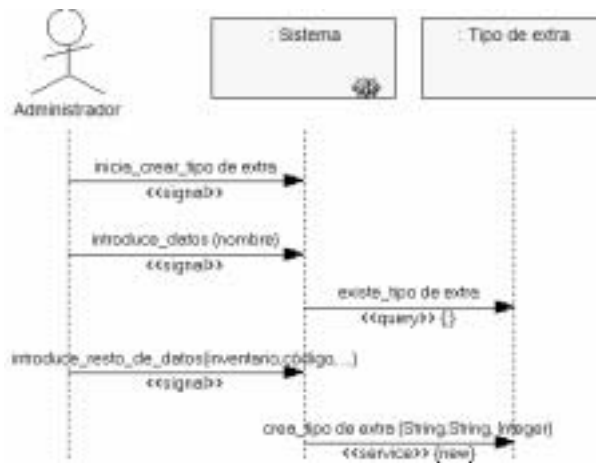
Otra de las funcionalidades importantes, es la devolución del vehículo, la devolución de los extras y el consecuente cobro al cliente.



**Figura 38.** Escenario de devolución de vehículo

### ***Crear tipo de extra***

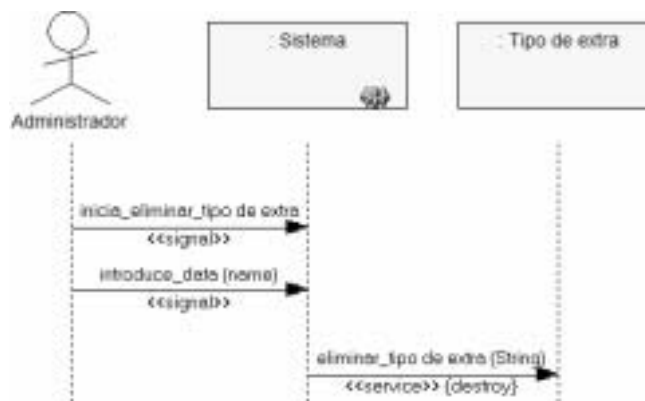
Creación de Tipos de Extras. Cada tipo de extra va a tener un atributo que indicará el stock disponible en cada momento, con lo que no tendremos que crear un extra por cada uno de ellos.



**Figura 39.** Escenario de creación de tipo de extra

***Eliminar tipo de extra***

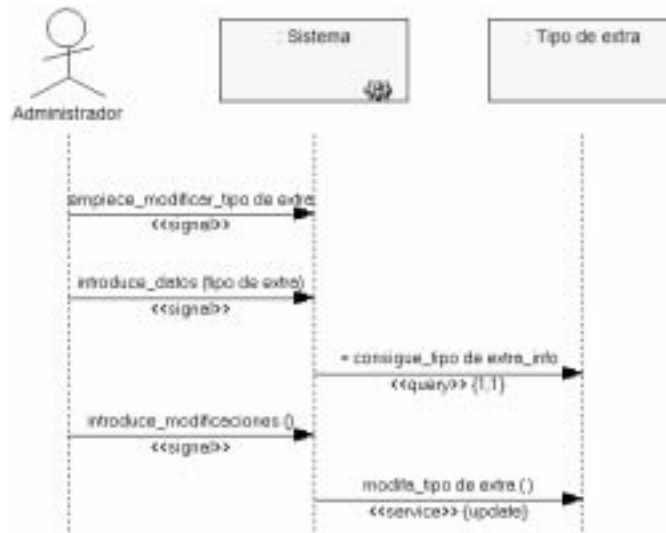
Función destinada a la eliminación de extras de un tipo determinado, bien por falta de existencias, por desfase del producto, por sustitución por otro, etc. Esta sujeto a determinadas restricciones.



**Figura 40.** Escenario de eliminación de tipo de extra

***Modificar tipo de extra***

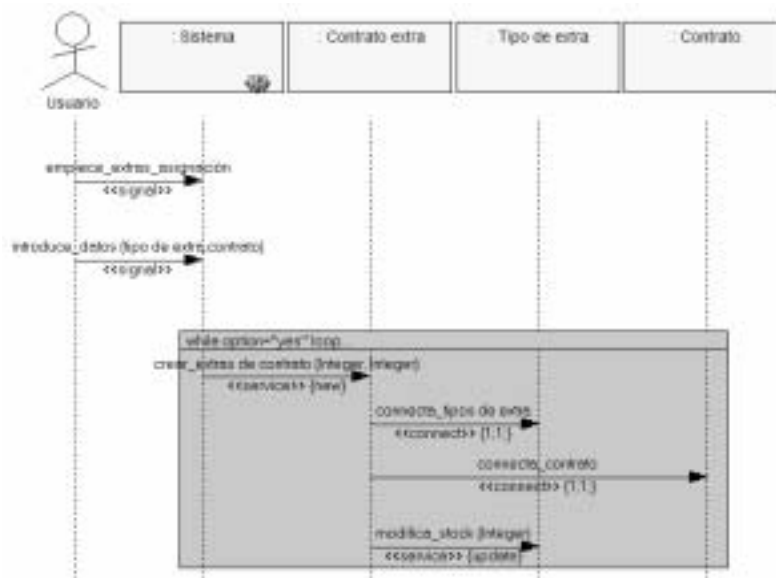
Función destinada a modificar las características de un extra, tal y como el precio de alquiler por día, descripción del mismo, etc.



**Figura 41.** Escenario de modificación de tipo de extra

### ***Asignar extras***

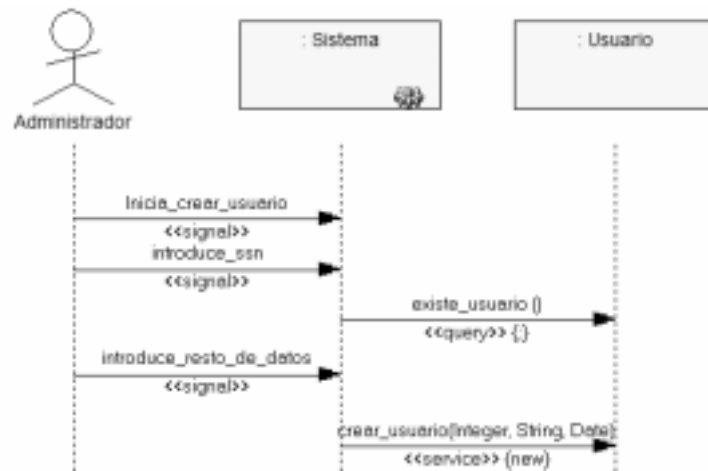
Función destinada a añadir a un contrato un determinado extra.



**Figura 42.** Escenario de Asignación de extras

### ***Crear usuario***

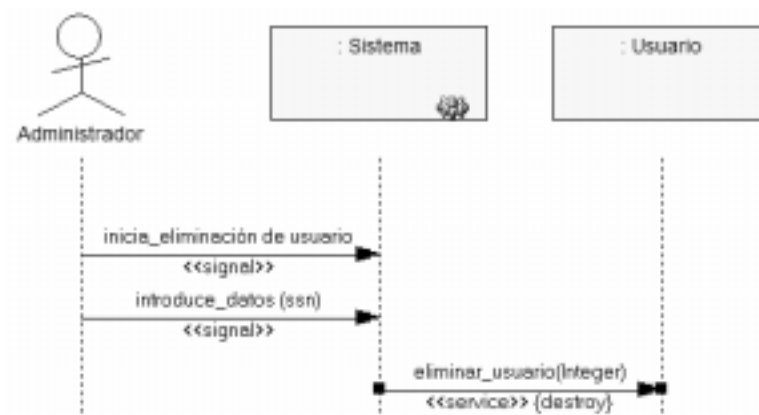
Función destinada a crear nuevos empleados de la empresa. Esta función es llevada a cabo por el administrador, el cual crea a los empleados.



**Figura 43.** Escenario: Crear usuario

### ***Eliminar usuario***

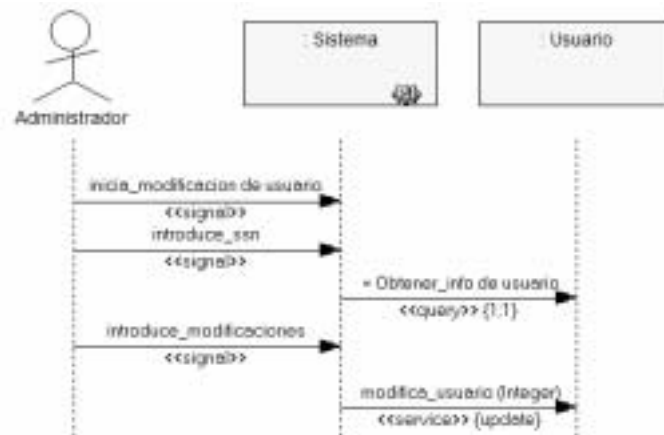
Función destinada a eliminar empleados. Eliminación de objetos de la clase User, debido a que ya ha dejado de trabajar para la empresa, etc.



**Figura 44.** Escenario: Eliminar usuario

### ***Modificar usuario***

Función destinada a la modificación de algunos de los atributos de un usuario, tales como la dirección, teléfono, cargo que ocupa en la empresa, etc.



**Figura 45.** Escenario: Modificar usuario

### ***Ascender***

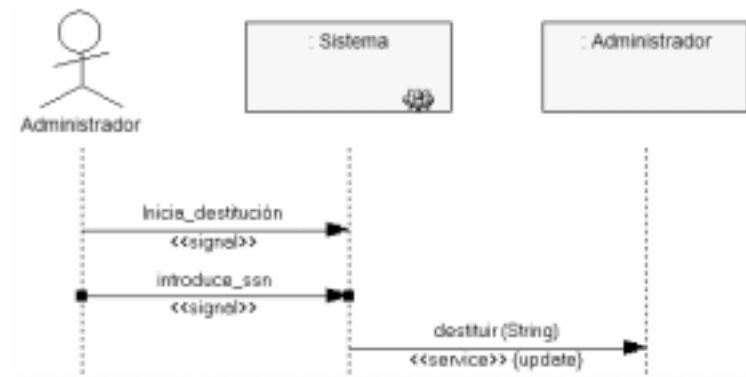
Cuando un usuario eleva su categoría a administrador, es este el que promociona al usuario. Para ello está esta función, mediante la cual un usuario pasaría a ser administrador.



**Figura 46.** Escenario: Ascender usuario

### ***Destituir***

Quando a un administrador se le restringe su categoría a usuario, es otro administrador el que destituye al administrador en cuestión. Para ello está esta función, mediante la cual un administrador pasaría a ser usuario.



**Figura 47.** Escenario: Destituir administrador

Anexo	<b><i>RmFFP: un procedimiento de medición</i></b>
<b>B</b>	

### **I.- Descripción general**

RmFFP es un procedimiento de medición basado en el método estándar COSMIC-FFP v2.2, cuyo tamaño funcional es expresado en “Cfsu”, la cual representa al movimiento de dato como unidad básica.

### **II.- Proceso de RmFFP**

Los pasos a seguir para aplicar el procedimiento de medición RmFFP son los siguientes:

*1.- Identificación del contexto de la medición.*

*2.- Construcción del modelo del software (Fase de Mapeo)*

*2.1. Se identifican las capas de software*

*2.2. Se identifica la frontera del sistema en el Diagrama de Casos de Uso para definir los límites del sistema a ser medido.*

*2.3. Se analiza el Modelo de Casos de Uso para la identificación de los procesos funcionales.*

*2.4. Se analizan los Diagramas de Secuencia para la identificación de los grupos de datos.*

*2.5. Se analizan los Diagramas de Secuencia para*



<p>3.- Aplicación de las reglas de asignación numérica. (Fase de Medición)</p>	<p>la identificación de los atributos de datos.</p>
	<p>2.6. Se analizan los Diagramas de Secuencia para la identificación de los tipos de movimientos de datos.</p>
	<p>2.7. Se eliminan movimientos de datos duplicados aplicando las reglas de duplicidad.</p>
	<p>3.1. Se aplica la función de medición a cada uno de los movimientos de datos identificados.</p>
	<p>3.2. Se aplica las respectivas funciones de agregación a nivel de proceso funcional.</p>
	<p>3.3. Se aplica las respectivas funciones de agregación a nivel de capa.</p>
	<p>3.4. Se aplica la función de agregación para los requisitos funcionales que han sido modificados</p>

### **1.- Identificación del contexto de la medición**

El contexto de la medición es determinado por la identificación del propósito, alcance y punta de vista de medición.

- a) El *propósito* de RmFFP es medir el tamaño funcional de sistemas orientado a objetos producidos en el contexto de OO-Method.
- b) El *alcance de medición* es comprendida por la funcionalidad a ser incluida en una medición particular; esta funcionalidad es especificada usando el Modelo de Requisitos de OO-Method.
- c) El *punto de vista de medición* corresponde al punto de vista del usuario final.

## ***2.- Construcción del modelo del software***

### ***2.1. Identificación de capas de software:***

Se identifica *una sola capa de software*, ya que todos los requisitos funcionales especificados pertenecen a un mismo nivel de abstracción.

### ***2.2. Identificación de la frontera de software:***

Como la frontera se constituye en la interfaz que existe entre el sistema a ser medido y los usuarios, las reglas a aplicar para este paso son:

***Regla 1:*** “Considere cada actor de un diagrama de casos de uso como un usuario del sistema”.

***Regla 2:*** “Considere la frontera del sistema como el borde entre el conjunto de casos de uso a ser medidos y los usuarios identificados”.

### ***2.3. Identificación de Procesos Funcionales:***

***Regla 3:*** “Considere cada caso de uso primario como un proceso funcional”.

***Regla 4:*** “Considere cada caso de uso secundario como un proceso funcional”.

### ***2.4. Identificación de Grupos de Datos:***

***Regla 5:*** “Considere a cada actor diferente que participa en algún diagrama de secuencia como un grupo de dato”.

***Regla 6:*** “Considere a cada clase de tipo entidad que aparece en algún diagrama de secuencia como un grupo de dato”.

### ***2.5. Identificación de Atributos de Datos:***

**Regla 7:** “Considere a cada argumento constante o variable en los mensajes <<service/new>> de un diagrama de secuencia como un atributo de dato”.

#### *2.6. Identificación de Movimientos de Datos:*

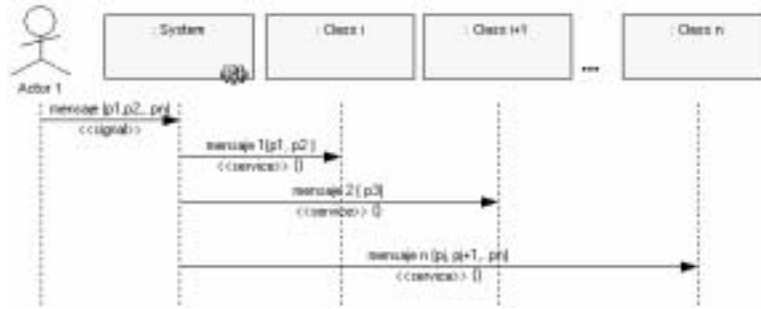
- a) Un movimiento de dato es identificado por cada grupo de dato involucrado en dicho movimiento.
- b) Cuatro tipos de movimientos de datos son identificados: entrada, lectura, escritura, y salida.

##### *2.6.1 Tipo de Movimiento de dato: ENTRADA*

**Regla 8:** “Considere a cada mensaje etiquetado con el estereotipo <<signal>> y con el valor input como un movimiento de dato tipo ENTRADA”.

##### *2.6.1.1 Consideraciones adicionales*

- a) Excepción a la Regla 8: Todo mensaje que sólo inicia un escenario no será considerado como movimiento de dato tipo entrada.
- b) Como la recepción de un mensaje <<signal>> induce a la clase system enviar n mensajes a n líneas de vida con al menos un parámetro (Vea Figura 1), el número de grupos de datos involucrados en un mensaje <<signal>> es determinado por el número de mensajes inducidos por la clase system.



**Figura 1.** Fragmento de interacción tipo 1

- c) Como un mensaje de tipo <<connect>> se activa cuando un mensaje de tipo <<service>> tiene la necesidad de establecer o eliminar vínculos entre los objetos de las clases (Vea Figura 2), el número de grupos de datos involucrados en un mensaje <<signal>> es determinado por la clase receptora del mensaje <<service>> más las clases receptoras del mensaje <<connect>>.



**Figura 2.** Fragmento de interacción tipo 2

### 2.6.2 Tipo de Movimiento de dato: LECTURA

**Regla 9:** “Considere a cada mensaje etiquetado con el estereotipo <<query>> como un movimiento de dato tipo LECTURA”.

**Regla 10:** “Considere a cada condición de mensaje como un movimiento de dato tipo LECTURA”.

**Regla 11:** “Considere a cada precondition definida en la especificación de un caso de uso como un movimiento de dato tipo LECTURA”.

**Regla 12:** “Considere a cada condición de una relación extend como un movimiento de dato tipo LECTURA”.

**Regla 13:** “Considere a cada restricción de integridad como un movimiento de dato tipo LECTURA”.

#### *2.6.3 Tipo de Movimiento de dato: ESCRITURA*

**Regla 14:** “Considere a cada mensaje etiquetado con los estereotipos <<service/new>>, <<service/destroy>> o <<service/update>> como un movimiento de dato tipo ESCRITURA”.

#### *2.6.4 Tipo de Movimiento de dato: SALIDA*

**Regla 15:** “Considere a cada mensaje etiquetado con el estereotipo <<signal>> y con el valor output como un movimiento de dato tipo SALIDA”.

#### *2.7. Eliminación de movimientos de datos duplicados:*

**Regla 16:** “Los n-mensajes del mismo tipo relacionados entre el mismo objeto de la clase receptora y el mismo objeto de la clase origen son considerados como un único tipo de movimiento de dato”.

**Regla 17:** “El conjunto de movimientos de datos identificados en un bloque del diagrama de secuencia debe ser considerado una sola vez”.

**Regla 18:** “Las relaciones INCLUDE especificados dentro de un bloque del diagrama de secuencia deben ser considerados una sola vez”.

**Regla 19:** “Los mensajes de error y confirmación especificados dentro del alcance de un caso de uso deben ser considerados solamente una vez como un movimiento de dato tipo SALIDA”.

### **3. Aplicación de las reglas de asignación numérica**

#### **3.1. Aplicación de la Función de Medición:**

Cada instancia de un movimiento de dato identificado (entrada, lectura, escritura y salida) recibe un valor numérico de 1 Cfsu (Cosmic Functional Size Unit).

$f(x)=1Cfsu$  Donde “x” es un movimiento de dato que involucra a un grupo de dato.

#### **3.2. Aplicación de la Función de Agregación a Nivel de Proceso Funcional**

**Regla 20:** “El tamaño funcional de un caso de uso es igual a la suma de todos los movimientos de datos identificados en el mismo caso de uso”.

La regla es expresada en la siguiente ecuación:

$$\text{Tamaño}(\text{caso\_de\_uso}) = \sum_{i=1}^n f(x)_i$$

Dos reglas adicionales son definidas debido a las relaciones que se presentan entre casos de uso.

**Regla 21:** “El tamaño funcional de un caso de uso base extendido por otro conjunto de casos de uso secundarios es igual a la suma de los movimientos de datos identificados en cada caso de uso secundario más los movimientos de datos del caso de uso base”.

**Regla 22:** “El tamaño funcional de un Caso de Uso Base (CUB) que incluye otros Casos de Uso Secundarios (CUS) es igual a la suma de los

movimientos de datos identificados en cada caso de uso secundario más los movimientos de datos del caso de uso base”.

Estas dos reglas son expresadas en la siguiente ecuación:

$$\text{Tamaño}(CUB) = \sum_{i=1}^n \text{Tamaño}(CUS_i) + \text{Tamaño}_p(CUB)$$

### 3.3. Aplicación de la Función de Agregación a Nivel de Capa de Software

**Regla 23:** “El tamaño funcional de una capa de software es igual a la suma de los tamaños funcionales de todos los Casos de Uso Primarios (CUP)”. Esta regla es expresada en la siguiente ecuación:

$$\text{Tamaño}(Capa_1) = \sum_{i=1}^n \text{Tamaño}(CUP)_i$$

### 3.4. Aplicación de la Función de Agregación para requisitos funcionales que han sido modificados

El Tamaño funcional de los cambios realizados en un caso de uso es determinado por la sumatoria del tamaño funcional de los correspondientes movimientos de datos añadidos, eliminados o modificados.

$$\text{Tamaño}(CUM) = \sum_{h=1}^m \text{Tamaño}(MDA_h) + \sum_{i=1}^n \text{Tamaño}(MDM_i) + \sum_{j=1}^p \text{Tamaño}(MDE_j)$$

Donde: CUM significa Caso de Uso Modificado

MDA significa Movimientos de Datos Añadidos

MDM significa Movimientos de Datos Modificados

MDE significa Movimientos de Datos eliminados







Anexo	<b><i>Cuestionario: Conformidad de RmFFP</i></b>
<b>C</b>	

### ***Estructura***

El presente documento comprende de un listado de preguntas las cuales están agrupadas en 12 partes que corresponden al estándar ISO/IEC 19761:2003

- La parte 1 corresponde a la unidad del método COSMIC-FFP
- La parte 2 corresponde a la identificación del contexto de medición,
- Las siguientes 7 partes (3-9) corresponden a la identificación de los conceptos relevantes de COSMIC-FFP.
- La parte 10 corresponde a la obtención del tamaño funcional
- Las partes 11 y 12 son partes complementarias, las cuales se refieren a la personalización del método de medición en caso de haber sido necesario, y a la presentación de los resultados.

### ***Instrucciones***

Cada pregunta debe ser respondida en función de si satisface o no al requisito establecido por el estándar ISO/IEC 19761:2003 (su ubicación es indicado en “requisito correspondiente”). Además debe indicarse la ubicación de la información relevante que responde a la pregunta. En caso de no satisfacer a la pregunta o no encontrar información relevante, se debe justificar las razones.

## **Evaluación de conformidad**

### **1. Unidad**

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿La unidad del procedimiento de medición es expresada como un movimiento de dato?			5.a
¿Cfsu es el símbolo de unidad utilizado en el procedimiento de medición?			5.b

### **2. Determinación del propósito y alcance del tamaño funcional**

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición considera a la determinación del propósito como una actividad requerida?			6.2
¿El procedimiento de medición considera a la determinación del alcance como una actividad requerida?			6.2

### **3. Identificación de los requisitos funcionales**

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿Los requisitos funcionales son identificados dentro del alcance del procedimiento de medición bajo un punto de vista de medición?			6.3

#### 4. Identificación de las capas de software

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición permite la identificación de las capas de software?			6.4.1
¿El software en cada capa identificada entrega funcionalidad a sus usuarios?			6.4.2.a
¿El software en cada capa subordinada provee servicios funcionales a otras capas?			6.4.2.b

#### 5. Identificación de fronteras de software

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición permite la identificación de las fronteras de software?			6.5

#### 6. Identificación de procesos funcionales

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿ El procedimiento de medición permite la identificación de los procesos funcionales?			6.6
¿Estos procesos funcionales son inicializados por algún evento disparador?			6.6.a
¿Cada Proceso funcional comprende de al menos 2 movimientos de datos (entrada y salida o escritura)?			6.6.b
¿Cada proceso funcional pertenece únicamente a una capa?			6.6.c

## 7. Identificación de grupos de datos

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición permite la identificación de grupos de datos?			6.7
¿Estos grupos de datos representan una colección de atributos de datos?			6.7.a

## 8. Identificación de movimientos de datos

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición no considera algún tipo de movimiento diferente a entrada, salida, lectura o escritura?			6.8.a y 6.8.c
¿Los movimientos de datos son contados sólo una sola vez dentro de un proceso funcional?			6.8.b

## 9. Clasificación de movimientos de datos

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El procedimiento de medición permite la identificación de los movimientos de dato tipo entrada?			6.9.1
¿El procedimiento de medición permite la identificación de los movimientos de dato tipo salida?			6.9.2
¿El procedimiento de medición permite la identificación de los movimientos de dato tipo lectura?			6.9.3

¿El procedimiento de medición permite la identificación de los movimientos de dato tipo escritura?

		6.9.4
--	--	-------

## 10. *Obtención del tamaño funcional*

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿Cada movimiento de dato es asignado por el valor de 1 Cfsu?			6.10.1
¿Alguna función de agregación es definida para obtener el tamaño funcional a nivel de proceso funcional?			6.10.2
¿Alguna función de agregación es definida para obtener el tamaño funcional a nivel de capa de software?			6.10.3
¿Alguna función de agregación es definida para obtener el tamaño funcional de los requisitos funcionales que han sido modificados?			6.10.4

## 11. *Personalización local*

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿Alguna personalización ha sido necesaria para medir algunos requisitos funcionales que no han sido contemplados por el método COSMIC-FFP?			7

## 12. Reporte de la medición

<i>Pregunta de evaluación</i>	<i>Ubicación</i>	<i>Satisface?</i>	<i>Requisito correspondiente</i>
¿El resultado de la medición es etiquetado con el símbolo de la unidad y el valor numérico obtenido?			8.1 y 8.2
¿Existe alguna plantilla que contemple la información necesaria para documentar los resultados de medición?			8.3

### JUSTIFICACIONES

.....

.....

.....

.....

.....

DISTANCE proporciona procedimientos constructivos para modelar los atributos de interés del software y definir las correspondientes medidas. Este marco se basa en los conceptos de distancia y disimilitud. Los atributos del software son modelados como distancias conceptuales entre las entidades del software que se caracterizan y otras entidades del software que sirven como puntos de referencia. Estas distancias son medidas mediante funciones matemáticas que satisfacen el conjunto de axiomas del espacio de las métricas.

Antes de presentar este conjunto de axiomas es necesario previamente definir el concepto de *estructura de proximidad* y *estructura de proximidad aditiva segmentada*.

1.- Sea  $A$  un conjunto, la relación cuaternaria  $\bullet \geq$  sobre  $A$  es una relación binaria sobre  $A \times A$ ;

Tal que  $\forall a, b, c, d \in A: (a, b) \bullet \geq (c, d)$  si y solamente si la disimilaridad entre  $a$  y  $b$  es mayor entre  $c$  y  $d$ . La disimilaridad es un atributo de un par de elementos.

La relación de equivalencia  $\approx$  y la relación de orden estricto  $\bullet >$  son derivadas de la relación de orden de disimilaridad  $\bullet \geq$ , las que son presentadas a continuación.



La relación cuaternaria  $\approx$  sobre A es una relación equivalente sobre  $A \times A$  tal que :

$$\forall a, b, c, d \in A : (a, b) \approx (c, d) \Leftrightarrow (a, b) \bullet \geq (c, d) \text{ y } (c, d) \bullet \geq (a, b)$$

La relación cuaternaria  $\bullet >$  sobre A es una relación de orden estricta sobre  $A \times A$  tal que:

$$\forall a, b, c, d \in A : (a, b) \bullet > (c, d) \text{ y } \neg((c, d) \bullet \geq (a, b)).$$

$(A, \bullet \geq)$  es una **estructura de proximidad** si y solamente se cumple los siguientes axiomas

El orden de disimilaridad  $\bullet \geq$  es una relación de orden débil.

- $(a, b) \bullet \geq (a, a)$  siempre que  $a \neq b$  .....positividad
- $(a, a) \approx (b, b)$  .....minimicidad
- $(a, b) \approx (b, a)$  .....simetría

2.- Sea A un conjunto. Para  $a, b, c \in A$  y la relación ternaria  $\langle abc \rangle$  se tiene:

$$\forall a', b', c' \in A : (a, b) \bullet \geq (a', b') \text{ y } (a', c') \bullet \geq (a, c) \Rightarrow (b', c') \bullet \geq (b, c)$$

Una estructura relacional  $(A, \bullet \geq)$  es una **estructura de proximidad aditiva segmentada**, si y solamente si, se cumplen los siguientes axiomas:

- $(A, \bullet \geq)$  es una estructura de proximidad
- Dada dos distancias (a,b) y (c,d) podemos encontrar siempre un segmento aditivo a partir de "a" hacia "e" sobre (a,b) que es exactamente igual a la

distancia entre “c” y “d”. Si  $(a, b) \bullet \geq (c, d), \Rightarrow \exists e \in A$  tal que  $(a, e) \approx (c, d) \wedge \langle aeb \rangle$

El fin de este axioma es garantizar que (c,d) no puede ser infinitamente pequeño comparado con (a,b). Si  $c \neq d$ , existe  $e'_0, \dots, e'_n \in A$  tal que  $e'_0 = a, e'_n = b$  y  $(c, d) \bullet \geq (e'_{i-1}, e'_i), \forall i = 1, \dots, n$ .

**Definición.** Métrica y Espacio de Métrica

3) Sea A un conjunto. La función  $\delta$  es una métrica si y sólo si:

- A<sub>1</sub>.  $a, b \in A: \delta(a, b) \geq 0$  (no-negatividad)
- A<sub>2</sub>.  $a, b \in A: \delta(a, b) = 0 \Leftrightarrow a = b$  (identidad)
- A<sub>3</sub>.  $a, b \in A: \delta(a, b) = \delta(b, a)$  (simetría)
- A<sub>4</sub>.  $a, b, c \in A: \delta(a, b) \leq \delta(a, c) + \delta(c, b)$  (la desigualdad triangular)

4) Si  $\delta : A \times A \rightarrow \Re$  es una métrica, entonces  $(A, \delta)$  es un **espacio de métrica**.

Básicamente la aproximación para la medición del software basada en la distancia modela los atributos del software como distancias. Las métricas o las funciones de distancia se consideran como “medidas” de la distancia. Por lo tanto, puede utilizarse para definir medidas de los atributos del software que están modelados como distancias.



**INSTRUMENTO :**

**LEA POR FAVOR CADA PREGUNTA CUIDADOSAMENTE ANTES DE DAR SU RESPUESTA**

1. El proceso a seguir para aplicar el procedimiento de medición es complejo y difícil de seguir	0	0	0	0	0	El proceso a seguir para aplicar el procedimiento de medición es simple y fácil de seguir
2. Creo que este procedimiento reduciría el tiempo requerido para medir el tamaño funcional de aplicaciones orientado a objetos	0	0	0	0	0	Creo que este procedimiento aumentaría el tiempo requerido para medir el tamaño funcional de aplicaciones orientado a objetos
3. De manera general, el procedimiento de medición es difícil de utilizar	0	0	0	0	0	De manera general, el procedimiento de medición es fácil de utilizar
4. Las reglas de medición del procedimiento son claras y fáciles de entender	0	0	0	0	0	Las reglas de medición del procedimiento son confusas y difíciles de entender
5. De manera general, el procedimiento de medición es útil	0	0	0	0	0	De manera general, el procedimiento de medición NO es útil
6. El procedimiento de medición es difícil de aprender	0	0	0	0	0	El procedimiento de medición es fácil de aprender

7. Utilizaré este procedimiento si tengo que medir el tamaño funcional de aplicaciones orientado a objetos en el futuro	0	0	0	0	0	NO utilizaré este procedimiento si tengo que medir el tamaño funcional de aplicaciones orientado a objetos en el futuro
8. Pienso que este procedimiento NO mejoraría la precisión de las mediciones de las aplicaciones orientado a objetos	0	0	0	0	0	Pienso que este procedimiento mejoraría la precisión de las mediciones de las aplicaciones orientado a objetos
9. Me pareció difícil de aplicar el procedimiento de medición al caso de estudio	0	0	0	0	0	Me pareció fácil de aplicar el procedimiento de medición al caso de estudio
10. De manera general, pienso que este procedimiento NO proporcionó una manera eficaz de medir el tamaño funcional de aplicaciones orientado a objetos durante la fase de especificación de requisitos	0	0	0	0	0	De manera general, pienso que este procedimiento proporcionó una manera eficaz de medir el tamaño funcional de aplicaciones orientado a objetos durante la fase de especificación de requisitos
11. El uso de este procedimiento mejoraría mi rendimiento en la medición del tamaño funcional de aplicaciones orientado a objetos	0	0	0	0	0	El uso de este procedimiento NO mejoraría mi rendimiento en la medición del tamaño funcional de aplicaciones orientado a objetos
12. Pienso que sería fácil ser hábil usando este procedimiento	0	0	0	0	0	Pienso que sería difícil ser hábil usando este procedimiento
13. Tengo la intención de utilizar este procedimiento en el futuro	0	0	0	0	0	NO tengo la intención de utilizar este procedimiento en el futuro

Escriba por favor cualquier otro comentario que quieras hacer sobre el procedimiento de medición en el espacio de abajo

---

---

---

**DATOS EXPERIMENTALES :**

**Tabla B1.** Puntajes obtenidos al aplicar el cuestionario sobre el uso de RmFFP

Estudiantes	P1	P3	P4	P6	P9	P2	P5	P8	P10	P11	P7	P12	P13
1	5	5	5	5	4	4	4	3	3	4	5	4	4
2	4	4	3	5	3	3	4	4	4	2	3	2	3
3	5	5	4	5	5	5	4	4	3	4	5	4	4
4	4	4	4	4	3	4	4	4	4	4	3	5	3
5	3	4	4	4	4	2	3	3	3	4	3	4	3
6	4	5	4	5	5	4	3	3	3	3	3	5	3
7	5	5	5	5	5	2	4	4	5	4	4	5	4
8	4	4	4	4	4	3	5	5	4	5	5	4	4
9	4	5	4	4	4	4	4	5	5	4	5	4	4
10	5	5	4	5	5	5	5	4	4	4	3	4	3
11	3	5	2	4	2	4	3	3	3	3	3	4	3
12	5	4	5	5	4	5	3	4	4	4	5	5	3
13	3	4	4	3	4	4	5	4	4	4	4	4	4
14	3	3	4	4	4	4	3	4	4	4	3	4	4
15	5	5	4	5	4	1	4	4	4	4	3	5	3
16	5	4	5	4	5	4	5	4	4	4	3	4	2
17	3	4	3	3	3	4	4	4	4	4	3	4	3
18	4	4	5	5	4	4	4	4	4	5	4	4	4
19	4	5	5	5	4	4	5	4	4	5	4	5	3
20	4	4	3	4	4	4	5	5	4	5	4	3	4
21	5	5	5	5	5	4	5	4	4	5	4	5	5
22	4	4	5	5	5	4	4	4	4	4	4	4	3
23	4	4	4	5	4	3	4	4	4	4	3	4	3
24	4	4	2	4	4	3	4	4	4	4	4	5	4
25	5	4	5	5	2	5	5	3	3	4	4	5	3
26	5	4	3	5	4	5	4	3	5	4	4	5	3
27	5	5	4	5	4	3	4	3	4	4	3	3	3
28	5	5	5	4	3	4	4	4	4	4	4	5	3
29	5	5	4	4	4	5	4	4	4	5	3	5	4
30	4	4	4	4	3	3	3	5	4	2	4	4	4
31	5	5	5	5	5	4	4	4	4	3	5	5	4
32	5	5	4	5	4	4	4	4	4	4	4	5	3
33	4	4	2	3	5	5	5	4	5	3	5	5	3

**Tabla B2.** Análisis de datos para cada uno de los sujetos experimentales

Sujeto	Tamaño (Cfsu)	Tiempo de medición (hora)	Productividad (Cfsu/hora)	REP	FUP	UP	ITU
1	144	1,08	133,33	0,11	4,80	3,60	4,33
2	160	1,48	108,11	0,01	3,80	3,40	2,67
3	153	1,15	133,04	0,05	4,80	4,00	4,33
4	142	1,50	94,67	0,12	3,80	4,00	3,67
5	159	1,50	106,00	0,01	3,80	3,00	3,33
6	154	1,25	123,20	0,05	4,60	3,20	3,67
7	155	1,00	155,00	0,04	5,00	3,80	4,33
8	154	1,45	106,21	0,05	4,00	4,40	4,33
9	149	1,20	124,17	0,08	4,20	4,40	4,33
10	156	1,17	133,33	0,03	4,80	4,40	3,33
11	160	1,58	101,27	0,01	3,20	3,20	3,33
12	168	1,20	140,00	0,04	4,60	4,00	4,33
13	161	1,50	107,33	0,00	3,60	4,20	4,00
14	181	1,55	116,77	0,13	3,60	3,80	3,67
15	162	1,05	154,29	0,01	4,60	3,40	3,67
16	159	1,15	138,26	0,01	4,60	4,20	3,00
17	176	1,60	110,00	0,10	3,20	4,00	3,33
18	166	1,15	144,35	0,03	4,40	4,20	4,00
19	184	1,00	184,00	0,15	4,60	4,40	4,00
20	180	1,45	124,14	0,12	3,80	4,60	3,67
21	175	1,00	175,00	0,09	5,00	4,40	4,67
22	148	1,00	148,00	0,08	4,60	4,00	3,67
23	152	1,25	121,60	0,06	4,20	3,80	3,33
24	160	1,50	106,67	0,01	3,60	3,80	4,33
25	149	1,30	114,62	0,08	4,20	4,00	4,00
26	166	1,25	132,80	0,03	4,20	4,20	4,00
27	158	0,83	190,36	0,02	4,60	3,60	3,00
28	168	1,33	126,32	0,04	4,40	4,00	4,00
29	164	1,00	164,00	0,02	4,40	4,40	4,00
30	168	1,45	115,86	0,04	3,80	3,40	4,00
31	166	1,05	158,10	0,03	5,00	3,80	4,67
32	161	1,17	137,61	0,00	4,60	4,00	4,00
33	157	1,42	110,56	0,03	3,60	4,40	4,33





**INSTRUMENTO AJUSTADO:****LEA POR FAVOR CADA PREGUNTA CUIDADOSAMENTE ANTES DE DAR SU RESPUESTA**

1.	El proceso a seguir para aplicar el procedimiento de medición es complejo y difícil de seguir	0	0	0	0	0	El proceso a seguir para aplicar el procedimiento de medición es simple y fácil de seguir
2.	Creo que este procedimiento reduciría el tiempo requerido para medir el tamaño funcional de aplicaciones orientado a objetos	0	0	0	0	0	Creo que este procedimiento aumentaría el tiempo requerido para medir el tamaño funcional de aplicaciones orientado a objetos
3.	De manera general, el procedimiento de medición es difícil de utilizar	0	0	0	0	0	De manera general, el procedimiento de medición es fácil de utilizar
4.	Las reglas de medición del procedimiento son claras y fáciles de entender	0	0	0	0	0	Las reglas de medición del procedimiento son confusas y difíciles de entender
5.	De manera general, el procedimiento de medición es útil	0	0	0	0	0	De manera general, el procedimiento de medición NO es útil
6.	El procedimiento de medición es difícil de aprender	0	0	0	0	0	El procedimiento de medición es fácil de aprender
7.	Utilizaría este procedimiento si tengo que medir el tamaño funcional de especificaciones de requisitos en el futuro	0	0	0	0	0	NO utilizaría este procedimiento si tengo que medir el tamaño funcional de especificaciones de requisitos en el futuro

8.	Pienso que este procedimiento <u>NO</u> mejoraría la exactitud de las mediciones de las aplicaciones orientado a objetos	0	0	0	0	0	Pienso que este procedimiento mejoraría la exactitud de las mediciones de las aplicaciones orientado a objetos
9.	Me pareció difícil de aplicar el procedimiento de medición al caso de estudio	0	0	0	0	0	Me pareció fácil de aplicar el procedimiento de medición al caso de estudio
10	Si tuviera algún proyecto de software a mi cargo, utilizaría este procedimiento de medición para apoyar a la gestión de dicho proyecto.	0	0	0	0	0	Si tuviera algún proyecto de software a mi cargo, <u>NO</u> utilizaría este procedimiento de medición para apoyar a la gestión de dicho proyecto
11	De manera general, pienso que este procedimiento <u>NO</u> proporcionó una manera eficaz de medir el tamaño funcional de aplicaciones orientado a objetos durante la fase de especificación de requisitos	0	0	0	0	0	De manera general, pienso que este procedimiento proporcionó una manera eficaz de medir el tamaño funcional de aplicaciones orientado a objetos durante la fase de especificación de requisitos
12	El uso de este procedimiento mejoraría mi rendimiento en la medición del tamaño funcional de aplicaciones orientado a objetos	0	0	0	0	0	El uso de este procedimiento <u>NO</u> mejoraría mi rendimiento en la medición del tamaño funcional de aplicaciones orientado a objetos
13	Si mi compañía necesitaría algún procedimiento de medición de tamaño funcional, recomendaría el uso de RmFFP	0	0	0	0	0	Si mi compañía necesitaría algún procedimiento de medición de tamaño funcional, <u>NO</u> recomendaría el uso de RmFFP
14	Pienso que sería fácil ser hábil usando este procedimiento	0	0	0	0	0	Pienso que sería difícil ser hábil usando este procedimiento
15	Tengo la intención de utilizar este procedimiento en el futuro	0	0	0	0	0	<u>NO</u> tengo la intención de utilizar este procedimiento en el futuro

**Tabla C1.** Puntajes obtenidos al aplicar el cuestionario ajustado

Estudiantes	P1	P2	P3	P4	P6	P9	P12	P14	P5	P8	P11	P7	P10	P13	P15
1	3	3	5	5	3	4	2	5	4	2	2	1	1	3	2
2	5	4	5	3	2	5	4	5	4	3	4	5	2	5	4
3	4	4	4	3	4	4	3	5	4	3	2	3	3	4	1
4	4	4	5	4	3	4	4	5	5	4	4	4	4	5	4
5	4	4	5	5	5	4	3	5	4	3	3	4	4	3	4
6	4	3	4	3	4	4	4	4	3	4	4	4	4	4	3
7	3	4	4	3	3	3	4	4	4	4	4	3	4	4	3
8	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5
9	4	5	4	3	4	4	5	5	5	5	5	5	4	5	4
10	2	4	3	2	3	3	3	4	4	3	3	3	3	3	3
11	4	4	4	4	4	2	4	4	4	2	1	3	4	4	3