

**ESTIMACIÓN DEL TAMAÑO FUNCIONAL DEL  
SOFTWARE EN LA ELICITACIÓN DE  
REQUERIMIENTOS**

Mg. Mabel Angélica Bertolami

Director: Dr. Gustavo Rossi

Asesor científico: Lic. Alejandro Oliveros

Tesis presentada para obtener el grado de Doctor en Ciencias  
Informáticas



Facultad de Informática  
Universidad Nacional de La Plata

La Plata, República Argentina, abril de 2010



*A Marcelo y Patricio.*



## **Agradecimientos**

A Gustavo Rossi, por sus valiosos aportes y la confianza depositada en este trabajo.

A Alejandro Oliveros, por su inestimable ayuda, su paciencia, sus acertadas observaciones y sugerencias.

A la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco, que a través de su “Programa de formación en posgrado y afianzamiento de las actividades científicas y de vinculación tecnológica” me otorgó la ayuda económica que facilitó la realización de este doctorado.

A Elena Centeno y Raquel Olinik, por su colaboración en la elaboración y revisión del LEL y Escenarios de los casos utilizados para la evaluación de la propuesta presentada en esta tesis.

A Gabriela Oriana y Pamela Ritter, por su contribución en el desarrollo de la herramienta de soporte.

A los señores Jorge Brugger, Bruno Gelman, Jorge González, Claudio Mayorga, a la Facultad de Ingeniería y al Departamento de Informática de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco por facilitarme material de gran utilidad para la realización de este trabajo.

Mabel Bertolami

Abril de 2010



## Resumen

*La medición del tamaño del software es una actividad esencial para la estimación y planificación de un proyecto de desarrollo o las actividades de mantenimiento de software. Las técnicas de medición de Puntos Función permiten determinar el tamaño funcional del software a partir de diversos documentos del desarrollo, en particular nos interesan los asociados con los requerimientos funcionales del usuario. Para las etapas más tempranas del ciclo de vida, cuando los requerimientos funcionales no están completamente definidos o se requiere una estimación rápida, han sido propuestos múltiples enfoques de estimación de tamaño funcional. Sin embargo, se ha observado que algunas de estas técnicas requieren documentación con un nivel de detalle similar al usado para una medición con un método estándar, la que frecuentemente no está disponible en las fases iniciales de un proyecto de software. Otras propuestas no especifican claramente el artefacto necesario para ejecutar la estimación, no se encontró información sobre la exactitud de las estimaciones o la evaluación realizada para comprobar los resultados de su aplicación. Por último, son escasos los enfoques que se aplican a los productos de la Elicitación de Requerimientos.*

*Con el objetivo de ofrecer una alternativa para las fases más tempranas de un proyecto de software, en esta tesis se presenta el procedimiento Scenario Function Points (SFP) que permite estimar el tamaño funcional de un sistema de software a partir de un producto de la Elicitación de Requerimientos, específicamente, los escenarios generados de acuerdo al enfoque de Leite. SFP está basado en el método estándar IFPUG FPA y su definición está soportada por un conjunto de reglas que establecen la asociación entre los conceptos del Modelo de Escenarios y los propios de dicho método.*

*Para guiar la aplicación de SFP se diseñó un proceso y se elaboró su especificación formal. El objetivo de esta última es facilitar la capacitación y comunicación de los usuarios, favorecer la efectiva ejecución del proceso y propiciar la repetibilidad de los resultados. Además, el proceso es soportado por el software SFP Tool, que facilita y agiliza el proceso de estimación.*

*La evaluación del procedimiento SFP se realizó mediante un Caso de Estudio múltiple basado en tres sistemas implementados. Los resultados obtenidos son alentadores en cuanto a su capacidad como herramienta de estimación y al esfuerzo requerido por el proceso. En lo referido al primer aspecto, la exactitud de las estimaciones está en el orden de  $\pm 30\%$  (considerando a la medición con el método estándar IFPUG como el valor de referencia), que está dentro del rango aceptable para una estimación temprana. Además, en comparación con otras propuestas de estimación, el resultado es similar y en algunos casos mejor. Con respecto al esfuerzo, los resultados demuestran que es significativamente menor que el reportado en la literatura para una medición estándar.*





# Contenido

<b>Resumen</b> .....	i
<b>Lista de Tablas</b> .....	vii
<b>Lista de Figuras</b> .....	ix
<b>Lista de Acrónimos</b> .....	xi
<b>1. Introducción</b> .....	1
<b>1.1. Descripción del problema: estimación del tamaño de un proyecto de software</b> .....	1
<b>1.2. Motivación</b> .....	2
<b>1.3. Objetivos</b> .....	3
<b>1.4. Contribución</b> .....	5
<b>1.5. Publicaciones relacionadas con la tesis</b> .....	5
<b>1.5.1. Congresos internacionales</b> .....	5
<b>1.5.2. Congresos nacionales</b> .....	5
<b>1.5.3. Revistas con referato</b> .....	5
<b>1.6. Organización de la tesis</b> .....	6
<b>2. Marco conceptual</b> .....	7
<b>2.1. Léxico Extendido del Lenguaje y Escenarios</b> .....	7
<b>2.1.1. Léxico Extendido del Lenguaje</b> .....	7
<b>2.1.2. Escenarios</b> .....	9
<b>2.2. Medición del tamaño del software</b> .....	11
<b>2.2.1. Medidas técnicas</b> .....	12
<b>2.2.2. Medidas funcionales</b> .....	13
<b>2.2.3. El Método IFPUG FPA</b> .....	14
<b>2.2.4. El estándar ISO 14143 para la Medición del tamaño funcional</b> .....	15
<b>2.2.5. El Método IFPUG, el estándar ISO 20926 y su relación con ISO 14143-1</b> .....	18
<b>2.2.6. Usos y beneficios de la Medición del tamaño funcional</b> .....	19
<b>2.2.7. Modelos asociados a la Medición del tamaño funcional</b> .....	20
<b>2.3. Especificación de procesos software</b> .....	27
<b>2.3.1. Introducción</b> .....	27
<b>2.3.2. Criterios para la definición de procesos</b> .....	27
<b>2.4. Resumen</b> .....	29
<b>3. Estado del arte</b> .....	31

3.1. Introducción .....	31
3.2. Medición de Puntos Función.....	35
3.2.1. IFPUG FPA.....	35
3.2.2. Extensión de IFPUG FPA.....	37
3.2.3. Resumen sobre la evaluación de los Métodos de medición.....	42
3.3. Estimación de Puntos Función.....	43
3.3.1. Extrapolación de medidas.....	44
3.3.2. Estimación de la complejidad media.....	48
3.3.3. Early & Quick Function Points (E&QFP).....	51
3.3.4. Resumen sobre la evaluación de los Métodos de estimación.....	52
3.4. Panorama actual .....	53
3.5. Conclusiones.....	54
<b>4. SFP: Procedimiento de estimación de tamaño funcional de Escenarios..</b>	<b>59</b>
4.1. Diseño del Procedimiento SFP.....	59
4.1.1. Definición de los objetivos .....	59
4.1.2. Caracterización del concepto a medir.....	61
4.1.3. Diseño del metamodelo.....	70
4.1.4. Definición de las reglas de asignación numérica .....	73
4.2. Aplicación del Procedimiento SFP.....	77
4.2.1. El Proceso SFP .....	78
4.2.2. Un ejemplo de aplicación del proceso SFP.....	80
4.3. Resumen y Conclusiones.....	85
<b>5. Especificación del Proceso SFP.....</b>	<b>87</b>
5.1. Diseño de la plantilla para especificar el proceso SFP.....	87
5.2. Especificación del Proceso SFP.....	88
5.3. Medición del esfuerzo .....	97
5.4. Documentación de soporte.....	97
5.5. Soporte automatizado.....	98
5.6. Resumen.....	98
<b>6. Evaluación del Procedimiento SFP .....</b>	<b>101</b>
6.1. Evaluación de la aplicabilidad del procedimiento SFP.....	101
6.2. Evaluación de la exactitud de la estimación de tamaño funcional .....	102
6.2.1. Objetivo.....	102
6.2.2. Metodología de investigación.....	103
6.2.3. Selección de los casos.....	104

6.2.4. Descripción del proceso de evaluación .....	105
6.2.5. Caso de Estudio .....	106
6.2.6. Alcance de la evaluación .....	122
6.2.7. La incertidumbre de la estimación .....	123
6.2.8. Conclusiones del Caso de Estudio .....	124
6.2.9. Comparación con otras propuestas de estimación .....	125
6.3. Resumen y Conclusiones .....	127
<b>7. Conclusiones y Trabajos futuros.....</b>	<b>129</b>
7.1. Conclusiones .....	129
7.2. Resumen de las contribuciones.....	130
7.3. Trabajos futuros .....	130
<b>Referencias.....</b>	<b>133</b>
<b>A. Método IFPUG FPA Versión 4.1.1.....</b>	<b>143</b>
A.1. Punto de vista del usuario .....	143
A.2. Procedimiento de Medición de Puntos Función.....	143
A.2.1. Determinar el Tipo de medición de Puntos Función.....	143
A.2.2. Identificar el Alcance de la medición y el Límite de la aplicación.....	144
A.2.3. Calcular los FP no ajustados .....	144
A.2.4. Medir las funciones de datos .....	144
A.2.5. Medir las funciones transaccionales .....	146
A.2.6. Calcular la cuenta de FP no ajustados .....	150
<b>B. Un marco para especificar procesos software .....</b>	<b>151</b>
B.1. El marco operativo del SPF.....	151
B.2. Plantillas del SPF .....	152
<b>C. Formularios .....</b>	<b>155</b>
C.1. Plantillas de formularios .....	155
C.1.1. Normas para completar los formularios.....	156
<b>D. Caso SFP Tool .....</b>	<b>159</b>
D.1. LEL y Escenarios.....	159
D.2. Estimación SFP.....	176
D.3. Especificación de Requerimientos del Software .....	189
D.4. Medición IFPUG.....	194
D.4.1. Archivos lógicos .....	194
D.4.2. Transacciones .....	194
<b>E. Software SFP Tool.....</b>	<b>197</b>

<b>E.1. El software SFP Tool.....</b>	<b>197</b>
<b>E.2. Uso de SFP Tool.....</b>	<b>197</b>
<b>E.2.1. Iniciar una sesión de trabajo .....</b>	<b>197</b>
<b>E.2.2. Realizar una nueva medición.....</b>	<b>197</b>
<b>E.2.3. Abrir medición existente.....</b>	<b>203</b>
<b>E.2.4. Continuar una medición.....</b>	<b>203</b>
<b>E.2.5. Revisar una medición .....</b>	<b>203</b>
<b>E.2.6. Ayuda .....</b>	<b>203</b>
<b>E.2.7. Mediciones.....</b>	<b>204</b>
<b>E.3. Conclusiones.....</b>	<b>205</b>

## Lista de Tablas

Tabla 2.1. Modelo de Referencia del FPA (traducido de [NESMA 2002]) .....	23
Tabla 3.1. Clasificación de los enfoques de medición/estimación de Puntos Función.....	33
Tabla 3.2. Resumen de los enfoques de medición analizados .....	56
Tabla 3.3. Resumen de los enfoques de estimación analizados .....	57
Tabla 4.1. Definición de los objetivos del diseño del procedimiento de estimación.....	60
Tabla 4.2. Clasificación de los objetos del LEL.....	61
Tabla 4.3. Clasificación de los Recursos.....	63
Tabla 4.4. Asociación de conceptos entre SFP, IFPUG e ISO/IEC 14143-1 .....	68
Tabla 4.5. Instanciación del Modelo de Referencia para SFP .....	70
Tabla 4.6. Criterios para identificar episodios candidatos.....	71
Tabla 4.7. Resumen de las funciones realizadas por cada tipo de episodio (adaptada de [IFPUG 2000]) .....	73
Tabla 4.8. Complejidad y Contribución para EI (adaptada de [IFPUG 2000]).....	76
Tabla 4.9. Complejidad y Contribución para EO (adaptada de [IFPUG 2000]) .....	76
Tabla 4.10. Complejidad y Contribución de los recursos (adaptada de [IFPUG 2000])...	77
Tabla 4.11. Fragmento del Formulario 1: Identificación y clasificación de episodios.....	81
Tabla 4.12. Fragmento del Formulario 2: Complejidad de los episodios.....	82
Tabla 4.13. Fragmento del Formulario 3: Complejidad de los recursos .....	83
Tabla 4.14. Resultado de la estimación.....	84
Tabla 6.1. Resumen de los resultados de la estimación con SFP .....	101
Tabla 6.2. Resumen de la estimación del Caso A .....	109
Tabla 6.3. Resumen de la medición del Caso A.....	111
Tabla 6.4. Relación entre transacciones y archivos .....	112
Tabla 6.5. Resumen de la estimación/medición de tamaño funcional del Caso A.....	113
Tabla 6.6. Resumen de la estimación del Caso B.....	115
Tabla 6.7. Resumen de la medición del Caso B.....	116
Tabla 6.8. Ejemplos de valores de alcance, clase y tipo [Jones 2007].....	117
Tabla 6.9. Relación entre transacciones y archivos .....	118
Tabla 6.10. Resumen de la estimación/medición de tamaño funcional del Caso B.....	119
Tabla 6.11. Resumen de la estimación del Caso C.....	120
Tabla 6.12. Resumen de la medición del Caso C.....	121
Tabla 6.13. Resumen de la estimación/medición de tamaño funcional del Caso C.....	121
Tabla 6.14. Taxonomía para el tamaño del software [Jones 2008a].....	122
Tabla 6.15. Resumen de los resultados de Caso de estudio.....	124

Tabla 6.16. Resumen de los resultados.....	126
Tabla E.1. Comparación de la medición manual y automatizada .....	204

## Lista de Figuras

Figura 1.1. Paradoja del tamaño del software [Lelli y Meli 2005].....	2
Figura 1.2. Fase del proceso y artefacto de software propuestos para la estimación de Puntos Función (adaptado de [Lavazza y Valetto]).....	3
Figura 2.1. Modelo del LEL [Leite <i>et al.</i> 2000].....	8
Figura 2.2. Modelo de Escenarios [Leite <i>et al.</i> 2000].....	10
Figura 2.3. Evolución de los métodos de FSM [Functional Size Measurement 2008].....	13
Figura 2.4. La aplicación desde el punto de vista del usuario [IFPUG 2000].....	15
Figura 2.5. Procedimiento de Medición de Puntos Función [IFPUG 2000].....	15
Figura 2.6. Modelo de Referencia para la FSM (traducido de [NESMA 2002]).....	21
Figura 2.7. Proceso de medición del tamaño funcional [Fetcke <i>et al.</i> 2001].....	24
Figura 2.8. Proceso de Medición - Modelo de alto nivel (traducido de [Jacquet y Abran 1997]).....	25
Figura 3.1. Clasificación de los métodos de estimación (traducido de [Meli y Santillo 1999]).....	32
Figura 4.1. Diseño del método de medición (traducido de [Jacquet y Abran 1997]).....	59
Figura 4.2. Asociación con los conceptos de la abstracción orientada a los datos de Fetcke (adaptado de [Fetcke <i>et al.</i> 2001]).....	70
Figura 4.3. El proceso SFP comparado con otros modelos (adaptado de [Abrahão <i>et al.</i> 2004b]).....	79
Figura 6.1. Método de Casos de Estudio [Yin 2003].....	104
Figura 6.2. Proceso de evaluación de un FSMM [ISO 2002b].....	105
Figura 6.3. Estrategia de evaluación de SFP (adaptado de [ISO 2002b]).....	105
Figura 6.4. La aplicación A desde el punto de vista del usuario [IFPUG 2000].....	109
Figura 6.5. Pantalla “Nueva Unidad” del Programa Analítico.....	110
Figura 6.6. Diagrama de contexto del Sistema Gestión de un comercio [SGC 2005].....	114
Figura 6.7. La aplicación B desde el punto de vista del usuario [IFPUG 2000].....	115
Figura 6.8. Ventana “Crear Artículo”.....	116
Figura 6.9. La aplicación C desde el punto de vista del usuario [IFPUG 2000].....	121
Figura 6.10. Cono de incertidumbre [Boehm <i>et al.</i> 1998].....	124
Figura B.1. Marco operativo, componentes y definiciones [Olson <i>et al.</i> 1993].....	151
Figura B.2. Ejemplo de plantilla del SFP.....	153
Figura E.1. Opciones del menú Archivo.....	197
Figura E.2. Instrucciones para cargar el L&E.....	198
Figura E.3. Cuadro de diálogo para ingresar el nombre del autor.....	198
Figura E.4. Ventana del Formulario 1.....	198

Figura E.5. Ventana del Formulario 1 con episodios clasificados.....	199
Figura E.6. Visor de Reglas.....	199
Figura E.7. Ventana del Formulario 2 .....	200
Figura E.8. Opción Visor LEL y Escenarios .....	200
Figura E.9. Pantalla del Visor del LEL y Escenarios .....	201
Figura E.10. Opción Volver a Formularios .....	201
Figura E.11. Ventana del Formulario 3 .....	202
Figura E.12. Ventana del Formulario 4 .....	202
Figura E.13. Ventana del Formulario 5 .....	203
Figura E.14. Ventana del Formulario 2 con datos de la revisión.....	203
Figura E.15. Ventanas de ayuda .....	204



## Lista de Acrónimos

<b>BFC</b>	Base Functional Component / Componente Funcional Básico
<b>BMW</b>	Baseline Mentor Workbench
<b>CASE</b>	Computer-Aided Software Engineering
<b>CMM</b>	Capability Maturity Model
<b>COSMIC</b>	Common Software Measurement International Consortium
<b>COTS</b>	Common Off The Shelf
<b>DET</b>	Data Element Type / Tipo de Dato Elemental
<b>DFD</b>	Data Flow Diagram / Diagrama de Flujo de Datos
<b>E&amp;QFP</b>	Early & Quick Function Points
<b>EFPM</b>	Early Function Point Method
<b>EI</b>	External Input / Entrada Externa
<b>EIF</b>	External Interface File / Archivo de Interfaz Externo
<b>EO</b>	External Output / Salida Externa
<b>EQ</b>	External Inquiry / Consulta Externa
<b>ER</b>	Entity-Relationship / Entidad-Relación
<b>FISMA</b>	Finnish Software Measurement Association
<b>FP</b>	Function Points / Puntos Función
<b>FPA</b>	Function Points Analysis / Análisis de Puntos Función
<b>FPS</b>	Function Points Simplified
<b>FSM</b>	Functional Size Measurement / Medición del Tamaño Funcional
<b>FSMM</b>	Functional Size Measurement Method / Método de Medición del Tamaño Funcional
<b>FTR</b>	File Type Referenced / Tipo de Archivo Referenciado
<b>FUR</b>	Functional User Requirements / Requerimientos Funcionales del Usuario
<b>IFPUG</b>	International Function Points Users Group
<b>ILF</b>	Internal Logical File / Archivo Lógico Interno
<b>ISBSG</b>	International Software Benchmarking Standards Group
<b>ISO</b>	International Organization for Standardization
<b>IT</b>	Information Technology / Tecnología de la Información
<b>L&amp;E</b>	LEL y Escenarios
<b>LEL</b>	Léxico Extendido del Lenguaje
<b>MIS</b>	Management Information Systems / Sistemas de Información de Gestión
<b>NESMA</b>	Netherlands Software Metrics Association

<b>OO</b>	Object Oriented / Orientado a Objetos
<b>OOmFP</b>	OO-Method Function Points
<b>OOSE</b>	Object-Oriented Software Engineering
<b>RET</b>	Record Element Type / Tipo de Registro Elemental
<b>RTR</b>	Resource Type Referenced / Tipo de Recurso Referenciado
<b>RUR</b>	Reference User Requirements / Requerimientos de Usuario de Referencia
<b>SEI</b>	Software Engineering Institute
<b>SFP</b>	Scenario Function Points / Puntos Función de Escenarios
<b>SLOC</b>	Source Lines of Code / Líneas de código fuente
<b>SPF</b>	Software Process Framework
<b>SRS</b>	Software Requirement Specification / Especificación de Requerimientos del Software
<b>UAW</b>	Unadjusted Actor Weight
<b>UC</b>	Caso de Uso
<b>UCP</b>	Use Case Points
<b>UFP</b>	Unadjusted Function Points / Puntos Función no Ajustados
<b>UKSMA</b>	United Kingdom Metrics Association
<b>UML</b>	Unified Modeling Language
<b>UUCP</b>	Unadjusted Use Case Points
<b>UUCW</b>	Unadjusted Use Case Weight

# Capítulo 1

## Introducción

Las métricas ayudan a *comprender* y *evaluar* los productos, procesos y servicios del software, pueden proveer la información necesaria para *controlar* los recursos y procesos usados para producir software y se pueden usar para *predecir* atributos de entidades del software en el futuro [Humphrey 1989]. El principal propósito de las mediciones en una organización es comprender los procesos y productos de software para tomar mejores decisiones y lograr los objetivos [McAndrews 1993]. Por lo tanto, visto el papel fundamental que desempeñan las mediciones del software, es esencial disponer de ellas desde las primeras etapas del desarrollo.

Para la gestión eficiente y efectiva de los proyectos de desarrollo o mantenimiento de software es necesario disponer de una medida confiable del tamaño de los artefactos de software en cada fase y especialmente de aquéllos producidos en las primeras etapas del desarrollo [Meli *et al.* 2000], [Morasca 2001]. El tamaño es un factor clave para la estimación de esfuerzo, personal, cronograma y costo de un proyecto; la deficiente estimación de tamaño es una de las causas de los desfases en el costo y el cronograma de un proyecto [STSC 2000].

En 1979 Albrecht introdujo el Análisis de Puntos Función (FPA – Function Points Analysis) con el objetivo de medir la productividad en el desarrollo de software para Sistemas de Información de Gestión (MIS – Management Information Systems). El FPA es un método estándar para medir el desarrollo de software desde el punto de vista del usuario, es independiente del lenguaje de programación, metodología de desarrollo y tecnología usados para desarrollar la aplicación [IFPUG 2000]. Esta técnica, ampliamente usada en la industria, puede ser aplicada desde la fase de especificación de requerimientos y durante las siguientes fases del desarrollo. Sin embargo, hay dos situaciones en que no es posible el conteo de Puntos Función (FP – Function Points): la primera ocurre en las fases más tempranas, antes de la definición de los requerimientos, en las cuales no se puede aplicar el método estándar; la segunda, cuando la documentación, el tiempo o los recursos requeridos no están disponibles. En ambos casos se debe usar un método de estimación de los FP [Meli y Santillo 1999].

### 1.1. Descripción del problema: estimación del tamaño de un proyecto de software

La estimación de un proyecto de software presenta un fenómeno paradójico: en las primeras etapas del ciclo de vida del software, cuando la estimación es requerida (por ejemplo, durante el estudio de factibilidad), el tamaño funcional del software es muy difícil de estimar, mientras que cuando es posible estimarlo con la mayor exactitud (cuando el producto final está listo), ya no es necesario para propósitos de planificación (por ejemplo, para predecir el esfuerzo y duración del proyecto) [Meli *et al.* 2000], [Lelli y Meli 2005]. Esta situación es descripta como la paradoja del tamaño y es presentada en la Figura 1.1.

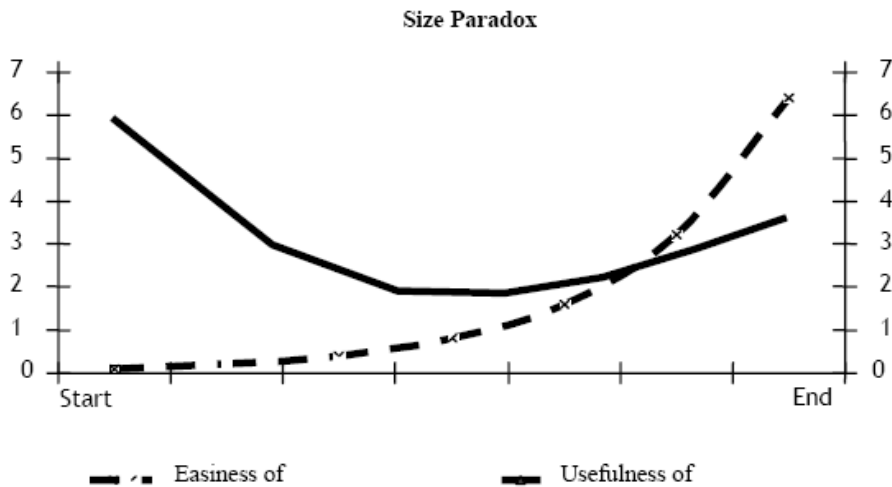


Figura 1.1. Paradoja del tamaño del software [Lelli y Meli 2005]

En las etapas más tempranas de un proyecto, a veces, lo único disponible es una descripción verbal o un bosquejo. Sin embargo, la falta de una especificación formal no debería ser un obstáculo para la estimación inicial de un proyecto [Peters 1999]. Ahora bien, ¿cuál es el mínimo de información necesaria para hacer una estimación de tamaño? Con relación a esta cuestión, la experiencia demuestra que las estimaciones muy tempranas son propensas a errores debido al escaso conocimiento disponible sobre el software a construir, pero, ¿cuál es el error aceptable para una estimación temprana? ¿es suficiente con esa estimación inicial? En el contexto de las estimaciones, las buenas prácticas indican que, aún cuando el error de una estimación estuviera dentro del rango admisible, es esencial comunicar el nivel de incertidumbre de la estimación y repetir la estimación del proyecto en cuanto se disponga de más información.

Existen múltiples propuestas para estimar los FP, sin embargo, se han observado los siguientes problemas:

- las técnicas basadas en Especificaciones de Requerimientos (SRS – Software Requirement Specification) requieren un nivel de detalle en la documentación que frecuentemente no está disponible en las fases más tempranas de un proyecto de software.
- algunas técnicas basadas en modelos estadísticos formulados a partir de datos históricos de proyectos requieren, como mínimo, el modelo de datos del sistema.
- algunas propuestas no especifican la exactitud de las estimaciones; en otros casos, el error de las estimaciones es significativo.
- hay escasos enfoques aplicables a los artefactos producidos en la Elicitación de Requerimientos.

## 1.2. Motivación

El Léxico Extendido del Lenguaje (LEL) y escenarios [Hadad *et al.* 1999] es una técnica de Elicitación de Requerimientos que utiliza descripciones en lenguaje natural. Como los escenarios se usan para entender la aplicación y su funcionalidad [Leite *et al.* 2000], desde el año 2001 se ha investigado la factibilidad de aplicar el FPA sobre estos escenarios. Con un fin exploratorio y como una primera aproximación al problema, en

2003 fue presentado un enfoque [Bertolami 2003] basado en el método MkII FPA<sup>1</sup> [Symons 1991]. Esta experiencia permitió vislumbrar la viabilidad del conteo de FP en los escenarios. Sin embargo, dicha propuesta presenta algunos aspectos objetables: en primer lugar, el uso de MkII puede ser calificado como extemporáneo, puesto que ya no es usado en la actualidad, además, no ha sido actualizado desde 1988 e incluso su propio autor, Symons, quien participó del consorcio internacional que desarrolló el nuevo método COSMIC-FFP [COSMIC 2003] destaca las ventajas de este último frente a MkII [CAI 2006]; en segundo lugar, el enfoque no cuenta con un proceso formalmente definido y documentado; por último y más importante aún, las estimaciones no fueron testeadas frente a la medición con el método estándar y sobre casos reales. En ese contexto, se orientaron los esfuerzos hacia la definición de un enfoque superador de tales problemas.

Concretamente, en esta tesis se intenta dar respuesta a las siguientes cuestiones:

1. ¿es posible estimar el tamaño funcional de un sistema de software en base a los escenarios generados en la etapa de Elicitación de Requerimientos?
2. ¿es consistente la estimación del tamaño funcional basada en escenarios con las que se obtienen en las etapas posteriores del proceso de desarrollo?
3. ¿es comparable la exactitud de la estimación de tamaño funcional basada en escenarios con la de otras propuestas?
4. ¿es posible resolver el problema mediante un proceso eficiente y con un mínimo esfuerzo?

El esquema de la Figura 1.2 presenta la fase del proceso de desarrollo en que se propone estimar el tamaño funcional y el artefacto de software usado como documento fuente.

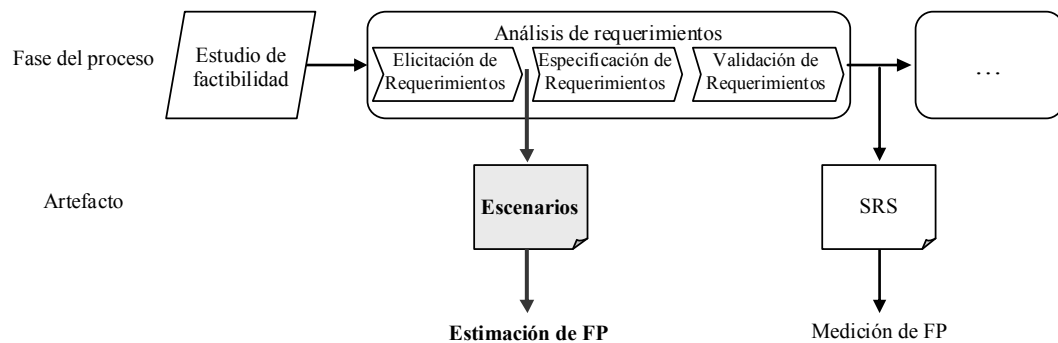


Figura 1.2. Fase del proceso y artefacto de software propuestos para la estimación de Puntos Función (adaptado de [Lavazza y Valetto]).

### 1.3. Objetivos

En este trabajo de tesis se propone un nuevo enfoque para la estimación temprana del tamaño funcional del software. Específicamente, el objetivo es presentar un procedimiento para estimar el tamaño funcional de un sistema de software a partir de los escenarios generados en el marco del enfoque de Leite (2000). Este objetivo general comprende la construcción, la aplicación y la evaluación del procedimiento de estimación [Jacquet y Abran 1997]. Para concretar estas actividades y considerando las preguntas formuladas en la sección anterior, se establecieron los siguientes objetivos específicos:

<sup>1</sup> También denominado Mark II FPA.

- diseñar el procedimiento de estimación: definir los objetivos del procedimiento de estimación, determinar los componentes de los escenarios que contribuyen al tamaño funcional, definir las reglas para identificarlos y clasificarlos y las reglas de asignación numérica.
- proveer un proceso bien definido para ejecutar la estimación: definir los pasos que se deben seguir para calcular el tamaño funcional y elaborar la especificación formal del proceso.
- evaluar el procedimiento de estimación: aplicar una estrategia para testear la exactitud de la estimación y el esfuerzo requerido por el proceso.

Para el logro del objetivo se diseñó el procedimiento Scenario Function Points (SFP – Puntos Función de Escenarios). Este procedimiento de estimación del tamaño funcional de los escenarios está basado en el método IFPUG (2000) y fue desarrollado siguiendo los lineamientos del estándar ISO/IEC 14143-1 [ISO 1998] para la Medición del Tamaño Funcional. La adopción del método IFPUG como base para el desarrollo de SFP estuvo determinada por su gran utilización en la industria. Al respecto, vale destacar la respuesta del propio Symons al ser consultado sobre las estadísticas de uso de COSMIC: “*I do not know of any statistics, but I am quite certain that less than 1% of all IT organizations use any type of functional size measure. Of those who do, I would say that 99% of them are probably using the IFPUG method*”<sup>2</sup> [CAI 2006].

Para su diseño de SFP se aplicó el Modelo de Proceso para los Métodos de Medición del software [Jacquet y Abran 1997]. Para la especificación del proceso se utilizó una plantilla (*template*) diseñada de acuerdo al Software Process Framework [Olson *et al.* 1993]. La evaluación se realizó mediante un Caso de Estudio múltiple - según la clasificación propuesta por Yin (2003) - formado por tres casos sobre los que se aplicaron el procedimiento SFP y el método IFPUG y luego se determinó la exactitud de la estimación basada en los escenarios frente a la medición de los sistemas implementados.

La denominación “procedimiento de estimación” fue establecida considerando:

- el significado del verbo *estimar* (Oxford Advanced Learner’s Dictionary [Hornby 2003]): hacerse una idea del costo, tamaño, valor, etc. de algo, pero sin calcularlo exactamente. Específicamente, la propuesta presentada en esta tesis permite establecer una aproximación del tamaño del software. Esto es debido a que los escenarios no cuentan con un nivel de detalle equivalente a la SRS, comúnmente requerida por los métodos estándar para producir un resultado cuya exactitud es aceptada y respaldada por un gran número de aplicaciones en la industria a través de muchos años.
- la definición de *procedimiento de medición* (ISO International Vocabulary of Basic and General Terms in Metrology [ISO 1993]): descripción detallada de una medición de acuerdo a uno o más principios de medición y a un método de medición dado. Un *método de medición* es una descripción genérica de una secuencia lógica de operaciones usadas en una medición. El procedimiento de medición usualmente está documentado con suficiente detalle como para permitir que un operador ejecute una medición. En particular, el procedimiento aquí presentado adhiere a los conceptos básicos del método estándar IFPUG (2000) y consta de un proceso definido que establece los pasos para su aplicación.

---

<sup>2</sup> No conozco ninguna estadística, pero estoy bastante seguro de que menos del 1% de todas las organizaciones de IT (Tecnología de la Información – Information Technology) usan algún tipo de medida del tamaño funcional. De aquéllas que sí utilizan alguna de estas medidas, diría que el 99% de ellas probablemente está usando el método IFPUG [CAI 2006]. (traducción propia).

#### 1.4. Contribución

El trabajo realizado en función del objetivo propuesto permitió responder satisfactoriamente a los interrogantes planteados en la sección 1.2, por lo tanto, la principal contribución al campo de la estimación del tamaño del software es la siguiente: una propuesta para estimar el tamaño funcional de los escenarios generados en la fase de Elicitación de Requerimientos, cuyas estimaciones tienen una exactitud similar a la de otros enfoques, requiere un mínimo esfuerzo y puede ser aplicada en proyectos de mejoras o de reemplazo de un sistema carente de documentación.

#### 1.5. Publicaciones relacionadas con la tesis

A continuación se enumeran las publicaciones vinculadas con el tema de la tesis en las que participó el autor de esta tesis en calidad de autor o coautor.

##### 1.5.1. Congresos internacionales

- Bertolami, M., Oliveros, A., “Análisis de Puntos Función en la elicitación de requerimientos”, *Proceedings 6th Workshop on Requirements Engineering*, Piracicaba, Brasil, 2003, pp. 32-47.
- Bertolami, M., Oliveros, A., “Estimación del tamaño funcional en la Elicitación de Requerimientos”, *Proceedings 7th Workshop on Requirements Engineering*, Tandil, Argentina, 2004, pp. 1-12.
- Bertolami, M., Oliveros, A., “Proceso de medición de funcionalidad en la Elicitación de Requerimientos”, *Proceedings 7º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software*, IDEAS2004, Arequipa, Perú, 2004, pp. 91-102.
- Bertolami, M., Oliveros, A., “SFP: Un Procedimiento de Estimación de Puntos Función de Escenarios”, *Proceedings 9th Workshop on Requirements Engineering*, Rio de Janeiro, Brasil, 2006, pp. 101-108.<sup>3</sup>
- Bertolami, M., Oliveros, A., “Evaluación Empírica de la Estimación de Tamaño Funcional de Escenarios”, *Proceedings 11th Workshop of Requirements Engineering*, Barcelona, España, 2008, pp. 25-33.<sup>3</sup>

##### 1.5.2. Congresos nacionales

- Bertolami, M., Oliveros, A., “Medición de Puntos Función sobre especificaciones en UML, una evaluación experimental”, *Congreso Argentino de Ciencias de la Computación (CACIC 2008)*, Universidad Nacional de Chilecito, La Rioja, Argentina, 2008.<sup>3</sup>
- Ritter, P., Bertolami, M., Oriana, G., “SFP Tool: una Herramienta para Medir Puntos Función”, *Workshop de Investigadores de Ciencias de la Computación (WICC 2009)*, Universidad Nacional de San Juan, San Juan, Argentina, 2009.<sup>3</sup>

##### 1.5.3. Revistas con referato

- Bertolami, M., Oliveros, A., “Estimate of the Functional Size in the Requirements Elicitation”, *Journal of Computer Science & Technology*, Special Issue on Software Requirements Engineering, Vol. 5 - No. 2 - Julio 2005 - ISSN 1666-6038, 2005.

---

<sup>3</sup> El contenido o los resultados presentados en este artículo están incluidos en esta tesis.

### 1.6. Organización de la tesis

Esta tesis está estructurada del siguiente modo:

En el **Capítulo 2** se presentan los fundamentos teóricos, propuestas y estándares considerados para el diseño del procedimiento SFP y la especificación del proceso.

En el **Capítulo 3** se presenta el estado del arte en el campo de la medición y estimación del tamaño funcional de los artefactos generados en las etapas más tempranas del desarrollo de software. Esta revisión en la literatura está centrada en las técnicas basadas en el Análisis de Puntos Función y sus extensiones al software orientado a diferentes dominios. Las técnicas son evaluadas de acuerdo a un conjunto de criterios previamente establecidos.

En el **Capítulo 4** se presenta el diseño de SFP en base al Modelo de Proceso de Jacquet y Abran. Se describe el análisis realizado para asociar los conceptos del Modelo de escenarios con los del Método IFPUG considerando el marco de definiciones del estándar ISO 14143-1, definir las reglas y la función que permite determinar el tamaño funcional. Por último se presenta el proceso SFP, la comparación con otros procesos estándar y un ejemplo de la aplicación.

En el **Capítulo 5** se presenta la especificación del proceso SFP que está basada en el Software Process Framework propuesto por Olson *et al.* (1993).

En el **Capítulo 6** se presenta la evaluación del procedimiento SFP. En primer lugar se aplica SFP sobre un conjunto de casos con el objetivo de evaluar la utilidad de las reglas, la coherencia del proceso y el esfuerzo. En segundo lugar se desarrolla un Caso de Estudio múltiple que apunta a verificar la exactitud de la estimación de SFP frente a la medición basada en IFPUG y otras propuestas de estimación.

En el **Capítulo 7** se presentan las conclusiones y contribuciones de la propuesta presentada y los trabajos futuros.

En el **Apéndice A** se resumen los principales conceptos del método IFPUG FPA.

En el **Apéndice B** se resumen los conceptos del marco para especificar procesos software propuesto por Olson *et al.* (2003).

En el **Apéndice C** se presentan los formularios propuestos para recolectar la información generada durante el proceso de estimación.

En el **Apéndice D** se presenta un caso completo, incluyendo el LEL y Escenarios, los formularios con el detalle de la estimación con SFP, la Especificación de Requerimientos del Software y la medición con el método IFPUG basada en dicha especificación.

En el **Apéndice E** se presentan los detalles de la herramienta que soporta el proceso SFP.



## Capítulo 2

### Marco conceptual

En este capítulo se introducen los conceptos, definiciones y modelos usados como soporte para el diseño y aplicación del procedimiento Scenario Function Points (SFP) y para la especificación del proceso SFP.

#### 2.1. Léxico Extendido del Lenguaje y Escenarios

Una cuestión fundamental en el desarrollo de software es establecer una buena comunicación entre los distintos participantes. Uno de los obstáculos para la comunicación es el uso de diferentes léxicos por parte del ingeniero de requerimientos y el usuario. Por lo tanto, el uso de un vocabulario común, en particular el uso del lenguaje propio del usuario, mejora considerablemente esta comunicación [Hadad *et al.* 1999].

El Léxico Extendido del Lenguaje (LEL) y los Escenarios son herramientas para la Elicitación de Requerimientos y pueden ser usadas a lo largo del ciclo completo de desarrollo de software [Leite *et al.* 1997]. Este enfoque presenta una metodología basada en el uso de LEL para registrar el vocabulario del macrosistema y escenarios para modelar el comportamiento. Ambas herramientas presentan la ventaja de utilizar el lenguaje natural para las descripciones, lo que favorece la comunicación y validación con el usuario [Hadad *et al.* 1999], [Leite *et al.* 2000].

##### 2.1.1. Léxico Extendido del Lenguaje

El Léxico Extendido del Lenguaje es un metamodelo diseñado para facilitar la elicitación y representación del lenguaje usado en la aplicación [Leite *et al.* 2000]. El LEL tiene como finalidad la comprensión del vocabulario de la aplicación, sin considerar en esta etapa la comprensión del problema. Se propone construir el LEL como primer paso en la fase de Elicitación de Requerimientos, tratando de conocer el vocabulario que utiliza el usuario en su mundo real. El principal propósito de LEL es capturar el vocabulario y la semántica de la aplicación, posponiendo la comprensión de la funcionalidad de la aplicación [Leite *et al.* 2000]. Una vez que se está familiarizado con ese léxico, la comunicación con el usuario y la comprensión del problema tendrá un obstáculo menos: el vocabulario [Hadad *et al.* 1999].

El LEL es una representación hipertextual de los símbolos del lenguaje del cliente en el contexto de la aplicación. Cada símbolo representa una palabra o frase, frecuentemente las más repetidas por el cliente o de importancia relevante para el sistema, excluyendo aquellas palabras o frases que son de conocimiento general, las que son demasiado específicas y las que tienen un significado muy amplio [Hadad *et al.* 1997].

Para conocer el vocabulario propio del dominio del sistema, el ingeniero de software planifica alguna estrategia para recolectar información, principalmente entrevistas con el usuario, las que eventualmente pueden ser reemplazadas por documentos descriptivos de la aplicación. En esta etapa se registran las frases o palabras usadas con más frecuencia o que tienen un significado especial. Como resultado de esta fase se obtiene una *lista candidata* formada por *símbolos* o *entradas candidatas*, que será utilizada como base para realizar nuevas entrevistas con el cliente.

Los símbolos se clasifican en las categorías generales *Sujeto* (entidad activa que realiza actividades en el dominio de la aplicación), *Objeto* (entidad pasiva a la que se le

aplican acciones en el dominio de la aplicación, sin realizar acciones por sí misma), *Verbo* (actividad o acción que ocurre en el dominio de la aplicación) y *Estado* (condición o situación en la cual sujetos, objetos o verbos del dominio de la aplicación están o pueden estar en un momento dado). En caso de ser necesario puede considerarse refinar algunos tipos de la clasificación general o crear algunos nuevos, ajustando la clasificación al dominio específico de la aplicación [Hadad 2008].

A partir del conocimiento obtenido, cada símbolo se describe en términos de *nombre*, *noción* e *impacto* (la Figura 2.1 presenta el Modelo del LEL). El *nombre* identifica el símbolo, la *noción* denota lo que significa el símbolo y el *impacto* (puede no existir) cómo repercute en el sistema. Al describir *noción* e *impacto* deben tenerse en cuenta dos principios: *circularidad* que implica maximizar el uso de los símbolos en el significado de otros símbolos y *vocabulario mínimo* que significa minimizar el uso de símbolos externos al lenguaje de la aplicación. La imposición de ambos principios da como resultado la formación de una red de elementos vinculados entre sí, que se representa como un documento de hipertexto [Leite *et al.* 1997]. Para conocer todo el vocabulario del dominio se puede navegar a través del hipertexto [Hadad *et al.* 1999].

La *lista candidata* es validada con el cliente con el fin de rectificar o ratificar las descripciones. Durante esta etapa puede haber modificaciones en los símbolos, incorporación de nuevos símbolos y eliminación de otros. Este proceso iterativo de interacción con el cliente permite refinar la lista hasta obtener la lista definitiva. Forma parte de esta etapa de validación el desarrollo de escenarios.

<p><b>LEL:</b> representación de los símbolos en el lenguaje del dominio de la aplicación.          Sintaxis:  <math>\{\text{Símbolo}\}_1^N</math></p> <p><b>Símbolo:</b> entrada del léxico que tiene un significado especial en el dominio de la aplicación.          Sintaxis:  <math>\{\text{Nombre}\}_1^N + \{\text{Noción}\}_1^N + \{\text{Impacto}\}_1^N</math></p> <p><b>Nombre:</b> identificación del símbolo. Más de uno representa sinónimo.          Sintaxis:          Palabra   Frase</p> <p><b>Noción:</b> denotación del símbolo. Debe ser expresada usando referencias a otros símbolos y usando un vocabulario mínimo.          Sintaxis:          Sentencia</p> <p><b>Impacto:</b> connotación del símbolo. Debe ser expresada usando referencias a otros símbolos y usando un vocabulario mínimo.          Sintaxis:          Sentencia</p> <p>donde Sentencia está formada por Símbolos y No-Símbolos, los últimos pertenecen al vocabulario mínimo.</p> <p>+ significa composición, {x} significa cero o más ocurrencias de x,   significa <b>or</b></p>
---

Figura 2.1. Modelo del LEL [Leite *et al.* 2000]

El siguiente es un ejemplo<sup>4</sup> de un símbolo del LEL (el texto subrayado representa un símbolo del LEL) editado con la herramienta Baseline Mentor Workbench (BMW) [Antonelli *et al.* 1999].

<p><u>no show</u></p> <ul style="list-style-type: none"> <li>• Notion           <ul style="list-style-type: none"> <li>– Es la baja de la <u>solicitud de reserva</u> por no presentación del <u>pasajero</u>.</li> <li>– Es consignado por el <u>repcionista</u>.</li> <li>– Se hace en la <u>recepción</u>.</li> </ul> </li> <li>• Behavioral responses           <ul style="list-style-type: none"> <li>– <b>If</b> el <u>pasajero</u> no se presenta entre las 12 hrs. la fecha indicada en la <u>solicitud de reserva</u> y las 06 hrs. del día siguiente <b>then</b> se elimina la <u>solicitud de reserva</u> en la <u>planilla de reservas</u>.</li> <li>– Se actualiza la <u>disponibilidad de habitaciones</u> en la <u>planilla de ocupación de habitaciones</u>.</li> </ul> </li> </ul>
---

### 2.1.2. Escenarios

Los enfoques basados en escenarios son ampliamente usados en la comunidad de los Sistemas de Información [Potts 1998], [Sutcliffe 1997], [Weidenhaupt *et al.* 1998]. Los escenarios son descripciones parciales del comportamiento del sistema y su entorno en situaciones particulares. Pueden usarse para representar el comportamiento de sistemas existentes, aunque más comúnmente son usados durante la planificación y diseño de nuevos sistemas. Los escenarios desempeñan un importante rol a través del proceso de desarrollo de Sistemas de Información, pues son fundamentales para las siguientes actividades: describir y aclarar las propiedades relevantes del dominio de la aplicación, descubrir los requerimientos del sistema, evaluar alternativas de diseño, validar diseños [Benner *et al.* 1993], facilitar la comunicación entre los participantes, favorecer el acuerdo entre los requerimientos del usuario y el soporte provisto por el sistema [Sutcliffe 1997], facilitar la comprensión de las prácticas de trabajo y los procesos del negocio, representar requerimientos de comportamiento, validar requerimientos, generar y validar diseños orientados a objetos, generar y revisar casos de prueba de sistemas [Potts 1998].

En el contexto de esta tesis se considerará la definición de Leite. Los escenarios son usados para entender la aplicación y su funcionalidad: cada escenario describe una situación específica de la aplicación centrando la atención en su comportamiento. Los escenarios se representan en lenguaje natural, son derivados desde el LEL mediante heurísticas específicas, evolucionan durante el proceso de construcción del software y se representan de manera estructurada [Leite *et al.* 1997], [Leite *et al.* 2000].

LEL y Escenarios (L&E) constituyen el producto resultante de la fase de Elicitación de Requerimientos. Los escenarios no son especificaciones ni requerimientos, son descripciones auxiliares para el proceso de definición de requerimientos. Ellos proveen una fuente de conocimiento donde pueden ser encontrados los requerimientos y en los que pueden basarse las especificaciones [Leite *et al.* 2000].

Un escenario se compone de: *nombre, objetivo, contexto, recursos, actores y episodios* (la Figura 2.2 presenta el Modelo de Escenarios).

<sup>4</sup> Los ejemplos corresponden al caso Recepción del Hotel [Bertolami y Centeno 2001].

**Escenario:** descripción de una situación en el dominio de la aplicación.  
 Sintaxis: Título + Objetivo + Contexto + {Recursos}<sub>1</sub><sup>N</sup> + {Actores}<sub>1</sub><sup>N</sup> + {Episodios}<sub>2</sub><sup>N</sup> + {Excepciones}

**Título:** identificación del escenario. En el caso de un subescenario, el título es el mismo que la sentencia del episodio, sin las restricciones.  
 Sintaxis: Frase | ([Actor | Recurso] + Verbo + Predicado)

**Objetivo:** meta a ser alcanzada en el dominio de la aplicación. El escenario describe el logro del objetivo.  
 Sintaxis: [Actor | Recurso] + Verbo + Predicado

**Contexto:** compuesto al menos por uno de los siguientes subcomponentes:  
 Ubicación Geográfica: ubicación física del escenario.  
 Ubicación Temporal: especificación de tiempo para el desarrollo del escenario.  
 Precondición: estado inicial del escenario.  
 Sintaxis: {Ubicación Geográfica} + {Ubicación Temporal} + {Precondición}  
 donde Ubicación Geográfica es:  
     Frase + {Restricción}  
 donde Ubicación Temporal es:  
     Frase + {Restricción}  
 donde Precondición es:  
     [Sujeto | Actor | Recurso] + Verbo + Predicado + {Restricción}

**Recursos:** elementos físicos relevantes o información que debe estar disponible en el escenario.  
 Sintaxis: Nombre + {Restricción}

**Actores:** personas, dispositivos u organización que tienen un rol en el escenario.  
 Sintaxis: Nombre

**Episodios:** conjunto de acciones que detallan el escenario y proveen su comportamiento. Un episodio también puede ser descrito como un escenario.  
 Sintaxis (usando BNF parcial):  
 <episodios> ::= <serie de grupos> | <serie de episodios>  
 <serie de grupos> ::= <grupo> <grupo> | <grupo no secuencial> | <serie de grupos> <grupo>  
 <grupo> ::= <grupo secuencial> | <grupo no secuencial>  
 <grupo secuencial> ::= <sentencia básica> | <grupo secuencial> <sentencia básica>  
 <grupo no secuencial> ::= # <serie de episodios> #  
 <serie de episodios> ::= <sentencia básica> <sentencia básica> |  
     <serie de episodios> <sentencia básica>  
 <sentencia básica> ::= <sentencia simple> | <sentencia condicional> | <sentencia opcional>  
 <sentencia simple> ::= <sentencia episodio> CR  
 <sentencia condicional> ::= **IF** <condición> **THEN** <sentencia episodio> CR  
 <sentencia opcional> ::= [ <sentencia episodio> ] CR  
 donde <sentencia episodio> es descripta:  
     (( [Actor | Recurso] + Verbo + Predicado ) | ([Actor | Recurso] + [Verbo] + Título) ) + {Restricción}

**Excepciones:** usualmente reflejan la falta o mal funcionamiento de un recurso necesario. Una excepción impide el cumplimiento del objetivo del escenario. El tratamiento de la excepción puede ser expresado mediante otro escenario.  
 Sintaxis: Causa [(Solución)]  
 donde Causa es:  
     Frase | ([Sujeto | Actor | Recurso] + Verbo + Predicado)  
 donde Solución es:  
     Título

*Restricción:* un requerimiento de alcance o calidad referido a una entidad dada. Es un atributo de los Recursos, Episodios básicos o subcomponentes de Contexto.  
 Sintaxis: ([Subject | Actor | Resource] + [**No**] **Debe** + Verbo + Predicado) | Frase  
 + significa composición, {x} significa cero o más ocurrencias de x, ( ) es usado para agrupar, | significa **or** y [x] denota que x es opcional.

Figura 2.2. Modelo de Escenarios [Leite *et al.* 2000]

A continuación se reproduce un ejemplo de escenario.

<p><b><u>solicitud de reserva</u></b></p> <ul style="list-style-type: none"> <li>• Goal             <ul style="list-style-type: none"> <li>– Atender una <u>solicitud de reserva</u> realizada por una persona, <u>agencia</u> u otro hotel.</li> </ul> </li> <li>• Context             <ul style="list-style-type: none"> <li>– Se realiza en la <u>recepción</u> del Hotel.</li> </ul> </li> <li>• Resources             <ul style="list-style-type: none"> <li>– <u>lista de precios</u></li> <li>– <u>planilla de reservas</u></li> <li>– <u>planilla de ocupación de habitaciones</u></li> <li>– teléfono</li> <li>– fax</li> <li>– e-mail</li> </ul> </li> <li>• Actors             <ul style="list-style-type: none"> <li>– <u>repcionista</u></li> <li>– <u>agencia</u></li> <li>– otro hotel</li> <li>– <u>pasajero</u></li> </ul> </li> <li>• Episodes             <ul style="list-style-type: none"> <li>– El <u>repcionista</u> verifica en la <u>planilla de ocupación de habitaciones</u> la <u>disponibilidad de habitaciones</u> para el período requerido y si hubiera, informa la <u>tarifa</u> y solicita la aprobación de la persona, <u>agencia</u> o de otro hotel.</li> <li>– El <u>repcionista</u> registra el nombre del <u>pasajero</u>, cantidad de ocupantes, características de la <u>habitación</u>, fecha de ingreso, fecha de egreso y <u>tarifa</u> en la <u>planilla de reservas</u></li> <li>– El <u>repcionista</u> actualiza la <u>disponibilidad de habitaciones</u> en la <u>planilla de ocupación de habitaciones</u>.</li> <li>– <b>restriction:</b> No hay <u>disponibilidad de habitaciones</u></li> <li>– <b>exception:</b> El teléfono, el fax o el e-mail no funcionan.</li> </ul> </li> </ul>
--

## 2.2. Medición del tamaño del software

Antes de avanzar sobre el tema específico de esta sección, conviene recordar el significado e implicancias del término “medición”. Fenton y Pflieger (1996) lo definen formalmente como: “*el proceso por el cual se asignan números o símbolos a los atributos de las entidades del mundo real, para describirlos de acuerdo a reglas claramente definidas*”. Por lo tanto, una medición captura información de atributos de las entidades, entendiendo por entidad un objeto o evento del mundo real. Para ello se requiere distinguir a los objetos según ciertas propiedades que los identifican. Esas propiedades o características de las entidades son conocidas como atributos y son expresadas mediante números o símbolos. Por ejemplo, el precio de un objeto es expresado por un número que representa la cantidad de pesos.

En las organizaciones de software exitosas la medición es una práctica cotidiana vinculada a las actividades técnicas o de gestión. La medición provee la información objetiva necesaria para tomar decisiones que impacten positivamente en el desenvolvimiento de los negocios y de las tareas de ingeniería [McGarry *et al.* 2001]. En este contexto, la medición del software juega un rol esencial como actividad de soporte para planificar, controlar y mejorar el proceso de desarrollo, lo que la ha convertido en una disciplina clave de la Ingeniería del Software.

En las etapas iniciales del desarrollo de software, el tamaño del software es un factor decisivo para la estimación de costos, cronograma, personal y esfuerzo del proyecto. El

tamaño del software, como el de cualquier otro artefacto, puede ser medido. La pregunta inmediata es: ¿qué se va a medir?, es decir, se debe establecer qué entidad va a ser considerada como representativa del tamaño. Dependiendo del punto de vista desde el cual es observado, diferentes artefactos pueden ser cuantificados. Básicamente, existen dos tipos de medidas de tamaño del software: *medidas técnicas* y *medidas funcionales*. Las medidas técnicas son usadas para cuantificar procesos y productos software desde el punto de vista del desarrollador. Las medidas funcionales son usadas para cuantificar productos y servicios del software desde la perspectiva del usuario [Meli 2000].

### 2.2.1. Medidas técnicas

Este enfoque se basa en la medición de las características físicas (es decir, instrucciones comprensibles por la computadora) del software. Esta categoría de mediciones incluye a líneas de código fuente (SLOC), número de módulos, reportes, pantallas, clases, objetos, componentes, etc. [Meli 2000]. La principal desventaja de estos métodos de medición es que sólo son aplicables en etapas tardías del proceso de desarrollo.

En particular, las SLOC miden la longitud del código del software. El método es muy usado en las organizaciones puesto que el código es un artefacto completamente formal que puede ser procesado automáticamente para extraer medidas [Morasca 2001]. Sin embargo, la técnica es cuestionada por investigadores y profesionales. En los siguientes párrafos son mencionadas las principales críticas.

No existe un estándar universalmente aceptado acerca de qué es una SLOC. Existen diferentes convenciones para contar las líneas de código cuyos resultados pueden diferir significativamente. En general hay consenso que las líneas en blanco y comentarios deben ser excluidas y cierta controversia sobre las declaraciones de datos, líneas físicas, sentencias ejecutables, etc. [Fenton y Pfleeger 1996]. El problema de las convenciones se complica aún más cuando el sistema usa más de un lenguaje. En general, en una misma organización es posible establecer e imponer algún conjunto de convenciones, pero es difícil determinar el significado de las SLOC cuando se utilizan repositorios, herramientas CASE (Computer Aided Software Engineering), etc. [Symons 1991]. Por otro lado, en 2008 hay más de 700 lenguajes (y dialectos) de programación en uso y sólo hay reglas para contar SLOC para menos de 100 lenguajes. [Jones 2008a].

Las SLOC son dependientes del lenguaje de programación, lo que dificulta la comparación entre proyectos desarrollados en diferentes lenguajes [Fenton y Neil 1999]. Otro problema es que la mayoría de las medidas del código asume que el código es puramente textual. El advenimiento de la programación visual y los ambientes basados en ventanas así como el desarrollo orientado a objetos, planteó nuevas cuestiones, como por ejemplo, cómo medir la longitud de los objetos provistos por el entorno, los objetos gráficos y los objetos construidos externamente [Fenton y Pfleeger 1996]. Al respecto Jones (2008a) afirma que para algunos lenguajes, como los visuales, no hay reglas efectivas debido a que gran parte de la programación es realizada usando alternativas tales como menús y botones en lugar de código procedural.

El número total de SLOC en un programa es muy subjetivo y dependiente del estilo del programador. Diferentes programadores, usando el mismo lenguaje, pueden generar diferente número de SLOC para implementar la misma funcionalidad.

No pueden aplicarse temprano en el proceso de desarrollo de software. Las SLOC sólo pueden ser medidas cuando el sistema está completamente implementado en un lenguaje. Sólo aquellos profesionales con gran experiencia y habilidad para estimar por analogía con otros proyectos pueden utilizar esos métodos de manera confiable [Symons 1991].

### 2.2.2. Medidas funcionales

Las métricas orientadas a la función miden la *funcionalidad* suministrada al usuario por el software. A continuación se presentan algunas definiciones del término funcionalidad (del inglés *functionality*):

*Merriam-Webster Online Dictionary*: “the quality or state of being functional; especially: the particular set of functions or capabilities associated with computer software or hardware or an electronic device”<sup>5</sup> [Merriam-Webster 2009].

*Oxford Advanced Learner’s Dictionary*: (computing) “the range of functions that a computer or other electronic system can perform”<sup>6</sup> [Hornby 2000].

*Pressman*: “utilidad” del programa [Pressman 1993].

Como la funcionalidad no puede medirse directamente, debe ser derivada indirectamente desde otras medidas directas. Albrecht (1979) introdujo la primera de estas métricas, el Análisis de Puntos Función (FPA – Function Points Analysis), en el marco de un enfoque para medir la productividad en el desarrollo de software para Sistemas de Información de Gestión (MIS – Management Information Systems). En ese contexto, propuso a los Puntos Función (FP – Function Points) como una medida del tamaño funcional del software [Pressman 2001]. Los FP son derivados usando una relación empírica basada en la cuantificación y evaluación de la complejidad de cinco (en la versión de 1983, cuatro en la original de 1979) categorías de funciones identificables por el usuario.

A principios de la década de 1980 el enfoque de Albrecht fue adoptado por el International Function Point Users Group (IFPUG), organización que difunde, mantiene y atiende a los usuarios de la técnica. Desde entonces han sido propuestos múltiples extensiones y enfoques alternativos a los FP, principalmente orientados a extender la aplicabilidad más allá del dominio MIS, como fue propuesto originalmente por Albrecht. Esos métodos difieren en sus visiones y definiciones de tamaño funcional [Fetcke *et al.* 2001] pero comparten el concepto central de medir la funcionalidad. La Figura 2.3 presenta la evolución de los principales métodos de Medición del Tamaño Funcional (FSM – Functional Size Measurement).

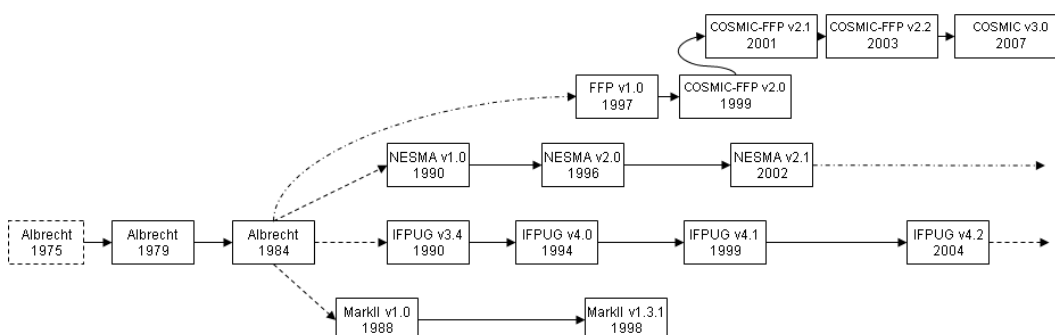


Figura 2.3. Evolución de los métodos de FSM [Functional Size Measurement 2008]

<sup>5</sup> la calidad o estado de ser funcional; especialmente: el conjunto particular de funciones o capacidades asociadas con el software o el hardware informático o un dispositivo electrónico. (traducción propia).

<sup>6</sup> (informática) la variedad de funciones que puede realizar una computadora u otro dispositivo electrónico. (traducción propia).

El FPA permite medir la funcionalidad del software independientemente de la tecnología usada para el desarrollo y de las decisiones de implementación. Comparado con las medidas técnicas, por ejemplo las basadas en SLOC, el tamaño funcional puede ser determinado en una fase más temprana del ciclo de vida, en cuanto los requerimientos funcionales del usuario son identificables. Disponer de una medida temprana del tamaño del software representa una ventaja significativa desde la perspectiva de la estimación y planificación de un proyecto.

A pesar de las ventajas, el FPA no está exento de críticas, que han sido señaladas por diferentes autores [Jeffery *et al.* 1993], [Fetcke *et al.* 2001], [MacDonell 1994], [Abran y Robillard 1994] y apuntan principalmente a que el método fue desarrollado empíricamente en base a la opinión de expertos y no cuenta con sólidas bases teóricas, no adhiere a los principios de la teoría de la medición, presenta inconsistencias en las escalas usadas en las diferentes fases del conteo de FPs y no cuenta con un modelo de medición explícito. No obstante, la métrica de FP definida por IFPUG es la más usada en la actualidad con un número total de proyectos medidos que probablemente exceda los 60.000 en los Estados Unidos y 100.000 en todo el mundo [Jones 2008a].

### 2.2.3. El Método IFPUG FPA

En el contexto de IFPUG FPA (2000), un sistema de software es visualizado como un conjunto de *funciones de datos* y *funciones transaccionales*<sup>7</sup>. La colección de funciones a considerar para una medición depende del propósito de la medición, es decir, el problema específico al que se requiere dar respuesta (por ejemplo, usar los FP para estimar el esfuerzo de un nuevo desarrollo). En función del propósito de la medición se determina el tipo de medición de FP (medición de un proyecto de desarrollo, medición de un proyecto de mejora<sup>8</sup>, medición de una aplicación) y el alcance de la medición (la funcionalidad que será incluida en una medición de FP concreta). Ambos influyen en la localización del límite entre el software a medir y otro software periférico. Es importante tener en cuenta que el límite del sistema debe fijarse desde la perspectiva del usuario y no en base a consideraciones físicas o técnicas.

Las *funciones de datos* representan la funcionalidad provista al usuario para satisfacer los requerimientos de datos internos y externos y por ende son clasificadas como *Archivos Lógicos Internos* (ILF – Internal Logical File) y *Archivos de Interfaz Externos* (EIF – External Interface File). La diferencia entre ellos radica en que los ILFs son mantenidos<sup>9</sup> por la aplicación que se está midiendo en tanto que los EIFs son referenciados por la aplicación medida pero mantenidos dentro del límite de otra aplicación.

Las *funciones transaccionales* representan la funcionalidad suministrada al usuario para el procesamiento de información. Estas funciones son clasificadas como *Entradas Externas* (EI – External Input), *Consultas Externas* (EQ – External Inquiry) y *Salidas Externas* (EO – External Output). Una EI es un proceso elemental<sup>10</sup> que procesa datos o información de control que viene desde fuera del límite de la aplicación. Una EQ es un

---

<sup>7</sup> *Función transaccional o transacción* son semánticamente equivalentes [Longstreet 2003].

<sup>8</sup> La *mejora del software* es la modificación del software para añadir, cambiar y/o eliminar funcionalidad, a diferencia del *mantenimiento del software* que es la modificación para corregir defectos, mejorar el rendimiento y/o soportar cambios en el entorno computacional [ISO 1998].

<sup>9</sup> El término *mantenido* se refiere a la capacidad de modificar datos a través de un proceso elemental [IFPUG 2000].

<sup>10</sup> Es la unidad de actividad más pequeña que es significativa para el usuario. Debe ser autosuficiente y dejar la aplicación en estado consistente.



proceso elemental que envía datos o información de control fuera del límite de la aplicación. Una EO es un proceso elemental que envía datos o información de control fuera del límite de la aplicación. A diferencia de las EQs, la lógica del proceso debe contener una fórmula matemática, un cálculo o crear datos derivados.

El Manual para la Medición de Puntos Función Versión 4.1.1<sup>11</sup> [IFPUG 2000] presenta un esquema con los componentes claves del FPA (Figura 2.4) y el procedimiento para medir los FP (Figura 2.5).

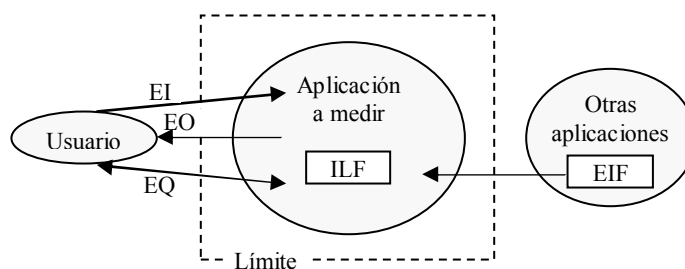


Figura 2.4. La aplicación desde el punto de vista del usuario [IFPUG 2000]

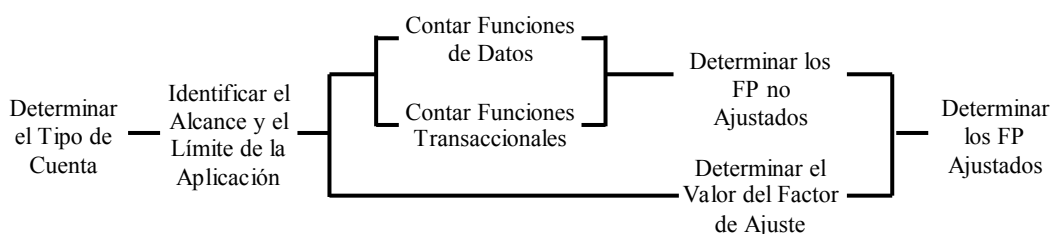


Figura 2.5. Procedimiento de Medición de Puntos Función [IFPUG 2000]

La *Cuenta<sup>12</sup> de FP no Ajustados* se basa en las funciones de datos y transaccionales. Para ello ambos tipos de funciones son clasificados en tres niveles de complejidad: *baja*, *media* o *alta*. La complejidad y contribución al tamaño funcional de los ILFs y EIFs es determinada desde tablas en función del número de *Tipos de Datos Elementales* (DET – Data Element Type) y *Tipos de Registros Elementales* (RET – Record Element Type); de modo similar, las EIs, EOs y EQs son valoradas según el número de *DETs* y *Tipos de Archivos Referenciados* (FTR – File Type Referenced). La suma de la contribución de todas las funciones produce el valor Puntos Función no Ajustados (UFP – Unadjusted Function Points) [IFPUG 2000]. El Apéndice A incluye una descripción más detallada del Método IFPUG.

Los FP Ajustados son calculados aplicando un Factor de Ajuste a los UFP. Este factor es determinado en función de las características técnicas del sistema. En el contexto de la medición del tamaño funcional esta parte del proceso no debe ser considerada. Los conceptos desarrollados en la sección 2.2.4 justifican esta afirmación.

#### 2.2.4. El estándar ISO 14143 para la Medición del tamaño funcional

El estándar ISO 14143 es un conjunto de cinco documentos que constituyen el estándar internacional para la FSM del software. Este documento especifica un conjunto de conceptos requeridos para la FSM y lo enfoca desde la perspectiva de un método de

<sup>11</sup> La versión vigente en la actualidad es *Counting Practices Manual, Release 4.2*, pero el autor de esta tesis no ha tenido acceso a la misma.

<sup>12</sup> Traducción del inglés “*count*” usada en el Manual de IFPUG Versión 4.1.1 [IFPUG 2000].

medición [Abran y Jacquet 1999]. El grupo de trabajo WG12 del subcomité ISO para la Ingeniería del Software (ISO/IEC/JTC1/SC7) avanzó en la estrategia de identificación y documentación de los criterios que debería cumplir un Método de Medición del Tamaño Funcional (FSMM – Functional Size Measurement Method) para ser reconocido como un estándar internacional.

En particular, el documento ISO/IEC 14143-1, Part 1 (1998): “Definition of concepts” [ISO 1998] incluye la definición de los conceptos fundamentales de la FSM y los principios generales para aplicar un FSMM. Este documento *no* provee reglas detalladas acerca de cómo:

- medir el tamaño funcional del software usando un método particular,
- usar los resultados obtenidos desde un método particular,
- seleccionar un método particular.

### **Definiciones** [ISO 1998]

**Usuario:** *cualquier persona o cosa que se comunica o interactúa con el software en cualquier momento durante el tiempo de vida del software.*

Ejemplo: administrador, empleado en un sistema de gestión de ventas.

**Requerimientos Funcionales del Usuario (FUR – Functional User Requirements):** *un subconjunto de los requerimientos del usuario. Los Requerimientos Funcionales del Usuario representan las prácticas y procedimientos que el software debe ejecutar para cumplir con las necesidades del usuario. Excluyen las características de calidad descritas en ISO 9126 y las características técnicas que describen como se provee el soporte al software.*

Ejemplo: un FUR podría ser “Mantener Clientes”.

**Tamaño Funcional:** *un tamaño del software derivado cuantificando los Requerimientos Funcionales del Usuario.*

**Medición del Tamaño Funcional:** *el proceso de medir el Tamaño Funcional.*

**Método de Medición de Tamaño Funcional:** *una implementación específica de FSM definida por un conjunto de reglas que cumplen con las características obligatorias de este estándar internacional.*

Ejemplo: el método IFPUG FPA.

**Dominio Funcional:** *una clase de software basado en las características de los Requerimientos Funcionales del Usuario que son pertinentes para la FSM.*

Ejemplo: MIS, sistemas de tiempo real, software científico, sistemas de control.

**Componente Funcional Básico (BFC – Base Functional Component):** *una unidad elemental de Requerimientos Funcionales del Usuario definida y usada por un método de FSM para propósitos de medición. Esta definición es reforzada por el siguiente requerimiento: sólo expresa Requerimientos Funcionales del Usuario, no expresa Requerimientos Técnicos y de Calidad.*

Ejemplo: el FUR “Mantener Clientes” puede consistir de los siguientes BFCs: “Agregar un nuevo cliente”, “Actualizar datos del cliente” y “Borrar cliente”.

**Límite:** *interfaz conceptual entre el software bajo estudio y sus usuarios. El límite actúa como una “membrana” a través de la cual los datos provenientes del mundo externo pasan hacia el mundo interno del software e inversamente [IFPUG 2000].*

Ejemplo: el BFC “Agregar un nuevo cliente” requiere que los datos del cliente atraviesen el límite desde afuera hacia adentro.

**Alcance:** determina los BFCs a ser incluidos en una aplicación específica de un método de FSM. El alcance está determinado por el propósito de la medición.

Ejemplo: si se requiere usar el tamaño como entrada para determinar el esfuerzo de un nuevo desarrollo, el alcance incluye todas las funciones impactadas (construidas o personalizadas) por las actividades del proyecto. En cambio, si se debe estimar el esfuerzo para una mejora, el alcance incluye todas las funciones que se van a añadir, cambiar o borrar [IFPUG 2000].

### **Características y requerimientos de la FSM**

*Un FSMM debe tener las siguientes características:*

- está basado en una visión conceptual del usuario del software,  
Ejemplo: no deben ser consideradas las funciones necesarias desde la visión del desarrollador, como puede ser la validación de datos, si no es requerida por el usuario.
- no está influenciado por los métodos o tecnologías usadas para desarrollar o implementar el software,  
Ejemplo: los métodos orientados a objetos o el lenguaje de programación.
- puede ser aplicado en cualquier momento durante el tiempo de vida del software, ni bien se hayan definido los requerimientos funcionales del usuario,  
Ejemplo: medición del tamaño funcional de un proyecto de mejora o de una aplicación terminada.
- deriva el Tamaño Funcional a partir de la evaluación de los BFCs,  
Ejemplo: el tamaño funcional obtenido a partir de una función que recibe como parámetros las contribuciones de los diferentes tipos de BFCs.
- es repetible<sup>13</sup>.

*Un BFC debe tener las siguientes características:*

- expresa FURs, no expresa requerimientos técnicos o de implementación,  
Ejemplo: especificaciones de rendimiento o confiabilidad del sistema, lenguaje de programación, sistema operativo o tecnología de bases de datos no son FURs.
- es clasificado como perteneciente a un único tipo de BFC.  
Ejemplo: en IFPUG un archivo lógico no puede ser contado a la vez como un Archivo Lógico Interno y un Archivo de Interfaz Externo en la misma aplicación.

*El Tamaño Funcional debe tener las siguientes características:*

- es obtenido sin hacer referencia al esfuerzo requerido y a los métodos usados para desarrollar y soportar el software y a los componentes físicos y tecnológicos del software que se está midiendo.  
Ejemplo: un archivo físico no necesariamente equivale a un archivo lógico cuando se ven los datos desde la perspectiva de la lógica del usuario [IFPUG 2000].

*Un FSMM debe definir:*

- los BFCs y cómo son valorados,  
Ejemplo: en IFPUG el *proceso elemental* es un BFC, el cual es valorado en función del número de Tipos de Datos Elementales y Tipos de Archivos Referenciados.

---

<sup>13</sup> Un método de medición es repetible si produce el mismo resultado cuando es aplicado sobre el mismo software por personas que tienen las mismas habilidades [Jacquet y Abran 1997].

- las unidades en que se expresa el *tamaño funcional*,  
Ejemplo: “Puntos Función”.
- el *Dominio Funcional* al cual puede aplicarse el método.  
Ejemplo: dominio MIS.

*Un FSMM debe definir:*

- los tipos de BFC,  
Ejemplo: en IFPUG los tipos de BFC son: Archivo Lógico Interno, Archivo de Interfaz Externo, Entrada Externa, Salida Externa y Consulta Externa.
- cómo agrupar y clasificar los BFCs en tipos de BFC,  
Ejemplo: en IFPUG un proceso elemental cuyo propósito principal es mantener un Archivo Lógico Interno es clasificado como Entrada Externa.
- cómo identificar los BFC en los FURs,  
Ejemplo: en IFPUG, existen reglas para identificar procesos elementales, una de ellas establece: “El proceso es la menor unidad de actividad que tiene sentido para el usuario” [IFPUG 2000].
- cómo determinar qué BFC estarán incluidos en el alcance de la FSM,  
Ejemplo: ver definición de *alcance*.
- las reglas que determinan el valor numérico asignado a cada tipo de BFC,  
Ejemplo: en IFPUG a una Salida Externa se le asigna el valor 4, 5 o 7 dependiendo de su complejidad (baja, media o alta).
- la relación entre el tipo de BFC y el límite,  
Ejemplo: en IFPUG una relación de un tipo de BFC con el límite es: “un Archivo Lógicos Interno debe residir dentro del límite del software”.
- la relación, si existe, entre los tipos de BFCs.  
Ejemplo: en IFPUG una relación entre tipos de BFC es: “un Archivo Lógico Interno debe ser mantenido por una o más Entradas Externas”.

*Un FSMM debe describir las siguientes actividades para derivar el tamaño funcional:*

1. Determinar el alcance del FSM.
2. Identificar los FUR dentro del alcance del FSM.
3. Identificar los BFCs dentro de los FUR.
4. Clasificar los BFCs en Tipos de BFC, si es aplicable.
5. Asignar el valor numérico apropiado a cada BFC.
6. Calcular el Tamaño Funcional.

#### **2.2.5. El Método IFPUG, el estándar ISO 20926 y su relación con ISO 14143-1**

Para cumplir con las definiciones de *tamaño funcional*, *Medición del Tamaño Funcional* y *Método de Medición del Tamaño Funcional* del estándar ISO 14143-1 [ISO 1998], el procedimiento de medición (Figura 2.5) finaliza con el cálculo de los UFP. De hecho, el estándar “ISO/IEC 20926:2003 IFPUG 4.1 Unadjusted functional size measurement method” reconoce al *método de medición de tamaño funcional no ajustado*. Por lo tanto, en esta tesis los acrónimos FP y UFP serán usados indistintamente.

### 2.2.6. Usos y beneficios de la Medición del tamaño funcional

ISO 14143-1 [ISO 1998] propone algunos usos de los métodos de FSM considerando dos áreas:

#### **Gestión de Proyectos**

***Seguimiento del progreso de un proyecto:*** en una etapa temprana del ciclo de vida de un proyecto, los métodos de FSM pueden crear un inventario con los BFCs de un sistema o la mejora de un sistema. Este inventario puede ser usado para evaluar y comunicar el progreso del proyecto, en primer lugar, haciendo el seguimiento de los cambios al conjunto de BFCs (es decir, identificando los BFCs añadidos y eliminados del inventario debido a los cambios en el alcance) y luego, observando cuáles BFCs fueron o no desarrollados. En consecuencia, el progreso del proyecto puede ser comunicado en función del avance en el desarrollo de los BFCs.

***Gestión de los cambios al software:*** el inventario de BFCs, generado al principio del ciclo de vida de un proyecto y acordado por usuarios y desarrolladores, es definido de acuerdo al alcance del software. Por cada cambio al conjunto de BFCs se podría calcular el tamaño funcional y usarlo en un modelo de estimación para predecir el esfuerzo y el impacto en el cronograma. Este impacto podría ser usado para negociar con los usuarios las modificaciones al software y al plan de proyecto.

***Conveniencia de un paquete de software:*** se podría evaluar si la funcionalidad proporcionada por un paquete se ajusta a los requerimientos funcionales. Para ello se debería calcular el tamaño funcional de los requerimientos funcionales del usuario y el tamaño funcional de los requerimientos funcionales satisfechos por el paquete. El grado de ajuste del paquete podría ser expresado en términos de la proporción de requerimientos funcionales satisfechos.

#### **Predicción y Gestión de rendimiento**

***Estimación del costo total de reemplazo del software:*** el tamaño funcional de una parte o todo el software podría ser usado en un modelo de estimación para determinar el costo total de una sustitución.

***Gestión del proceso de desarrollo, mejora y mantenimiento del software:*** se podrían analizar los indicadores de productividad (tamaño funcional dividido por esfuerzo, cronograma o costo) y las características demográficas<sup>14</sup> para determinar que características demográficas tienen el mayor impacto sobre la productividad. La gestión de la productividad podría ser realizada manipulando esas características y monitoreando las tendencias de productividad en los futuros proyectos para determinar si se logró el efecto deseado.

***Gestión de la calidad:*** se podría analizar la densidad de defectos (número de defectos por unidad de tamaño funcional) para determinar qué características demográficas tienen el mayor impacto sobre la misma. La gestión de la densidad de defectos podría realizarse manipulando esas características y monitoreando las tendencias en cuanto a la densidad de defectos en los proyectos futuros para determinar si se logró el efecto deseado.

***Predicción de los recursos para proyectos de desarrollo y mejora:*** se podría construir un modelo de predicción basado en tres tipos de datos recolectados de una muestra de proyectos terminados: tamaño funcional, recursos consumidos (costo, esfuerzo o

---

<sup>14</sup> Características demográficas son las características del entorno, el proyecto o el personal que podrían afectar los procesos de desarrollo, mejora o mantenimiento de software. Por ejemplo, la experiencia del personal, el uso de herramientas o el lenguaje de programación.

cronograma) y características demográficas que podrían haber influido en la cantidad de recursos usados. Ese modelo podría ser usado al comienzo del ciclo de vida de los futuros proyectos para predecir los recursos o para evaluar la factibilidad de un proyecto.

**Presupuesto del mantenimiento:** se podría monitorear el tamaño funcional y el costo o esfuerzo de mantenimiento en comparación con el tamaño funcional. Esta información podría ser usada para predecir los presupuestos de mantenimiento.

**Gestión de los contratos:** el contrato con un proveedor podría basarse en los requerimientos funcionales, el tamaño funcional, un factor de productividad esperada y un costo por unidad de tamaño funcional.

Otros autores como Rule (2001), Heller (1995), Dekkers y Aguiar (2001) enumeran un conjunto de beneficios del FPA.

**Mejora en la definición de los requerimientos:** las técnicas de FPA ayudan a producir requerimientos del software que pueden ser medidos, permiten reducir la ambigüedad y aumentar el rigor en la documentación de los requerimientos. Por lo tanto, pueden ser usadas como base para el acuerdo entre clientes y proveedores [Rule 2001].

**Comunicación de los requerimientos funcionales:** al evitar el uso de la terminología técnica y enfocar sobre los requerimientos del usuario, el FPA es un excelente medio para comunicarse con los usuarios. La técnica puede ser usada para dirigir las entrevistas a los clientes y documentar los resultados. La documentación resultante provee un marco que describe los requerimientos del usuario y técnicos [Heller 1995].

**Verificación de la completitud de los requerimientos:** el FPA provee un marco de referencia adicional para verificar que se ha identificado el conjunto completo de requerimientos funcionales. Cada uno de los pasos del FPA ayuda a aclarar los requerimientos, permitiendo confirmar los requerimientos originales o revelando las inconsistencias en la comprensión. El aporte a la completitud va mucho más allá de la fase de requerimientos. Simplemente revisando el listado de FP se puede establecer si una determinada funcionalidad está incluida y en caso de comprobarse que no está, decidir su inclusión o no e incrementar los FP si correspondiere [Dekkers y Aguiar 2001].

Morris (2001) sintetiza del siguiente modo los beneficios de la FSM:

- Se puede usar *temprano* en el ciclo de vida.
- Es *independiente* de la tecnología, diseño y métodos.
- Es *fácil* de entender por el usuario.
- Se obtienen resultados *consistentes* ( $\pm 10\%$ ).
- Es un método establecido y *estandarizado*.
- Es *rápido*, el tiempo ocupado por el conteo de FP es mínimo frente al requerido para el desarrollo ( $< 1\%$ ).

### 2.2.7. Modelos asociados a la Medición del tamaño funcional

En esta sección se presenta una revisión de los modelos para la FSM propuestos por diferentes autores.

### 2.2.7.1. Modelo de Referencia para la Medición del tamaño funcional

El estándar ISO/IEC 14143-1 para la FSM especifica que el objeto<sup>15</sup> de medición de tamaño funcional es descrito por los FURs, los cuales son interpretados por un método de FSM en términos de BFCs. Con la incorporación de nuevos ambientes de desarrollo aparecen nuevos tipos de especificaciones (por ejemplo, las que usan notación UML (Unified Modeling Language) [Fowler 2003], distintas de las tradicionales. Para facilitar la identificación de los elementos relevantes para la medición dentro de cada contexto y para un determinado propósito, el grupo de trabajo sobre nuevas tecnologías de Netherlands Software Metrics Association), NESMA Workgroup NT, desarrolló un Modelo de Referencia [NESMA 2002] para los métodos de FSM. El estándar ISO/IEC 14143-1 fue usado como referencia para este modelo, que está basado en una versión preliminar de Dekkers y Kammelar (2001).

El Modelo de Referencia, representado mediante un diagrama de clases UML en la Figura 2.6, es una estructura genérica para cualquier objeto de medición de tamaño funcional considerado en un contexto específico denominado *dominio de medición*. Dentro de un dominio de medición, el enfoque determina los elementos medibles pertinentes para la disciplina<sup>16</sup> y el propósito. Por ejemplo, si el enfoque es desarrollo de una aplicación, la disciplina es nuevo desarrollo y el propósito podría ser la estimación del esfuerzo.

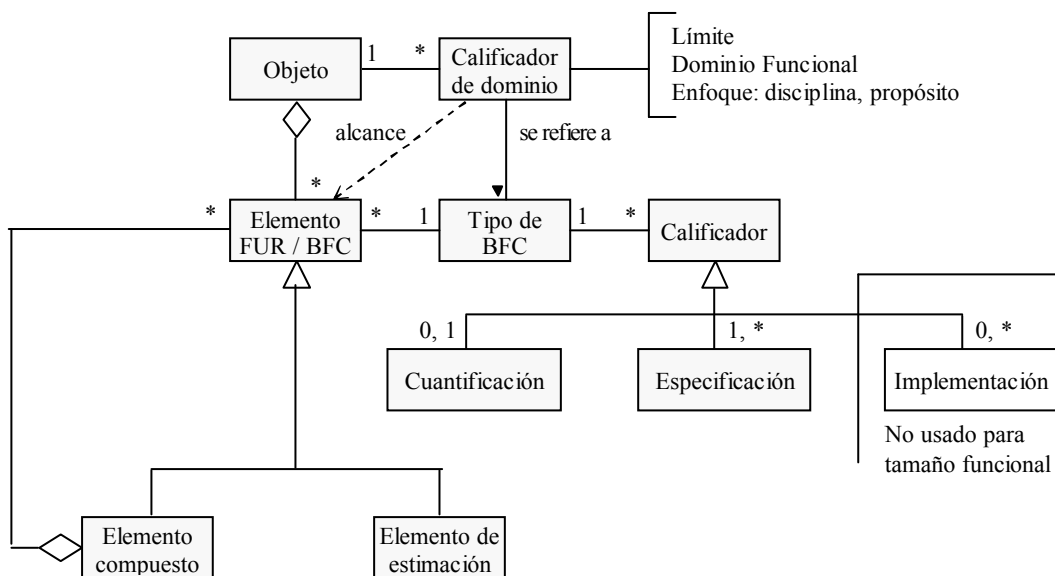


Figura 2.6. Modelo de Referencia para la FSM (traducido de [NESMA 2002])

<sup>15</sup> El término objeto (del inglés *subject*) se refiere a la entidad que va a ser medida y no tiene relación con el mismo vocablo usado en el contexto de la OO.

<sup>16</sup> Se considera equivalente a *Tipo de medición* en el manual de IFPUG (2000).

### Glosario de términos asociados al Modelo de Referencia

*Calificador*: una clase de indicador usado para el conteo y ponderación (o asignación de valores numéricos), relacionado con el tamaño funcional o esfuerzo.

*Cuantificación*: una categoría definida de calificador para el conteo y ponderación (o asignación de valores numéricos) que especifica las tablas de valuación y/o fórmulas relacionadas al tamaño funcional (y esfuerzo).

*Disciplina*: un sistema de normas que regulan las actividades, por ejemplo, desarrollo, prueba.

*Dominio de medición*: una clase de mediciones basada en las características de los objetos de medición que son pertinentes para la disciplina y el propósito.

*Elemento*: un subconjunto de los requerimientos del usuario que describe la aplicación en términos de lo “que el usuario puede hacer con la aplicación”.

*Elemento compuesto*: una unidad identificable de los requerimientos relacionados al tamaño, la mayoría representados en los FURs.

*Elemento de estimación*: una categoría definida de elementos que pueden ser medidos, principalmente representados en los BFCs.

*Especificación*: una categoría definida de calificadores para el conteo y ponderación (o asignación de valores numéricos) que describe los fundamentos relacionados al tamaño funcional.

El lado izquierdo del diagrama (Figura 2.6) muestra que un objeto de medición de tamaño comprende muchos *elementos*. Un elemento es una unidad identificable relacionada con el tamaño (FUR, BFC), que puede ser elemental o compuesta. Esta propiedad es modelada por una estructura de especialización. Los *elementos compuestos* pueden incluir muchos elementos de menor orden. Los *elementos de estimación* son las unidades elementales del objeto a las que se les puede asignar un tamaño. El nivel de la “unidad elemental” depende del grado de detalle usado para describir los elementos.

El objeto es modelado mediante una asociación uno-a-muchos con un calificador de dominio. Esto implica que un objeto puede adecuarse a más de un dominio de medición dependiendo de las propiedades que determinan el dominio de medición; es decir, el objeto tiene un cierto límite, pertenece a un determinado dominio funcional pero puede suponer más de un enfoque (desarrollo, prueba, mantenimiento).

El lado derecho del diagrama muestra las clases que permiten la medición de tamaño y sus relaciones. De arriba hacia abajo se muestran las siguientes clases: *calificador de dominio y tipo de BFC* que está relacionado a *calificador* con sus tres especializaciones.

El *calificador de dominio* especifica el dominio de medición a través de los siguientes atributos: límite, dominio funcional y enfoque; el enfoque representa la síntesis de la disciplina y el propósito. Entre el *calificador de dominio* y *elemento* se establece una asociación de dependencia etiquetada “alcance” (línea punteada). Esta asociación define la relevancia y la forma de los elementos identificables en relación al dominio de medición.

La relación “se refiere a” entre el *calificador de dominio* y el *tipo de BFC* implica que ciertos dominios de medición se refieren a un conjunto particular de tipos de BFC.



La clase *tipo de BFC* determina en qué categorías serán expresados los elementos del objeto de la medición de tamaño y qué calificadores serán usados para la medición. Tres tipos de calificadores son modelados: *especificación*, *cuantificación* e *implementación*. Un calificador relacionado a la *especificación* del elemento describe “para qué es el elemento” en términos de propósito, atributos, restricciones y propiedades. El calificador de tipo *cuantificación* especifica la fórmula para transformar los calificadores de especificación en un valor numérico. Un calificador relacionado con la *implementación* especifica “cómo se realiza u obtiene” (software, COTS – Common Off The Shelf). Este último calificador (implementación) no es considerado para la FSM.

Con respecto a los calificadores de los *elementos* en el diagrama, en la estructura del objeto ambos tipos de elementos (*compuesto* y *de estimación*) están relacionados a sus calificadores a través del *tipo de BFC*.

En la práctica el Modelo de Referencia puede ser aplicado a diferentes tipos de especificaciones. Para ello se deben seguir los siguientes pasos: identificar los BFCs descritos en las especificaciones, clasificar los BFCs y determinar los atributos cuantificables de los tipos de BFC (calificadores de especificación), determinar las reglas para valorar los BFCs (calificadores de cuantificación) y por último definir una unidad de medida pertinente para el dominio de la medición.

### El Modelo de Referencia para la FSM aplicado a IFPUG

En el dominio de medición del FPA, el objeto de medición es la aplicación, la disciplina es el desarrollo de la aplicación y el propósito es la medición del tamaño funcional. La Tabla 2.1 muestra la instanciación del Modelo de Referencia para el FPA estándar.

El FPA considera a la aplicación como una colección de elementos de estimación, no se reconocen elementos compuestos. Los elementos de estimación se clasifican como funciones de datos (1.1) y funciones transaccionales (1.2). Cada clase se refiere a tipos de BFC definidos por calificadores específicos (Tabla 2.1).

Tabla 2.1. Modelo de Referencia del FPA (traducido de [NESMA 2002])

Elementos		Calificadores	
	Objeto	Tipo	Descripción
	Aplicación	S <sup>1</sup>	descripción del dominio funcional descripción del límite descripción del alcance disciplina: desarrollo de software propósito: medir tamaño (costo, esfuerzo o valor relacionado)
Nivel	Elemento Compuesto	Tipo	Descripción
Nivel	Elemento Estimación	Tipo	Descripción
1.1	Funciones de datos		
1.2	Funciones transaccionales		
Nivel	Tipo de BFC	Tipo	Descripción
1.1.1	Archivo Lógico Interno (ILF)	S	#RET
1.1.2	Archivo de Interfaz Externo (EIF)		#DET
		Q <sup>2</sup>	Tablas de valoración para ILF, EIF
1.2.1	Entrada Externa (EI)	S	#FTR
1.2.2	Salida Externa (EO)		#DET
1.2.3	Consulta Externa (EQ)	Q	Tablas de valoración para EI, EO, EQ

<sup>1</sup>S: especificación; <sup>2</sup>Q: cuantificación.

**2.2.7.2. Proceso de Medición del tamaño funcional**

Desde la publicación del FPA en 1979 varias revisiones y enfoques alternativos para la FSM han sido propuestos. Fetcke *et al.* (2001) investigaron el FPA y algunas de sus variantes (IFPUG FPA 4.0 y 4.1, MkII FPA 1.3.1 [UKSMA 1998] y Full Function Points 1.0 [St-Pierre *et al.* 1997]) observando que, si bien esos métodos difieren en la visión del tamaño funcional, aún así comparten los conceptos fundamentales y están basados en un modelo implícito del software (es decir, cada método de FSM define una abstracción del software que representa los ítems considerados relevantes para el tamaño funcional). Con el objeto de uniformar el modelo del software para el FPA, Fetcke *et al.* (2001) propusieron una representación generalizada para los métodos antes mencionados. Esta representación formaliza la abstracción del software usada por esos métodos, que es calificada como *orientada a los datos*. El siguiente cuadro sintetiza los conceptos de la abstracción orientada a los datos.

*Concepto de usuario.* El usuario interactúa con una aplicación y no necesariamente se restringe a usuarios humanos, sino que puede incluir “usuarios” software y hardware.

*Concepto de aplicación.* La aplicación es el objeto de la medición. Las aplicaciones proveen funciones al usuario. Esas funciones son los atributos de interés.

*Concepto de transacción.* Las *transacciones* son procesos de interacción del usuario con la aplicación desde una perspectiva lógica o funcional.

*Concepto de dato.* Los datos son almacenados por la aplicación. Los datos elementales representan los ítems de datos más pequeños que son significativos para el usuario y están estructurados en grupos lógicamente relacionados, de modo similar a las tablas en una base de datos.

*Concepto de tipo.* Se refiere al principio que múltiples instancias del mismo componente lógico se consideran como un único tipo.

En la abstracción orientada a los datos (parte ubicada a la derecha en la Figura 2.7), los Requerimientos Funcionales del Usuario son representados en términos de transacciones y grupos de datos, que luego deben ser cuantificados para obtener el tamaño funcional del sistema. Por lo tanto, los métodos de FSM aplican un proceso basado en dos pasos principales: *paso de identificación* y *paso de medición* (Figura 2.7).

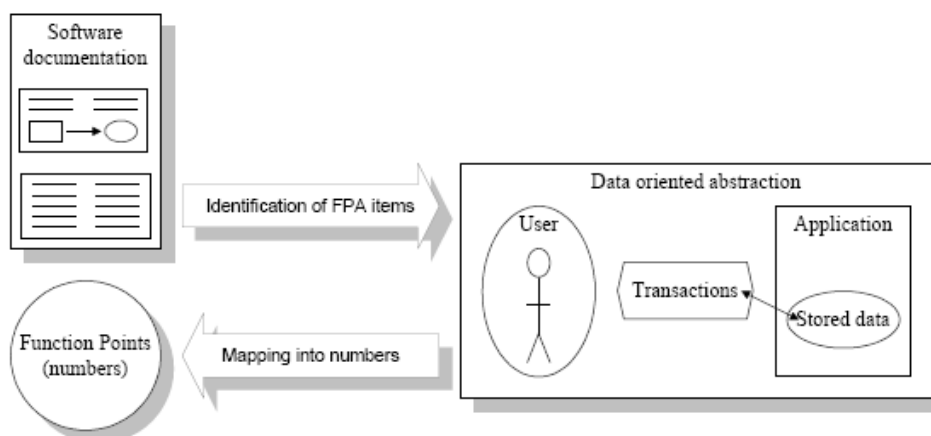


Figura 2.7. Proceso de medición del tamaño funcional [Fetcke *et al.* 2001]

En el primer paso, la abstracción orientada a los datos es aplicada a la documentación del software. De acuerdo a ISO 14143-1 la característica clave para la medición de un producto de software son los Requerimientos Funcionales del Usuario [ISO 1998]. Este estándar no prescribe un formato particular para esta documentación, por lo tanto un método de FSM debe definir las reglas para identificar los ítems que aportan tamaño funcional al sistema.

En el segundo paso, los ítems identificados son asociados con valores numéricos que representan su contribución al tamaño funcional, los que luego son sumados para obtener el valor total del tamaño funcional.

Aunque pueden no estar claramente separados, ambos pasos son definidos en los documentos que definen los métodos de FSM.

### 2.2.7.3. Modelo de Proceso para los Métodos de medición del software

En el marco de una investigación sobre la validación de las métricas del software, Jacquet y Abran propusieron un Modelo de Proceso de Medición [Jacquet y Abran 1997] que sirviera como base para el desarrollo de un marco de validación de un método de medición. Cabe destacar que este Modelo de Proceso fue usado como base para el diseño de la Guía para la Verificación de los Métodos de Medición de Tamaño Funcional (Guide to the Verification of Functional Size Measurement Methods (ISO 14143-3)).

El modelo de proceso de medición resalta los pasos claves de un método de medición, desde su diseño hasta la utilización de sus resultados en los subsiguientes modelos, como por ejemplo, modelos de estimación o de calidad. El modelo de alto nivel se compone de cuatro pasos (Figura 2.8) que describen las actividades para la construcción de un método de medición, la medición del software aplicando el método, la evaluación del resultado producido y su posterior uso en modelos cuantitativos o cualitativos.

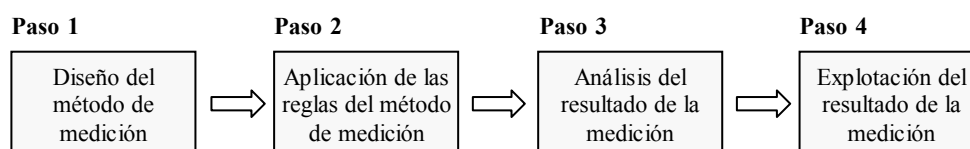


Figura 2.8. Proceso de Medición - Modelo de alto nivel (traducido de [Jacquet y Abran 1997])

Cada paso del modelo de alto nivel se compone de subpasos que se describen en el modelo detallado. Los objetivos específicos formulados para satisfacer el objetivo de esta tesis están asociados a los primeros tres pasos del modelo de alto nivel.

#### **Paso 1: Diseño del método de medición.**

*Subpaso 1. Definición de los objetivos del método de FSM.* Antes de diseñar el método de medición es importante conocer qué se quiere medir (qué clase de software y qué atributo), desde qué punto de vista se realizará la medición (usuario del software, desarrollador del software, etc.) y cuál es el uso previsto del método de medición.

*Subpaso 2. Caracterización<sup>17</sup> del concepto a medir.* Para construir el método de medición, el concepto a medir debe ser claramente definido. En el caso de conceptos abstractos como el de funcionalidad, se debe definir qué subconceptos lo integran, cómo está definido cada subconcepto y cuál es su rol como parte constitutiva del concepto. Por

<sup>17</sup> Caracterizar: determinar los atributos peculiares de alguien o de algo, de modo que claramente se distinga de los demás. Diccionario de la Real Academia Española (<http://buscon.rae.es/draeI/>).

ejemplo, en el marco de IFPUG, la funcionalidad es expresada mediante cinco tipos de funciones (ILF, EIF, EI, EO y EQ), las que son definidas en términos de otros subconceptos (por ejemplo, DET y FTR para las EIs, EOs y EQs).

*Subpaso 3. Selección o diseño del metamodelo.* El conjunto de características seleccionadas para representar un software o una parte del software y el conjunto de sus relaciones constituyen el metamodelo propuesto para la descripción del software al cual será aplicado el método de medición. Por lo tanto, el metamodelo debe describir los tipos de entidades usadas para representar el software y las reglas que permiten su identificación. Por ejemplo, el metamodelo de IFPUG describe todos los tipos de entidades que contribuyen al tamaño funcional del sistema (ILF, EIF, EI, EO y EQ) y el manual de medición especifica las reglas para identificarlos.

Los subpasos 2 y 3 están estrechamente relacionados, pues para definir el concepto es necesario contar con una representación del tipo de objetos a ser medidos, es decir, se requiere un metamodelo. Y por otro lado, el diseño del metamodelo está determinado por la forma en que el concepto es definido. Esto significa que los dos subpasos están en un mismo nivel en el modelo de proceso detallado.

*Subpaso 4. Definición de las reglas de asignación numérica.* En este paso son definidas las reglas que permiten asociar un valor numérico a un sistema de software. Estas reglas deben ser definidas de acuerdo al metamodelo propuesto y a la caracterización del concepto a medir. Por ejemplo, en el método IFPUG, una regla para contar DET en un ILF o EIF establece: “Contar un DET por cada campo único, no repetido, reconocible por el usuario y mantenido o recuperado desde un ILF o EIF mediante la ejecución de un proceso elemental” [IFPUG 2000].

### **Paso 2. Aplicación del método de medición.**

*Subpaso 1. Recopilación de la documentación del software.* La documentación requerida para aplicar el método de medición es obtenida desde el software a medir. Por ejemplo, para aplicar IFPUG se requiere contar con la Especificación de Requerimientos.

*Subpaso 2. Construcción del modelo del software.* En base al metamodelo y las reglas propuestas son identificados los componentes que formarán parte de la medición. Por ejemplo, en IFPUG, se pueden identificar los ILFs desde la especificación de requerimientos aplicando las respectivas reglas.

*Subpaso 3. Aplicación de las reglas de asignación numérica.* Las reglas derivadas del subpaso 4 del Paso 1 son aplicadas al modelo del software generado en el paso previo.

### **Paso 3. Análisis del resultado de la medición.**

*Subpaso 1. Presentación de los resultados de la medición.* El resultado de la medición, para ser evaluado, debe estar documentado (unidad, descripción de los resultados intermedios, etc.). Por ejemplo, los resultados intermedios y final de una medición con IFPUG constan en una planilla.

*Subpaso 2. Verificación de los resultados.* Los resultados deberían ser revisados de diferentes maneras para determinar su calidad. También podrían ser comparados con otros resultados conocidos (valores de referencia) para evaluar su exactitud. Por ejemplo, verificar los cálculos matemáticos o comparar el resultado con el producido por un experto para el mismo sistema.

### 2.3. Especificación de procesos software

Un requisito fundamental para mejorar la calidad de los procesos software dentro de una organización es desarrollar y usar procesos definidos. Un “proceso definido” es aquél que está documentado, incluye soporte (por ejemplo, capacitación) y es aplicable en la práctica en términos de tiempo y costo. La descripción de un proceso software debe ser tan rigurosa como sea posible, de modo tal que sirva de guía para su efectiva ejecución. En esta sección se describen los conceptos asociados a la definición de un proceso software y los criterios para la definición de procesos del Software Process Framework (SPF) propuesto por Olson *et al.* (1993).

#### 2.3.1. Introducción

Primeramente se debe establecer la diferencia entre los conceptos de proceso y descripción de un proceso. Un proceso es un medio para hacer una tarea mientras que la descripción de un proceso es una especificación de cómo hacer la tarea [Osterweil 1987]. El cuadro adjunto incluye las respectivas definiciones.

<p><b>Definiciones</b> [Olson <i>et al.</i> 1993]</p> <p><i>Proceso</i>: un enfoque sistemático para la creación de un producto o la realización de alguna tarea. También es definido como una secuencia de pasos ejecutados para un propósito dado.</p> <p><i>Descripción de un proceso</i>: las definiciones operacionales de los principales componentes de un proceso. La documentación que especifica de manera completa, precisa y verificable los requerimientos, diseño, comportamiento u otras características de un proceso.</p> <p><i>Proceso bien definido</i>: aquél que incluye criterios de entrada o precondiciones, entradas, estándares y procedimientos para ejecutar el trabajo, mecanismos de verificación, criterios de salida o poscondiciones y salidas.</p>
--

En el momento de desarrollar la documentación de un proceso software los diseñadores del proceso pueden enfrentarse con dudas sobre la clase de información del proceso que debe ser documentada o la forma en que esta información debe ser organizada. Estas cuestiones son respondidas por diferentes aspectos del SPF: la primera es considerada por los Criterios para la definición de procesos, la segunda por el Marco operativo (Operational framework). Existe otro aspecto referido al CMM que no es considerado en el contexto de esta tesis.

En la siguiente sección se presenta una breve descripción de los Criterios para la definición de procesos. El Apéndice B incluye el detalle del Marco operativo (Operational framework) del SPF.

#### 2.3.2. Criterios para la definición de procesos

Estos criterios representan el conjunto de información que debe estar incluida en la descripción de un proceso software para que éste pueda ser usado por las personas que ejecutarán el proceso. Concretamente, para ejecutar un proceso o actividad, la definición del proceso debe responder a preguntas claves (Key Process Questions) tales como [Olson *et al.* 1993]:

- ¿qué productos<sup>18</sup> o artefactos son producidos?
- ¿qué actividades son ejecutadas?
- ¿qué agentes están involucrados?
- ¿cómo son implementadas las actividades?
- ¿cuándo comienzan y finalizan las actividades?
- ¿qué agentes son responsables de ejecutar cada actividad?
- ¿qué artefactos son producidos (o consumidos) por cuáles actividades?
- ¿por qué o para qué es realizada una actividad?
- ¿cuál es la próxima actividad?

Las respuestas a estas preguntas son los “elementos del proceso” que deben estar documentados o definidos [Olson *et al.* 1993]. Estos elementos forman parte de las plantillas (*templates*) provistas por el SPF para facilitar la definición de procesos software.

A continuación se presenta el listado con las definiciones de la terminología del SPF. Cada definición es complementada con un ejemplo de la parte del proceso de medición de FP que comprende el conteo de FP de las funciones de datos en el método IFPUG (2000).

*rol*: unidad de responsabilidades definidas que pueden ser asumidas por uno o más individuos. El ejecutor que realiza la acción para alcanzar o crear un producto, servicio o resultado. Un rol puede estar automatizado.

Ejemplo: Analista, Ingeniero de Requerimientos.

*criterio de entrada*: está formado por el conjunto de condiciones bajo las cuales puede comenzar una actividad. Se representa mediante un predicado simple o compuesto que establece el estado de un producto, agente o actividad.

Ejemplo: La Especificación de Requerimientos está disponible y el límite del sistema está determinado.

*entrada*: son los productos usados por un proceso que fueron producidos como resultado de una actividad previa y son usados por la actividad actual.

Ejemplo: Especificación de Requerimientos.

*actividad*: acción realizada para crear o alcanzar un producto, servicio o resultado.

Ejemplo: Identificar los ILFs y EIFs. Por cada ILF y EIF identificar los DETs y RETs, determinar su complejidad y contribución en FP. Sumar los FP de los ILFs y EIFs.

*criterio de salida*: está formado por el conjunto de condiciones bajo las cuales una actividad se puede considerar completa. Se representa mediante un predicado simple o compuesto que establece el estado de un producto, agente o actividad.

Ejemplo: Se determinó la complejidad y contribución en FP de todos los ILFs y EIFs.

*salida*: son los productos que resultan de la ejecución de un proceso que serán usados por una actividad subsiguiente.

Ejemplo: Cuenta de FP de las Funciones de Datos.

Además de los elementos básicos del proceso hay otras piezas de información útiles para incluir en las descripciones de procesos. Son enumeradas a continuación:

---

<sup>18</sup> *producto (work product)*: cualquier artefacto, servicio o resultado de un proceso o actividad.

*revisiones y auditorias*: lista de revisiones y auditorías ejecutadas durante el proceso.  
Ejemplo: Proceso de revisión de la medición de FP propuesto por Morris y Desharnais (2001).

*mediciones*: descripción de los procesos de medición. Representan las acciones ejecutadas para cuantificar determinados parámetros o entidades del proceso.  
Ejemplo: Medición del esfuerzo requerido para ejecutar el conteo de FP.

*productos gestionados y controlados*: lista de productos (documentación, etc.) a ser gestionados y controlados.  
Ejemplo: ILFs, EIFs.

*procedimiento documentado*: actividades a realizar para ejecutar una tarea determinada. Describe las instrucciones paso a paso para realizar una función específica que implementa una actividad. Se requiere un procedimiento documentado para que los individuos responsables de una tarea o actividad sean capaces de ejecutarla en forma repetible y tal que otros con conocimientos generales puedan aprender y ejecutar la tarea o actividad de igual modo.

Ejemplo: Proceso de medición propuesto en el manual de IFPUG (2000).

*entrenamiento*: conocimiento y habilidades requeridas para ejecutar un proceso o usar un procedimiento.

Ejemplo: los miembros del equipo que realiza la medición tienen experiencia en la aplicación del método IFPUG.

*herramientas*: cualquier soporte automatizado necesario para implementar un procedimiento, por ejemplo, una herramienta específica o una hoja de cálculo.

Ejemplo: Planilla de cálculo o una herramienta específica para medir FP con el método IFPUG.

## 2.4. Resumen

En este capítulo, en primer lugar, fueron introducidas las bases conceptuales para el diseño del procedimiento SFP, que es desarrollado en el Capítulo 4.

El LEL y Escenarios son dos herramientas que permiten recolectar, describir y organizar la información obtenida a través de entrevistas con el usuario (o eventualmente a través de otro medio) usando una representación semiestructurada basada en lenguaje natural, con las consiguientes ventajas para la comunicación y validación con los agentes involucrados en el proceso de desarrollo. En el contexto de esta tesis, el LEL y los escenarios constituyen los documentos fuente para la medición del tamaño funcional de un sistema de software.

Con respecto al diseño de un método de medición del tamaño funcional, varios aspectos fueron considerados: 1) la normativa vigente para la FSM, representada por el estándar ISO 14143-1 que especifica los conceptos esenciales de la FSM; 2) la filosofía del FPA estándar, que visualiza la funcionalidad de un sistema en términos de transacciones y datos; 3) el modelo genérico provisto por el Modelo de Referencia para la FSM, que puede ser instanciado para diferentes métodos y tipos de especificaciones; 4) el modelo para los Métodos de FSM de Fetcke *et al.* que identifica los pasos claves del proceso de FSM; 5) el Modelo de Proceso para los Métodos de Medición del Software, que guía la construcción y aplicación de un método de medición.

Por ultimo, fue introducido un marco para especificar el proceso SFP que es presentado en el Capítulo 5.

La aplicabilidad de un método de medición, la repetibilidad de los resultados de las mediciones, el costo de ejecutar el proceso en términos de tiempo, esfuerzo y capacitación son afectados por la calidad de la documentación del proceso. El Software Process Framework provee las herramientas para seleccionar la información que debe ser documentada y organizarla de modo que facilite la comunicación, la capacitación, el mejoramiento del proceso y el reuso de la documentación.



## Capítulo 3

### Estado del arte

La medición del tamaño del software en las etapas tempranas del proceso de desarrollo es una actividad clave para estimar y planificar un proyecto. El Análisis de Puntos Función (FPA – Function Points Analysis) es una técnica eficiente para medir el tamaño del software a partir de los documentos de requerimientos. En las etapas tempranas del ciclo de vida del software, cuando los requerimientos funcionales no están completamente documentados, la aplicación del FPA se torna difícil o aún imposible. Una solución al problema es provista por las técnicas de estimación de Puntos Función (FP – Function Points), las cuales permiten obtener un valor aproximado del tamaño con una menor inversión en tiempo y esfuerzo y aún con documentación insuficiente. Estas ventajas, obviamente, son neutralizadas por una casi inevitable menor exactitud frente a los resultados de la medición estándar de FP.

En este capítulo se presenta una revisión del estado del arte en el campo de las técnicas de medición o estimación del tamaño funcional de los artefactos producidos en las primeras etapas del desarrollo de software. Específicamente, se apunta a los enfoques basados en el método IFPUG.

#### 3.1. Introducción

La introducción de las métricas de tamaño funcional permitió superar las limitaciones de las métricas de tipo tecnológicas, como las Líneas de Código Fuente (SLOC – Source Lines of Code), sin embargo, a nivel práctico el proceso de adopción ha sido bastante lento debido a la gran cantidad de mediciones de código disponibles en las organizaciones de software y al escepticismo con que fueron recibidas en algunos ambientes técnicos [Meli 1998]. No obstante, las métricas de Puntos Función evolucionaron en el tiempo, principalmente en los últimos años, debido al auge de las técnicas Orientadas a Objetos (OO – Object Oriented) y las aplicaciones Web, siendo actualmente la técnica más usada. Hasta el presente se han propuesto múltiples enfoques orientados a la Medición del Tamaño Funcional (FSM – Functional Size Measurement).

Las técnicas de FPA requieren, para la efectiva medición, un nivel de detalle en la documentación que puede no estar disponible en las etapas más tempranas del ciclo de vida del software. Distintos modelos de estimación de FP intentan salvar esta deficiencia.

La revisión que se presenta en este capítulo está orientada específicamente a las técnicas de medición/estimación del tamaño funcional del software basadas en el FPA tradicional y que son aplicables a los artefactos generados en las fases iniciales del desarrollo de software. Quedan fuera del alcance de esta revisión las propuestas basadas en documentos de diseño o de etapas posteriores. Los enfoques son agrupados en dos categorías principales: *métodos de medición* del tamaño funcional y *métodos de estimación* del tamaño funcional. La categoría *método de medición* comprende al método IFPUG, por ser la base del desarrollo de la propuesta presentada en esta tesis, y un conjunto de enfoques que son extensiones de IFPUG, pues incorporan reglas que permiten su aplicación a diferentes tipos de documentos.

La clasificación de los *métodos de estimación* está basada en la propuesta de Meli y Santillo (1999). Este modelo general para la clasificación de los métodos de estimación fue definido considerando que un método de estimación puede ser representado como un sistema entrada-proceso-salida (Figura.3.1). Las variables de entrada son, esencialmente,

la información sobre la aplicación en estudio mientras que la variable de salida es el tamaño funcional expresado en FP. Tanto las variables de entrada como las intermedias son medidas mediante el uso de métricas consistentes.

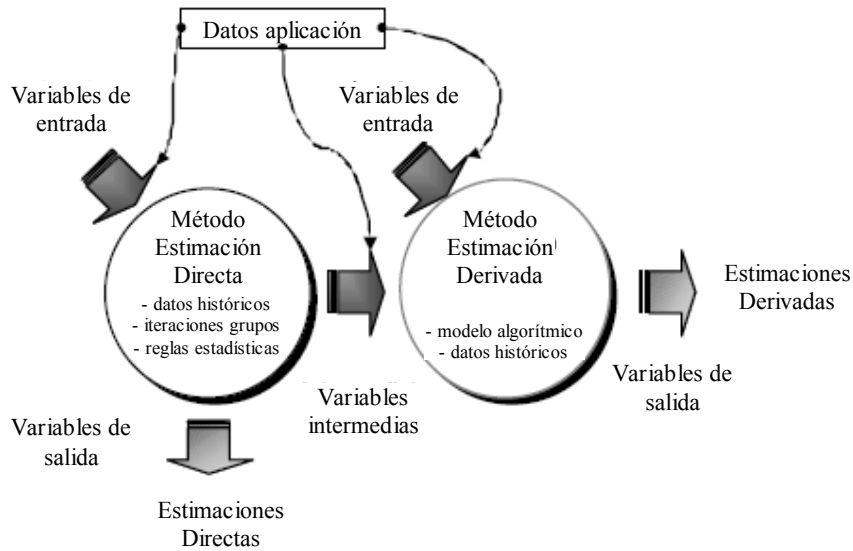


Figura 3.1. Clasificación de los métodos de estimación (traducido de [Meli y Santillo 1999])

Los métodos de estimación pueden ser clasificados en dos categorías: *Estimación Directa* y *Estimación Derivada*. La primera agrupa a los métodos basados en la intuición y el razonamiento analógico, donde el resultado en FP se obtiene directamente, por lo general a través de las predicciones realizadas por uno o más expertos. Tienen la ventaja que se pueden obtener los FP con un esfuerzo mínimo, a expensas de una menor exactitud pues son completamente dependientes de la pericia o experiencia del experto (por ejemplo, el experto consultado “predice” o “adivina” los FP de acuerdo a su intuición o experiencia con otros proyectos, aunque hay propuestas superadoras, como la técnica Delphi). La segunda agrupa a los métodos algorítmicos sustentados en modelos teóricos o estadísticos. Estos últimos producen estimaciones más exactas, objetivas y repetibles pero dependen fuertemente de la disponibilidad de datos históricos (estadísticos). La principal diferencia entre la estimación directa y la derivada es que en estas últimas la estimación es realizada sobre diferentes atributos del software que están correlacionados con los FP [Meli y Santillo 1999]. Esta revisión comprende a un conjunto de métodos pertenecientes a la categoría Métodos de Estimación Derivada.

En la Tabla 3.1 se presenta un resumen de los enfoques de medición y estimación que fueron seleccionados.

Tabla 3.1. Clasificación de los enfoques de medición/estimación de Puntos Función

		IFPUG FPA	
		Medición	Extensión de IFPUG
Tamaño Funcional	Estimación	Derivada o de Modelo Algorítmico	Extrapolación de medidas - Modelo basado en ILF - The Rule of the “Thirties” - FP Prognosis - NESMA Indicative FP - Propuesta de ISBSG <sup>(1)</sup> <sup>19</sup> - Early Function Point Method (EFPM) Estimación de la Complejidad Media - Propuesta de ISBSG <sup>(2)</sup> - NESMA Estimated FP - Simplified Method - FAST Function Points - Function Points Simplified (FPS) Early & Quick Function Points (E&Q FP)

La evaluación de los enfoques se realiza de acuerdo a un esquema *ad hoc* organizado en base a un conjunto de preguntas (ver cuadro adjunto debajo de este párrafo) que facilitan la determinación de las variables a observar y el diseño de una plantilla. Básicamente los aspectos considerados están relacionados a la *fase* del proceso de desarrollo en que puede ser usado el enfoque, al *producto* de software al que puede ser aplicado el enfoque, al *proceso* para determinar el tamaño funcional y al *resultado* de la medición/estimación.

Pregunta	Variable
¿A qué clase de software se puede aplicar?	<i>Dominio funcional</i>
¿Cuán temprano en el ciclo de vida del software se puede utilizar?	<i>Fase</i>
¿Qué producto de software se usa como entrada?	<i>Artefacto</i>
¿Cómo se identifican, clasifican y cuantifican los ítems funcionales en el producto de software?	<i>Reglas</i>
¿Cómo se obtiene el tamaño funcional?	<i>Función</i>
¿Está automatizado el proceso de medición/estimación?	<i>Herramienta</i>
¿Cómo fue verificado el enfoque?	<i>Validación</i>
¿Cuál es el error de la medición/estimación?	<i>Exactitud</i>

Este esquema se originó a partir de los requerimientos de ISO 14143-1 [ISO 1998] para un método de FSM (dominio funcional, artefacto<sup>20</sup>, reglas, función de medición<sup>21</sup>) y a la adopción de algunos de los criterios para evaluar un método de estimación/medición

<sup>19</sup> Los superíndices (1) y (2) son usados con la finalidad de diferenciar el nombre de ambas propuestas.

<sup>20</sup> ISO 14143-1 no usa este término. Establece que un método de FSM está basado en una “representación de los requerimientos del usuario desde la perspectiva de los usuarios”.

<sup>21</sup> Una *función de medición* es un algoritmo o cálculo que combina dos o más medidas [ISO 2002a]. ISO 14143-1 establece que un método de FSM debe “derivar el tamaño funcional a través de la valoración de los BFCs”. El concepto de función de medición está implícito, pues el valor del tamaño funcional se obtiene a partir de los valores acumulados de cada tipo de BFC.

(fase, validación, exactitud, dominio funcional, artefacto, reglas, herramienta de soporte) considerados por otros autores [Meli y Santillo 1999], [Ebert *et al.* 2005], [Abrahão 2004] y [Condori 2007], cuyos trabajos representaron un material de referencia para esta revisión. A continuación se presentan los ítems que componen la plantilla usada para la evaluación.

- *Dominio funcional*
- *Fase*
- *Artefacto*
- *Reglas*
- *Función de medición*
- *Herramienta de soporte*
- *Validación*
- *Exactitud*

En los siguientes párrafos se describen las variables que integran la plantilla.

*Dominio funcional*: clase de software al que se puede aplicar el enfoque. Por ejemplo, Sistemas de Información de Gestión, sistemas de tiempo real.

*Fase*: etapa más temprana del ciclo de vida del software en que se puede aplicar el enfoque. Por ejemplo, estudio de factibilidad, análisis de requerimientos.

*Artefacto de software*: producto de software que se puede usar como entrada al proceso de medición/estimación. Por ejemplo, Especificación de Requerimientos del Software (SRS – Software Requirements Specification), Casos de Uso (UC – Use Case), modelo de objetos.

*Reglas*: pautas establecidas en el contexto de un método de medición/estimación para identificar, clasificar y/o cuantificar los ítems relevantes para el tamaño funcional en el producto de software. Por ejemplo, en IFPUG, una regla para identificar Archivos Lógicos Internos (ILF – Internal Logical File) establece: “El grupo de datos es mantenido por un proceso elemental, dentro de los límites de la aplicación que se está midiendo” [IFPUG 2000].

*Función de medición*: función propuesta para calcular el tamaño funcional. Las medidas involucradas en el cálculo son determinadas de acuerdo al método de medición/estimación. Por ejemplo, la función de medición de IFPUG combina las contribuciones en FP de los cinco tipos de Componentes Funcionales Básicos (BFC – Base Functional Component).

*Herramienta de soporte*: software que soporta un proceso de medición/estimación. Una herramienta automática realiza el proceso con una mínima intervención del usuario (por ejemplo, para ingresar los datos), mientras que aquéllas que requieren la interacción con el usuario en uno o más pasos del proceso (por ejemplo, para identificar o clasificar ítems) son consideradas semiautomáticas. De acuerdo a la clasificación de IFPUG<sup>22</sup>, las primeras se corresponden con el Tipo 3 (Automatic Counting of Function Points) y las mencionadas en segundo lugar con el Tipo 2 (Expert System That Aids Counting of Function Points).

---

<sup>22</sup> <http://www.ifpug.org/certification/software.htm>.

*Validación<sup>23</sup> del enfoque*: estrategia adoptada para evaluar la aplicación del enfoque (no es considerada la validación teórica del enfoque). Por ejemplo, caso de estudio, proyecto real, experimento.

*Exactitud de la medición/estimación*: es definida como “la proximidad entre el resultado de una medición y el *valor verdadero* de la medida”. Este *valor verdadero* podría ser obtenido por una medición “perfecta”, sin errores. En la práctica esto es imposible, por lo tanto se usa un *valor verdadero convencional* o *valor de referencia* [JCGM 2008]. En el caso de la medición del software, el valor de referencia puede ser el resultado de una medición realizada por un especialista en un método. En el contexto de esta evaluación, el valor de referencia es el valor obtenido con el método IFPUG. Por ejemplo, si el valor de referencia para un determinado sistema es 100 FP y un método produce estimaciones con una exactitud de  $\pm 30\%$ , se puede esperar un tamaño funcional comprendido entre 70 y 130 FP. Es expresada en términos del error relativo, esto es, cuanto menor es el error relativo, mayor es la exactitud de la medición.

$$\text{Error relativo} = \left| \frac{\text{Valor}_{\text{real}} - \text{Valor}_{\text{estimado}}}{\text{Valor}_{\text{real}}} \right|$$

También puede ser multiplicado por 100 y expresado como un porcentaje (denominado error porcentual o error relativo porcentual). Como se observará en el resto del capítulo, en algunos casos es omitido el signo ( $\pm$ ) para respetar la información suministrada por los autores; sin embargo, por ejemplo, un error relativo del 5% debe ser entendido como un error de  $\pm 5\%$ .

### 3.2. Medición de Puntos Función

La medición de FP se realiza en el contexto de un modelo abstracto del software que consiste de transacciones y archivos lógicos. El FPA define dos pasos principales: identificar los ítems significativos para el tamaño funcional desde los documentos de requerimientos y asociar esos ítems con valores numéricos.

En la literatura pueden encontrarse muchos enfoques basados en FP. Sin embargo sólo cinco son reconocidos como estándares por ISO: IFPUG 4.1 Unadjusted (ISO/IEC 20926:2003), MkII (ISO/IEC 20968:2002), COSMIC-FFP V. 2.2 (ISO/IEC 19761:2003), NESMA 2.1 (ISO/IEC 24570:2005) y FISMA 1.1 (ISO/IEC 29881:2008). El comité ISO responsable de los estándares para la Ingeniería del Software (ISO/IEC JTC1 SC7) en el año 2000 resolvió no opinar sobre los méritos relativos entre los cuatro métodos (en ese entonces), permitiendo que los métodos fueran presentados para su aprobación como estándares internacionales y “dejar que el mercado decida” cual método usar [COSMIC 2003].

En las siguientes secciones se describen el método IFPUG FPA y las propuestas de medición enumeradas en la Tabla 3.1 y se analizan en base a la plantilla de evaluación propuesta en la sección 3.1.

#### 3.2.1. IFPUG FPA

Las características de este método, basado en el enfoque de Albrecht, han sido descritas en el Capítulo 2, sección 2.2.3. Como fue señalado, de acuerdo a la definición de tamaño funcional provista por ISO 14143-1 [ISO 1998], el estándar ISO/IEC

<sup>23</sup> confirmación a través de un examen sobre un determinado ítem y presentación de evidencias objetivas de que cumple los requisitos para un uso establecido [ISO 1993].

20926:2003 considera exclusivamente el valor en Puntos Función no Ajustados (UFP – Unadjusted Function Points). Esta métrica puede ser aplicada a través del ciclo de vida del desarrollo de software, desde la definición de los requerimientos hasta el final del proceso, después de la implementación. De acuerdo a las técnicas vigentes en la época de su desarrollo, fue concebido para ser aplicado sobre las especificaciones de requerimientos tradicionales; esfuerzos posteriores de IFPUG y otros ampliaron el enfoque para aplicarlo a otros productos, como por ejemplo, las especificaciones orientadas a objetos. En la comunidad de IFPUG se definió un caso de estudio que ilustra la aplicación de sus reglas a los productos del análisis y diseño Orientado a Objetos. El caso de estudio (Case Study 3 - Object Oriented Analysis, Object Oriented Design) utiliza modelos de objetos, considerando a las clases como archivos lógicos y a los métodos de las clases como transacciones [Fetcke *et al.* 1997]. Esta propuesta ha sido cuestionada por considerarse que no captura adecuadamente el tamaño funcional del software Orientado a Objetos. Algo similar ocurre con el software para tiempo real, ambiente desde el cual también señalan las limitaciones del FPA para ese dominio [Oligny *et al.* 1998], [Gencel *et al.* 2005]. Esto derivó en el desarrollo de nuevas propuestas basadas en el enfoque de Albrecht [Abran *et al.* 2000], algunas de las cuales se presentan en las siguientes secciones.

El enfoque de Albrecht fue originalmente desarrollado en base a 22 proyectos de Sistemas de Información de Gestión (MIS – Management Information Systems). Debido a que IFPUG es la técnica de FPA más usada en la industria, los resultados de su aplicación han sido ampliamente validados, lo que supera largamente su inexistencia en el enfoque original. Según Jones (2008b), el método IFPUG ha sido usado para más proyectos que todas las variantes del FPA juntas, con excepción del *backfiring*<sup>24</sup>. Hasta 2008 probablemente haya sido usado en más de 50.000 proyectos sólo en los Estados Unidos y otros 30.000 proyectos en Europa, la Cuenca del Pacífico y América del Sur.

La experiencia acumulada ha permitido determinar que la exactitud de las mediciones es de  $\pm 10\%$  con relación al proyecto finalizado. Jones (2008b) sugiere que la exactitud de las mediciones realizadas por especialistas certificados (CFPS)<sup>25</sup> es del 5%. No existen herramientas - certificadas por IFPUG - que realicen el proceso en forma automática (Tipo 3 según la clasificación provista por IFPUG).

## Evaluación

*Dominio funcional:* MIS.

*Fase:* análisis de requerimientos.

*Artefacto de software:* SRS.

*Reglas:* reglas para identificar, clasificar y cuantificar los cinco tipos de BFCs.

*Función de medición:*  $UFP = \sum_{i \in 1..5} \sum_{j \in 1..3} \#ElementTypes_{ij} W_{ij}$  [Morasca 2001]

donde:  $i$  es un tipo de elemento,  $j$  su valor de complejidad y  $W_{ij}$  es el peso asociado.

*Herramienta de soporte:* no.

*Validación del enfoque:* no fue validado en la propuesta original, ampliamente comprobado en la práctica.

<sup>24</sup> Esta técnica, desarrollada por Jones (1995), permite convertir líneas de código a FP.

<sup>25</sup> Certified Function Point Specialist (<http://www.ifpug.org/certification/cfps.htm>): certificación otorgada por IFPUG como reconocimiento formal de un nivel de conocimientos en el área del FPA.

*Exactitud de la medición:* la exactitud del método puede variar entre  $\pm 5\%$  y  $\pm 10\%$ , en este caso no es considerada porque el resultado de la medición realizada con este método es el valor de referencia para la comparación con las demás propuestas.

### 3.2.2. Extensión de IFPUG FPA

Esta categoría incluye algunas propuestas de FSM que incorporan nuevas reglas o adaptan las reglas de IFPUG a dominios funcionales o documentos de requerimientos específicos. En el campo particular de la FSM de sistemas OO se han propuesto extensiones para medir el software OO, algunas de las cuales aplican IFPUG sobre especificaciones de requerimientos expresadas en notación UML (Unified Modeling Language).

En lo que sigue se presenta el análisis de las propuestas seleccionadas para esta categoría.

#### 3.2.2.1. Propuesta de Fetcke *et al.*

Fetcke *et al.* (1997) aplican el FPA estándar para medir el software desarrollado con el método Object Oriented Software Engineering (OOSE)<sup>26</sup> [Jacobson 2003] en la fase de análisis de requerimientos. El enfoque se basa en la asociación de los conceptos del método OOSE (basado en UC) con los del FPA. Esta asociación fue formulada mediante un conjunto de reglas que soportan las reglas de IFPUG FPA. Los actores de los UC son los usuarios del sistema, los UC son asociados con las funciones transaccionales y las clases con los archivos lógicos (los autores proponen usar las clases del modelo de objetos del dominio o del modelo del análisis).

Con esta propuesta Fetcke *et al.* (1997) demostraron la aplicabilidad del FPA estándar sobre especificaciones desarrolladas con el método OOSE y reafirmaron la hipótesis que el FPA es independiente de la tecnología usada para la implementación del software y puede ser usado con el paradigma OO.

El enfoque fue aplicado sobre tres proyectos de la industria desarrollados con el enfoque OOSE, pero los resultados no fueron relacionados con otras métricas, como por ejemplo, los FP tradicionales [Antoniol *et al.* 1999]. En consecuencia no existen datos sobre la exactitud de la medición. En comparación con otros enfoques, los autores señalan las ventajas (por ejemplo, el uso de las reglas del estándar IFPUG) y las limitaciones de la propuesta (por ejemplo, sólo puede ser aplicado sobre modelos de requerimientos basados en OOSE).

#### Evaluación

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* modelo de UC y modelo de objetos generados con el método OOSE.

*Reglas:* IFPUG adaptadas (por ejemplo, para asociar los actores de los UC con los usuarios del sistema, los UC con las funciones transaccionales y las clases con los archivos lógicos).

*Función de medición:* IFPUG.

---

<sup>26</sup> Es un predecesor de UML [Rational Corp. 1997].

*Herramienta de soporte:* no.

*Validación del enfoque:* no.

*Exactitud de la medición:* no especificado.

Longstreet (2000), (2001) propone un enfoque similar para identificar las funciones transaccionales, pero sugiere que los archivos lógicos pueden ser determinados desde el modelo lógico de datos (diseño de base de datos relacional).

### **3.2.2.2. Propuesta de Harput *et al.***

Harput *et al.* (2005) aplican el FPA sobre especificaciones de requerimientos OO. El enfoque está orientado exclusivamente a los requerimientos y es aplicado sobre los UC, los diagramas de secuencia y de clases. Para ello definieron un conjunto de reglas que permiten transformar un modelo de requerimientos OO en un modelo del FPA tradicional. Las reglas definidas asocian a las clases con las funciones de datos y a los UC y los requerimientos funcionales con las funciones transaccionales. El proceso es soportado por una herramienta semiautomática.

Los autores destacan la ventaja de usar el FPA debido a que los resultados obtenidos pueden ser comparados con los existentes del método FPA. El enfoque fue aplicado al Caso de estudio 3 de IFPUG y a un caso del mundo real, que es justamente la extensión de la herramienta RETH (Requirements Engineering Through Hypertext) que soporta este enfoque. Como aspecto destacable, para el caso 3 de IFPUG uno de los autores aplicó ingeniería reversa a partir de los requerimientos del usuario para construir un modelo acorde al método RETH, que luego fue usado para la medición. En ambos casos se obtuvieron resultados compatibles con las mediciones realizadas por especialistas en IFPUG. La exactitud de la medición fue de 10% para el primer caso y de 14% para el segundo.

### **Evaluación**

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* SRS OO (diagramas de UC, diagramas de secuencia y diagramas de clases).

*Reglas:* IFPUG adaptadas (por ejemplo, para asociar las clases con las funciones de datos y los UC y requerimientos funcionales con las funciones transaccionales).

*Función de medición:* IFPUG.

*Herramienta de soporte:* sí, semiautomática.

*Validación del enfoque:* sí, un caso de estudio y un caso real.

*Exactitud de la medición:* entre 10% y 14%.

### **3.2.2.3. Propuesta de Tavares *et al.***

Tavares *et al.* (2002) establecen una correspondencia entre los conceptos de IFPUG y UML que permite identificar las funciones transaccionales a partir de los diagramas de UC y las funciones de datos desde el diagrama de clases y el diagrama de UC. El enfoque es similar al propuesto por Fetcke *et al.* (1997) y fue aplicado a un caso de estudio basado en un sistema real, en una empresa de Tecnología de la Información. El proyecto fue desarrollado por dos equipos diferentes que trabajaron con una SRS tradicional y una



SRS OO y aplicaron el enfoque sobre ambas especificaciones obteniendo prácticamente el mismo resultado. Al concluir el proyecto fue medido el tamaño del proyecto y comparado con la estimación inicial, con resultados muy aproximados. Los autores destacan que esta propuesta ha sido usada en más de 120 proyectos de la organización, lo que les permitió verificar la exactitud de las estimaciones, pero no mencionan específicamente cuál es su valor.

### **Evaluación**

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* SRS OO (diagramas de UC y diagramas de clases).

*Reglas:* IFPUG adaptadas (por ejemplo, para identificar las funciones transaccionales a partir de los diagramas de UC y las funciones de datos desde el diagrama de clases y el diagrama de UC).

*Función de medición:* IFPUG.

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, un proyecto real.

*Exactitud de la medición:* no especificado.

#### **3.2.2.4. OO-Method Function Points (OOmFP)**

Este enfoque [Abrahão *et al.* 2004a] fue desarrollado para medir el tamaño funcional de sistemas OO generados por un método de producción automática de software que se inicia con una fase de ingeniería de requerimientos denominado OO-Method. Dicho proceso es soportado por una herramienta que incluye un módulo para la medición de FP [Torres y Calatrava 2003].

Este método se ajusta a las reglas de IFPUG FPA y las redefine en términos de los conceptos usados en OO-Method. El proceso comienza con el desarrollo de un esquema conceptual OO-Method a partir de la SRS del usuario. Este esquema incluye un Modelo de Objetos (modelo estructural del dominio), un Modelo Presentación (patrones para describir la interacción sistema-usuario) y los Modelos Dinámico y Funcional que especifican el comportamiento de un sistema OO. Desde el Modelo de Objetos son identificadas las funciones de datos y desde el Modelo Presentación las funciones transaccionales. La complejidad de estas últimas es determinada mediante la información distribuida en los cuatro modelos. Cada función identificada es asociada con un nivel de complejidad (*baja, media, alta*) y ésta con el correspondiente valor numérico según las tablas de IFPUG. Finalmente, mediante un proceso de agregación es cuantificado el tamaño funcional.

La validación fue realizada mediante un experimento donde fueron comparados OOmFP e IFPUG con respecto a una serie de variables (por ejemplo, la reproducibilidad, exactitud, etc.). En lo referido a la exactitud, el estudio permitió determinar que el enfoque produce valores similares a IFPUG. La exactitud de las mediciones realizadas sobre un mismo sistema por 20 estudiantes varía entre 1% y 13%.

### **Evaluación**

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* esquema conceptual OO-Method.

*Reglas:* IFPUG adaptadas (por ejemplo, para identificar las funciones de datos desde el Modelo de Objetos y funciones transaccionales desde el Modelo Presentación).

*Función de medición:*  $OOmFP = OOmFP_{Data} + OOmFP_{Transaccio\ n}$

donde

$$OOmFP_{Data} = \sum_{i=1}^n OOmFP_{class_i} + \sum_{j=1}^m OOmFP_{legacy\ view_j}$$

$$OOmFP_{Transaccio\ n} = \sum_{i=1}^n OOmFP_{service_i} + \sum_{j=1}^m OOmFP_{IU_j | PIU_j | MDIU_j}$$

*Herramienta de soporte:* si.

*Validación del enfoque:* sí, experimento de laboratorio, un caso.

*Exactitud de la medición:* entre 1% y 13%.

### 3.2.2.5. Propuesta de Kusumoto *et al.*

Kusumoto *et al.* (2000) proponen un conjunto de reglas y una herramienta de medición para aplicar IFPUG a las especificaciones de requerimientos generadas por un sistema de análisis de requerimientos OO denominado REQUARIO, desarrollado en Hitachi Ltd. El enfoque describe un conjunto de reglas basadas en el estándar IFPUG y adaptadas para extraer, clasificar y cuantificar las funciones de datos y funciones transaccionales. La validación fue realizada usando la herramienta para medir tres aplicaciones de gestión y comparando con la medición manual realizada por especialista en IFPUG. Si bien la exactitud no es especificada, los autores mencionan que se obtuvieron los mismos resultados con ambas mediciones. Por otro lado, la propuesta fue aplicada sobre varios sistemas reales produciendo resultados “considerablemente adecuados” pero por razones de confidencialidad no son publicados los datos. En el marco de este enfoque se planea expandir las reglas para que puedan ser aplicadas a especificaciones de requerimientos OO en general.

#### Evaluación

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* SRS OO generada por REQUARIO.

*Reglas:* basadas en el estándar IFPUG y adaptadas (por ejemplo, para extraer, clasificar y cuantificar las funciones de datos y funciones transaccionales).

*Función de medición:* IFPUG.

*Herramienta de soporte:* si.

*Validación del enfoque:* sí, tres sistemas reales.

*Exactitud de la medición:* no especificado.

### 3.2.2.6. Propuesta de Choi *et al.*

Choi *et al.* (2006) describen una metodología para extraer los FP desde el texto obtenido usando la técnica de Elicitación de Requerimientos basada en objetivos y escenarios. El método apunta a establecer y mantener la *traceability* entre los FP y el

texto de requerimientos. Las reglas de extracción de FP están basadas en el estándar IFPUG y fueron adaptadas para aplicarlas a estos documentos de requerimientos. El proceso consta de tres pasos que permiten extraer elementos del contexto del sistema (para determinar el límite del sistema, identificar usuarios y aplicaciones externas), funciones de datos y funciones transaccionales desde el modelo de objetivos y escenarios. Luego son usadas las tablas de IFPUG para calcular los FP no ajustados. Los autores aplicaron el enfoque sobre dos sistemas y planean desarrollar una herramienta de soporte y la validación empírica. La exactitud de la medición no es especificada.

### **Evaluación**

*Dominio funcional:* MIS

*Fase:* Elicitación de Requerimientos.

*Artefacto de software:* documentos de requerimientos generados con la técnica de objetivos y escenarios.

*Reglas:* basadas en el estándar IFPUG 4.1.1 y adaptadas para extraer elementos del contexto y extraer, clasificar y cuantificar las funciones de datos y funciones transaccionales.

*Función de medición:* IFPUG.

*Herramienta de soporte:* no.

*Validación del enfoque:* no especificado.

*Exactitud de la medición:* no especificado.

#### **3.2.2.7. Use Case Points**

El modelado de UC es una técnica ampliamente usada en la industria para describir y capturar los requerimientos funcionales de un sistema de software. Diferentes autores han desarrollado enfoques para aplicar las reglas del FPA estándar sobre los UC, algunos de los cuales han sido mencionados previamente, como por ejemplo, Fetcke *et al.*, Harput *et al.* y Tavares *et al.* La técnica Use Case Points (UCP), desarrollada para estimar el esfuerzo desde los UC, tiene sus orígenes en el FPA, específicamente es una extensión del FPA estándar y MkII FPA [Banerjee 2001]. El primero de estos enfoques fue propuesto por Karner (1993). Arnold y Pedross (1998) desarrollaron un método UCP similar al de Karner para una compañía específica.

A diferencia del FPA clásico, la medición de UCP se basa en el conteo de los actores y las transacciones incluidas en los modelos de UC. El primer paso consiste en categorizar los actores y UC según su complejidad (*simple*, *medio* o *complejo*) y asignar los correspondientes factores de peso. La categoría de un UC depende del número de transacciones (incluyendo los caminos alternativos). La suma de los productos del total de actores en cada categoría y su respectivo factor de peso produce el *unadjusted actor weight (UAW)*. De modo similar se calcula el *unadjusted use case weights (UUCW)*. El valor *unadjusted use case points (UUCP)* se obtiene sumando UAW y UUCW [Damodaran y Washington 2002]. Por la misma razón que con el FPA estándar no debería ser considerado el ajuste por complejidad técnica.

Debido a la popularidad de UML y los UC, en la actualidad existen cientos de usuarios y miles de proyectos medidos con la métrica de UC. Sin embargo, no hay un estándar o práctica estandarizada para el cálculo de UCP ni una organización de usuarios

como IFPUG [Jones 2008b], lo que puede ocasionar discrepancias entre las organizaciones sobre cómo desarrollar los UC y cómo calcular los UCP.

Algunas dificultades con el uso de UCP están relacionadas directamente con el modelo de UC. La aplicación efectiva del método requiere UC bien escritos, bien estructurados y con apropiado nivel de detalle. Las descripciones de los UC varían en el detalle y el estilo, no hay estándares que especifiquen el nivel de detalle correcto para todos los proyectos. Por ejemplo, el número de transacciones en un UC determina su tamaño. Es muy difícil determinar el nivel de detalle apropiado en cada transacción cuando se construye un UC y esto tiene un impacto directo sobre la estimación obtenida por UCP. Los UC *included* y *extended* son omitidos en algunos estudios y considerados en otros. Se ha demostrado que el método es adecuado para aplicaciones relativamente pequeñas, pero no que pueda ser usado para medir todo tipo de software [Damodaran y Washington 2002].

Karner (1993) aplicó su propuesta a 3 proyectos, pero no efectuó una validación de los resultados con otra métrica. Arnold y Pedross (1998) aplicaron el enfoque sobre 23 proyectos del dominio MIS y compararon los resultados con los respectivos FP de la medición estándar, observando una diferencia de  $\pm 10\%$  entre ambas mediciones. Sin embargo, Jones (2008b) sugiere que el tamaño medido en UCP puede ser hasta un 25% más grande que el de IFPUG.

### Evaluación

*Dominio funcional:* MIS

*Fase:* análisis de requerimientos.

*Artefacto de software:* modelo de UC.

*Reglas:* para clasificar y asignar un factor de peso a los actores y los UC.

*Función de medición:* 
$$UUCP = \sum_{i=1}^6 n_i * W_i$$

donde  $n_i$  es el número de ítems de la variedad  $i$  y  $W_i$  es el peso de  $i$  [Karner 1993].

*Herramienta de soporte:* si, varias [Kusumoto *et al.* 2004], [Anda *et al.* 2005].

*Validación del enfoque:* no en la propuesta de Karner; 23 proyectos en la propuesta de Arnold y Pedross.

*Exactitud de la medición:*  $\pm 10\%$ .

### 3.2.3. Resumen sobre la evaluación de los Métodos de medición

En los siguientes párrafos se sintetizan los resultados de la evaluación de los métodos de medición de acuerdo al esquema propuesto. La Tabla 3.2 (al final del capítulo) presenta el resumen de la evaluación de cada método de medición.

- Dominio funcional: todas las propuestas seleccionadas han sido diseñadas para el software MIS.
- Fase más temprana del proceso de desarrollo de software en que puede ser aplicada: la mayoría de las propuestas pueden ser aplicadas a partir de la fase de análisis de requerimientos. Con excepción del enfoque de Choi *et al.*, no se encontraron otras propuestas para medir el tamaño funcional en la Elicitación de Requerimientos. En

este sentido se coincide con Choi *et al.*, quienes destacan que los métodos de FPA existentes no se prestan demasiada atención a la Elicitación de Requerimientos.

- Artefacto de software: en general se utiliza la SRS, la que varía en su forma de representación dependiendo de la metodología de desarrollo aplicada. El enfoque de Choi *et al.* es el único que utiliza documentos de texto en lenguaje natural producidos con la técnica de objetivos y escenarios.
- Reglas propuestas: la mayoría de los enfoques define reglas adicionales para asociar un modelo específico del software, como por ejemplo, los UC, con los conceptos de IFPUG. La excepción es UCP, que introduce una visión algo distinta del tamaño funcional.
- Función de medición: la mayoría de los enfoques utiliza la función propuesta por IFPUG. Las excepciones son OOmFP y Use Case Points.
- Herramienta de soporte: el 50% de los enfoques no dispone de un software para facilitar el proceso de medición. Entre los que proveen una herramienta, en algunos casos no se especifica si es automática o semiautomática.
- Validación del enfoque: con excepción de IFPUG, cuya amplia difusión en la industria ha permitido validar sus resultados a través de la aplicación sobre un gran número de proyectos, la mayoría de las propuestas ha sido evaluada mediante un número bastante reducido de casos de estudio o proyectos reales (entre 1 y 3, excepto UCP con 23 casos).
- Exactitud de la medición: el 50% de las propuestas no presenta conclusiones acerca de la exactitud de las mediciones. En los demás casos, ésta varía entre  $\pm 1\%$  y  $\pm 14\%$ .

### 3.3. Estimación de Puntos Función

Como ya fue mencionado, las técnicas de estimación de FP son una herramienta muy valiosa cuando no es posible medir los FP con un método estándar. Esta situación no sólo ocurre en las fases tempranas de un proyecto de desarrollo, cuando la información (a veces incompleta) contenida en los artefactos disponibles es insuficiente, sino también cuando se evalúa el reemplazo o la mejora de un software existente y no se cuenta con la documentación o el tiempo y los recursos requeridos [Meli y Santillo 1999]. En esas condiciones, los FP estimados pueden ser sólo una aproximación del tamaño real de un proyecto, sin embargo, constituyen una base para estimar el esfuerzo, el costo y la agenda; sin duda, esta práctica es mucho más conveniente que avanzar en el desarrollo sin ninguna planificación a la espera de tener suficiente información para una medición como lo prescribe el método estándar. Vale destacar que la estimación de FP debería ser complementada con una adecuada comunicación a los potenciales usuarios sobre el origen del tamaño que se les suministra, es decir, especificar que se trata de un valor aproximado que puede involucrar un cierto grado de incerteza.

Específicamente en lo referido a los métodos de estimación derivada que se propone evaluar, por lo general, determinan el tamaño en FP aplicando modelos estadísticos basados en datos de proyectos previos. Algunos de los métodos son examinados en las siguientes secciones.

En las siguientes secciones se describen los métodos y enfoques de estimación enumerados en la Tabla 3.1 y se analizan en base a la plantilla de evaluación propuesta en la sección 3.1.

### 3.3.1. Extrapolación de medidas

Estas técnicas asumen que se puede medir algún componente (generalmente los ILFs) de una aplicación y derivar los FP basándose en la estadística [Meli y Santillo 1999]. Este tipo de métodos debería ser cuidadosamente analizado para determinar su aplicabilidad en un dominio particular. En muchos casos deben ser adaptados para reflejar la distribución de los FP en un ambiente particular en lugar de usar los datos de repositorios públicos, como ISBSG. Estas técnicas son fáciles de usar pero bastante propensas a errores, ya que un objeto no identificado puede implicar un gran número de FP no tenidos en cuenta. Por lo tanto, se requiere una correcta identificación del componente (por ejemplo, los ILFs), lo cual usualmente no es posible cuando los requerimientos son inciertos o inestables [Iorio *et al.* 2007].

#### 3.3.1.1. Modelo basado en ILF

Tichenor<sup>27</sup> estableció un modelo de estimación (ILF Model) de los UFP basado en la fuerte correlación existente entre el número de ILF y los UFP medidos con IFPUG, para una muestra de sistemas (no se encontró información sobre el tamaño de los sistemas usados para definir el modelo). Si bien se afirma que este modelo tiene un coeficiente de determinación  $r^2 = 0.9482$  [Tichenor 1997] cuando es comparado con la medición IFPUG, no se encontró información sobre el número de sistemas usados. Para sistemas transaccionales (Create, Read, Update, Delete) propuso la siguiente fórmula para estimar los UFP [Meli y Santillo 1999]:

$$\text{UFP Est.} = \#\text{ILF} * 14.93 \text{ FP/ILF}$$

El modelo inicial fue desarrollado usando los datos de 27 aplicaciones medidas con el método IFPUG. Puede ser aplicado muy temprano en el ciclo de vida, cuando se dispone del modelo de datos. Actualmente, Tichenor recomienda realizar un análisis de regresión con los datos de las mediciones propias de cada organización para definir el modelo de ILF, usando un mínimo de 30 proyectos [IFPUG 2008]. Si no se dispone de datos para hacerlo, se puede hacer una estimación aproximada con 30-35 UFP por ILF (considerando que hay tres EIs, una EO y una EQ por cada ILF, alguna de las cuales tiene complejidad media o alta) [Tichenor 2008].

#### Evaluación

*Dominio funcional:* MIS.

*Fase:* no especificado explícitamente.

*Artefacto de software:* modelo de datos.

*Reglas:* IFPUG.

*Función de medición:*  $\text{UFP} = \#\text{ILF} * \text{factor (FP/ILF)}$ ; factor: propio de la organización o aproximación con 30 ó 35 FP/ILF.

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, no se encontró información sobre el número de sistemas.

*Exactitud de la estimación:* 10% [IFPUG 2008]

---

<sup>27</sup> El autor de esta tesis no tuvo acceso al documento original de Tichenor.

### 3.3.1.2. The Rule of the “Thirties”

Algunas organizaciones han determinado a partir de la experiencia que un archivo lógico equivale a “aproximadamente treinta” FP para un proyecto de desarrollo. La estimación de los FP del proyecto puede obtenerse multiplicando ese valor aproximado (entre 31 y 35 FP por archivo lógico) por el número de archivos lógicos identificados [ISBSG 2006]. Con esta clase de estimación podría esperarse una variación de  $\pm 30\%$  (no específica si es con relación a la medición estándar). Esta propuesta trabaja con el total de archivos lógicos, sin diferenciar entre archivos lógicos internos y externos. Esta técnica es válida si el proyecto de desarrollo es acorde al perfil de proyectos del repositorio de ISBSG (principalmente del dominio MIS).

#### Evaluación

*Dominio funcional:* MIS.

*Fase:* no especificado.

*Artefacto de software:* modelo de datos.

*Reglas:* -

*Función de medición:*  $FP = \#archivos \text{ lógicos} * FP/archivo \text{ lógico}$  (entre 31 y 35).

*Herramienta de soporte:* no.

*Validación del enfoque:* no especificado.

*Exactitud de la estimación:*  $\pm 30\%$ .

### 3.3.1.3. FP Prognosis

Bundschuh (1998) provee un modelo empírico para estimar los FP, especialmente para sistemas on-line. A partir de la fuerte correlación existente - considerando los datos de 20 proyectos de la compañía CNV AG - entre la suma del número de EIs y EOs (denominada “IO” por el autor) y los FP medidos con el método IFPUG, fue realizado un análisis de regresión lineal resultando la siguiente fórmula:

$$FP = 7.3 \times \#IO + 56$$

Este enfoque permite la estimación temprana cuando los componentes de datos (ILF y EIF) aún no han sido investigados. El autor sugiere usar la técnica para una estimación temprana considerando un error entre 15% a 20% y luego realizar la medición de los FP al finalizar la definición de los requerimientos.

La verificación de los resultados fue realizada comparando la estimación basada en la fórmula de regresión y la medición de FP, sobre la misma muestra de proyectos.

#### Evaluación

*Dominio funcional:* MIS.

*Fase:* estudio de factibilidad.

*Artefacto de software:* no especificado.

*Reglas:* IFPUG.

*Función de medición:*  $FP = 7.3 \times \#IO + 56$ , siendo  $\#IO = \#EI + \#EO$

*Herramienta de soporte:* no

*Validación del enfoque:* no, se usaron los datos de los mismos proyectos.

*Exactitud de la estimación:* entre 15% y 20%.

#### 3.3.1.4. NESMA Indicative FP

NESMA (Netherlands Software Metrics Association) propone esta técnica, también conocida como "*método Dutch*", para la estimación temprana [NESMA 2004]. Es determinada en función del número de ILF<sup>28</sup> y EIF.

$$\text{Indicative Size (UFP)} = 35 \times \#\text{ILF} + 15 \times \#\text{EIF}$$

Esta fórmula fue establecida considerando que como mínimo hay tres EI (agregar, modificar y borrar información en el ILF), dos EO y una EQ por cada ILF y una EO y EQ por cada EIF [Meli y Santillo 1999].

Usando una base de datos de aproximadamente 100 aplicaciones desarrolladas e implementadas fue investigada la exactitud de esta estimación en comparación con la medición *Detailed FP* (muy similar a IFPUG). Aunque los resultados mostraron una buena correlación, en algunos casos se observaron desviaciones importantes (hasta del 50%). Esto significa que se debería ser cuidadoso cuando se usa este tipo de aproximación del tamaño funcional. La ventaja es que fácilmente y en poco tiempo se puede tener una estimación del tamaño de una aplicación [NESMA 2004]. Según Jones (2008b), este método genera resultados que pueden variar en un 15% respecto a los otros dos (*Estimated* y *Detailed*). Este autor también menciona que es probable que los diferentes métodos de NESMA (*Indicative*, *Estimated* y *Detailed*) hayan sido aplicados a más de 3000 proyectos hasta 2008, pero no presenta la información discriminada por método.

#### Evaluación

*Dominio funcional:* MIS.

*Fase:* previa a la definición de los requerimientos.

*Artefacto de software:* modelo de datos.

*Reglas:* NESMA<sup>29</sup>.

*Función de medición:*  $\text{UFP} = 35 \times \#\text{ILF} + 15 \times \#\text{EIF}$

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, 100 proyectos [NESMA 2004].

*Exactitud de la estimación:* no especificado; 15% según Jones (2008b).

---

<sup>28</sup> NESMA e IFPUG identifican los mismos cinco tipos de funciones aunque usan diferente denominación (entre paréntesis el equivalente IFPUG): ILGV (ILF), KGV (EIF), IF (EI), UF (EO), OF (EQ) [NESMA 2008].

<sup>29</sup> NESMA e IFPUG usan las mismas reglas para determinar el tipo y complejidad de las funciones, con unas pocas excepciones, por ejemplo, en NESMA la complejidad de una consulta externa (EQ) es determinada del siguiente modo: la complejidad funcional de la parte de entrada está basada en las reglas de complejidad de las EI y la complejidad de la parte de salida en las reglas de complejidad de las EO. La más compleja de ambas será usada como la complejidad de la EQ [NESMA 2008].



### 3.3.1.5. Propuesta de ISBSG <sup>(1)</sup>

International Software Benchmarking Standards Group (ISBSG) es una organización que mantiene un repositorio de datos de métricas de proyectos de software (Benchmark Release 9<sup>30</sup> de 2005 cuenta con 3850 proyectos [Hill 2006]). Sobre la base de 209 proyectos que fueron medidos con IFPUG 4.0 se establecieron las siguientes relaciones entre los tipos de componentes funcionales de IFPUG (para nuevos proyectos de desarrollo):

ILF 21.7%, EIF 8.5%, EI 28.9%, EO 25%, EQ 15.9%

La estimación del tamaño puede hacerse conociendo el número de ILF (u otro componente). El análisis de los datos del repositorio muestra que la complejidad de la mayoría de los ILF varía desde *baja* a *media* y propone asignarle un valor promedio de 8.6 UFP, de modo que establece:

$$\text{UFP (ILF)} = 8.6 \times \#\text{ILF}$$

$$\text{UFP (total)} = \text{UFP (ILF)} \times 100 / 21.7$$

Si el tamaño va a ser utilizado como entrada para estimar el costo de un proyecto, se sugiere adicionar un 20%-30% para contemplar la funcionalidad que no es evidente en las primeras etapas. La aplicabilidad de la técnica está restringida (ídem sección 3.3.1.2) a proyectos de desarrollo acordes al perfil de proyectos en la base de datos de ISBSG.

Un análisis comparativo [ISBSG 2005] de la estimación temprana frente al tamaño final sobre 130 proyectos de la base de datos de ISBSG permitió determinar que el 50% de las estimaciones están comprendidas dentro del error admisible de 10%, 12% de los proyectos fueron sobreestimados (el error promedio es del 20%) y el 38% fueron subestimados (en algunos casos superaron el 50%). Además se observó que las estimaciones basadas en especificaciones funcionales resultaron más exactas que aquellas realizadas sobre modelos de datos.

#### Evaluación

*Dominio funcional:* MIS.

*Fase:* estudio de factibilidad.

*Artefacto de software:* modelo de datos.

*Reglas:* IFPUG.

*Función de medición:*  $\text{UFP (ILF)} = 8.6 \times \#\text{ILF}$ ;  $\text{UFP (total)} = \text{UFP (ILF)} \times 100 / 22$

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, 130 proyectos.

*Exactitud de la estimación:* 10% para el 50% de los proyectos evaluados, el restante 50% de los proyectos, entre 20% y 50%.

### 3.3.1.6. Early Function Point Method (EFPM)

Este método [Asensio Monge *et al.* 2004] permite estimar los FP conociendo alguno/s de los componentes (grupos de datos lógicos creados y mantenidos por la aplicación,

<sup>30</sup> ISBSG publica cada año una colección de métricas, la actual es *Benchmark Release 10*, pero el autor de esta tesis no ha tenido acceso a los datos.

grupos de datos lógicos procesados por la aplicación, número de procesos) y aplicando un modelo de regresión diferente según el componente. Los autores proponen distintas funciones de regresión establecidas a partir de la fuerte correlación observada entre los FP y los mencionados componentes considerando los datos de la medición de una muestra de 30 proyectos (todos pertenecientes al dominio MIS) usando el método IFPUG 4.1.

Para aplicarlo se requiere el documento de requerimientos del usuario o alguna documentación análoga. Sus autores manifiestan la necesidad de avanzar en la investigación para determinar la factibilidad de identificar esos componentes desde la documentación previa a la especificación funcional. No se encontró información referida a la validación de la métrica y por ende, sobre su exactitud.

### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* análisis de requerimientos.

*Artefacto de software:* especificación de requerimientos o documentación análoga.

*Reglas:* IFPUG.

*Función de medición:*  $FP = 130,327 + 15,902 * CILF$ , siendo CILF: #ILFs

$FP = 66,905 + 13,035 * CILFEIF$ , siendo CEIEOEQ: #EIs + #EOs + #EQs

$FP = 50,784 + 6,289 * CEIEOEQ$ , siendo CILFEIF: #ILFs + #EIFs

*Herramienta de soporte:* no.

*Validación del enfoque:* no especificado.

*Exactitud de la estimación:* no especificado.

### **3.3.2. Estimación de la complejidad media**

Estas técnicas se caracterizan por asignar un valor de complejidad promedio (o más probable) a cada componente del FPA. Estos valores de complejidad son determinados a partir del análisis estadístico de los datos de proyectos propios o de repositorios públicos. Parten del supuesto que todos los componentes del FPA estándar son identificables desde la documentación disponible. La simplificación del proceso de estimación es significativa, pues basta con identificar los componentes y asignarles el valor promedio de complejidad sin necesidad de investigar su estructura interna [Meli y Santillo 1999] (por ejemplo, para los ILFs, en vez de identificar y contabilizar los DETs y RETs, determinar la complejidad desde tablas y luego buscar el valor asociado a esa complejidad, directamente se les asigna un valor fijo a todos por igual).

#### **3.3.2.1. Propuesta de ISBSG <sup>(2)</sup>**

Esta propuesta dispone de valores de complejidad promedio para cada componente (ILF, EIF, EI, EO y EQ) que fueron determinados a partir del análisis estadístico de los datos de proyectos del repositorio de ISBSG. Por ejemplo, ISBSG Benchmark Release 5<sup>31</sup> establece: 4.3 FP para las EIs, 5.4 FP para las EOs, 3.8 FP para las EQs, 7.4 FP para los ILFs y 5.5 FP para los EIFs.

---

<sup>31</sup> Versiones más recientes podrían presentar valores ligeramente distintos, el autor de esta tesis no ha tenido acceso a las mismas.

Algunos autores [Bundschuh y Dekkers 2008] sugieren que es mejor usar los valores de complejidad promedio determinados a partir de los datos propios de cada organización. Por ejemplo, Ebert *et al.* (2005) estimaron los FP usando los valores típicos de complejidad (para su organización) y los compararon con la medición de FP, el error fue menor que el 13%.

### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* no especificado.

*Artefacto de software:* no especificado.

*Reglas:* IFPUG.

*Función de medición:* Est.UFP = #EI × 4.3 + #EO × 5.4 + #EQ × 3.8 + #ILF × 7.4 + #EIF × 5.5 (ISBSG Benchmark Release 5).

*Herramienta de soporte:* no.

*Validación del enfoque:* no se encontró información.

*Exactitud de la estimación:* no se encontró información.

#### **3.3.2.2. NESMA Estimated FP**

Consiste en identificar las funciones de todos los tipos (ILF, EIF, EI, EO, EQ), valorar la complejidad de cada función de datos (ILF, EIF) como *baja* y de cada función transaccional (EI, EO, EQ) como *media* (usando los valores de IFPUG) y por último calcular los UFP. La única diferencia con la medición estándar es que la complejidad no es determinada individualmente para cada función sino que es establecida por defecto.

Al igual que la otra propuesta de estimación temprana de NESMA (sección 3.3.1.4), la exactitud de *Estimated FP* fue determinada por comparación con la medición *Detailed FP* usando la misma base de datos de aproximadamente 100 aplicaciones. Los resultados son más exactos que los producidos por *Indicative FP* [Meli y Santillo 1999] y muy aproximados a los de la medición *Detailed FP* (en ninguno de los casos es especificada la magnitud).

### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* previa a la definición de los requerimientos.

*Artefacto de software:* no especificado.

*Reglas:* NESMA (ver sección 3.3.1.4).

*Función de medición:* ídem IFPUG, con valores de complejidad baja para las funciones de datos y media para las funciones transaccionales.

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, 100 proyectos.

*Exactitud de la estimación:* no especificado.

### 3.3.2.3. Simplified Method

Cândido y Sanches (2004) desarrollaron una variante de IFPUG destinada a la estimación de aplicaciones Web. El proceso es simplificado mediante la asignación de complejidad *baja* a todas las funciones (de datos y transaccionales), una vez que han sido identificadas aplicando las reglas de IFPUG. Un estudio empírico basado en 20 aplicaciones Web desarrolladas por una compañía demostró que los resultados son muy aproximados a los producidos por el método IFPUG. El error promedio fue del 4%, con un mínimo de 1% y un máximo de 10%. La propuesta incluye una herramienta que soporta la medición de FP, aunque no es especificado el alcance de dicho soporte. Los autores señalan que los resultados presentados sólo son válidos para el dominio del problema, los lenguajes de programación y los sistemas usados por la compañía en estudio.

#### **Evaluación**

*Dominio funcional:* aplicaciones Web.

*Fase:* análisis de requerimientos.

*Artefacto de software:* especificación de requerimientos.

*Reglas:* IFPUG.

*Función de medición:* IFPUG, usando valores de complejidad *baja* (IFPUG) para todas las funciones.

*Herramienta de soporte:* sí, no especificado si es automática o semiautomática.

*Validación del enfoque:* sí, 20 aplicaciones reales.

*Exactitud de la estimación:* entre  $\pm 1\%$  y  $\pm 10\%$ .

### 3.3.2.4. FAST Function Points

En este enfoque [Seaver 2000] se identifican todos los tipos de función de IFPUG desde los requerimientos y en base a la experiencia de los propios proyectos se determina un perfil de transacciones y archivos lógicos, así como de su complejidad típica. También pueden ser usados los datos disponibles en repositorios como el de ISBSG. Es muy similar a la propuesta de ISBSG<sup>(2)</sup> (sección 3.3.2.1). Si bien no es especificada la estrategia de validación, la técnica fue y es aplicada en diversas organizaciones sobre numerosos proyectos. Las estimaciones están dentro del 5% del “valor real” [Morrison y Seaver 2007] (no se indica si el “valor real” es la medición con el estándar IFPUG).

#### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* análisis de requerimientos.

*Artefacto de software:* especificación de requerimientos.

*Reglas:* IFPUG.

*Función de medición:* IFPUG, usando valores de complejidad promedio (propios o de ISBSG) para todas las funciones.

*Herramienta de soporte:* no.

*Validación del enfoque:* no especificado.

*Exactitud de la estimación: ±5%.*

### 3.3.2.5. Function Points Simplified (FPS)

Este enfoque fue desarrollado por Bock y Klepper<sup>32</sup> en McDonnell Douglas Corporation con el objetivo de agilizar el proceso de medición de los FP sin reducir significativamente la exactitud. En esta técnica de aproximación del tamaño funcional la contribución en FP de cada tipo de función (IFPUG) es valorada a través de un único factor (predeterminado para el tipo de función) en lugar de los tres del FPA [Desharnais y Abran 2003]. Para aplicarla se puede usar el valor de complejidad promedio de cada tipo de función obtenido desde las tablas de IFPUG o determinarlo a partir de una muestra de proyectos de una organización específica. Desharnais y Abran (2003) realizaron un estudio empírico donde, en primer lugar determinaron los factores de complejidad de cada tipo de función a partir de una muestra de 47 proyectos; luego, usando esos valores de complejidad, aplicaron FPS sobre 43 proyectos y compararon las estimaciones con la medición basada en el estándar IFPUG realizada por un especialista certificado. Los resultados demostraron que la técnica simplificada puede ser usada, en un contexto específico, en lugar del método de medición detallado, con una exactitud dentro del 5%.

#### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* análisis de requerimientos.

*Artefacto de software:* especificación de requerimientos.

*Reglas:* IFPUG.

*Función de medición:* IFPUG, usando valores de complejidad promedio (de IFPUG o propios) para todas las funciones.

*Herramienta de soporte:* no.

*Validación del enfoque:* sí, 43 proyectos reales.

*Exactitud de la estimación:* 5%.

### 3.3.3. Early & Quick Function Points (E&QFP)

Esta técnica [Iorio *et al.* 2007], [DPO 2008] combina diferentes enfoques de estimación para proveer una mejor estimación del tamaño funcional de un sistema de software. E&Q FP v3.0 es la última versión de la técnica y cumple con las definiciones de los conceptos y la estructura de un método de medición de tamaño funcional como son definidos por ISO 14143 [ISO 1998]. Desde este punto de vista, E&Q constituye una técnica de estimación general que puede ser extendida y aplicada a cualquier método que cumpla con dicho estándar ISO, por ejemplo, el método IFPUG FPA [Santillo *et al.* 2005]. Además, puede ser aplicada a lo largo de todo el ciclo de vida del software, a partir de la fase de estudio de factibilidad [DPO 2008].

La técnica permite una estimación rápida y temprana del tamaño funcional y con menor esfuerzo que la medición estándar. Una ventaja de este enfoque es que las estimaciones pueden estar basadas en los diferentes y no homogéneos niveles de detalle en la SRS disponible, permitiendo estimar a nivel de proceso funcional o a un nivel más

---

<sup>32</sup> El autor de esta tesis no tuvo acceso al artículo de Bock y Klepper.

alto de la clasificación según el nivel de conocimiento. Esta propiedad se conoce como estimación multinivel.

Los requerimientos funcionales del usuario, identificados en el artefacto de software, son asociados con los elementos básicos de E&Q - grupos de datos lógicos y procesos funcionales elementales - (esto es, los BFC del método IFPUG). Además, provee diferentes tipos de agregaciones: los BFC de tipo dato pueden ser agrupados en grupos de datos generales; los BFC de tipo transacción pueden ser agrupados en procesos lógicos “generales” y los procesos generales pueden ser agrupados en “macro” procesos lógicos.

A cada objeto lógico del software se le asigna un valor en UFP (*mínimo, más probable, máximo*) de acuerdo a una tabla analítico-estadística y luego esos valores son sumados para obtener el resultado (*mínimo, más probable, máximo*) de la estimación. La tabla de asignación está organizada en base a cuatro niveles de detalle de los requerimientos, los que corresponden a cuatro niveles de agregación de los componentes funcionales. La confiabilidad del método es directamente proporcional a la destreza del estimador para identificar y clasificar los elementos del sistema.

Las versiones previas del método fueron validadas usando más de 50 casos. Para la versión actual, un estudio comparativo con el estándar IFPUG sobre 12 sistemas reales demostró la consistencia de los resultados cuando es aplicado en condiciones similares a IFPUG. Usualmente la diferencia entre el valor estimado y el valor real es menor de  $\pm 10\%$  [Santillo *et al.* 2005] o  $< 15\%$  [DPO 2007]. Para la fase de factibilidad puede variar entre el  $\pm 20\%$  y el  $\pm 40\%$  [DPO 2008], aunque no es especificado el tipo de validación.

### **Evaluación**

*Dominio funcional:* MIS.

*Fase:* estudio de factibilidad.

*Artefacto de software:* documento de requerimientos del usuario.

*Reglas:* IFPUG y adicionales.

*Función de medición:* depende del nivel.

*Herramienta de soporte:* sí.

*Validación del enfoque:* para fase de factibilidad no se encontró información; 12 sistemas reales en condiciones similares a IFPUG.

*Exactitud de la estimación:* fase de factibilidad entre  $\pm 20\%$  y  $\pm 40\%$  [DPO 2008].

#### **3.3.4. Resumen sobre la evaluación de los Métodos de estimación**

En los siguientes párrafos se sintetizan los resultados de la evaluación de los métodos de estimación de acuerdo al esquema propuesto. La Tabla 3.3 (al final del capítulo) presenta el resumen de la evaluación de cada método de estimación.

- Dominio funcional: sólo una de las propuestas evaluadas no especifica el dominio, la mayoría (83%) está orientada al software MIS, lo cual es razonable debido a su relación con IFPUG.
- Fase más temprana del proceso de desarrollo de software en que puede ser aplicada: la mayoría de las propuestas son aplicables en las etapas tempranas de un proyecto, aunque en algunos casos (42%) no se especifica claramente a qué se considera temprano. Otras propuestas son aplicables en la fase de estudio de factibilidad (25%) y las restantes en la fase de análisis de requerimientos.

- Artefacto de software: algunas de las técnicas no especifican el tipo de documento necesario para hacer la estimación, otras requieren, explícita o implícitamente, un modelo de datos y un número significativo de enfoques (42%) usan la SRS, aunque el nivel de detalle requerido puede variar según la propuesta.
- Reglas propuestas: la mayoría de los enfoques usan las reglas de IFPUG (o similares, como es el caso de NESMA) para identificar los componentes funcionales (funciones de datos y/o transacciones). La técnica E&QFP incorpora reglas adicionales a las propias de IFPUG.
- Función de medición: cuatro enfoques producen la estimación de FP a partir del número de archivos lógicos, dos de ellos desde los ILFs, otro desde los ILFs y EIFs y el cuarto desde el total de archivos, sin diferenciar entre archivos internos y externos. Cinco de las propuestas estiman los FP considerando la contribución de todas las funciones de IFPUG, sin evaluar individualmente su complejidad. Una de las técnicas se basa en las transacciones, específicamente en las EIs y EOs. El enfoque EFPM propone varias fórmulas, algunas usan los archivos lógicos y otras las transacciones. Por último, la estimación de FP en E&Q FP puede incluir la contribución de diferentes ítems dependiendo del nivel.
- Herramienta de soporte: sólo dos propuestas incluyen un software que soporta el proceso, en un caso no se especifica si es automática o semiautomática.
- Validación del enfoque: en el 50% de las propuestas se describe la evaluación realizada, principalmente comparando el resultado de la estimación con la medición de proyectos reales. Puede observarse que las propuestas que fueron evaluadas usando un gran número de proyectos (100, 130) pertenecen a muy grandes organizaciones, como NESMA e ISBSG, mientras que los demás enfoques presentan entre 20 y 43 casos. El 50% restante no especifica si fue realizada una validación, no se encontró información o fueron usados los mismos datos para definir el modelo y validarlo.
- Exactitud de la estimación: en el 33% de las propuestas originales no es especificada la exactitud o la información no está disponible (un caso). En las restantes, la exactitud varía entre  $\pm 1\%$  y  $\pm 50\%$ , destacándose cuatro cuya exactitud es menor o igual a  $\pm 10\%$ , pero en tres de éstas se requiere la SRS. Esto significa que la exactitud está estrechamente relacionada con el tipo de documento utilizado, las que usan la SRS producen las mejores aproximaciones, pero también retrasan el momento de la estimación.

### 3.4. Panorama actual

A mediados de la década de 1990 se propuso la estandarización internacional de los métodos de FPA. Las autoridades de ISO y sus expertos en FSM [Abran *et al.* 2004]:

- acordaron acerca de los beneficios esperados de la FSM,
- no acordaron en que el FPA fuera la única solución,
- establecieron los criterios para controlar la calidad de los métodos de FSM.

A través de esas acciones se facilitó:

- el progresivo reconocimiento como estándares ISO de los métodos de FSM con suficientes fortalezas documentadas (MkII FPA, IFPUG, NESMA).
- la aparición y el reconocimiento oficial de un método de FSM de segunda generación (COSMIC-FFP).

- la integración de la FSM dentro de la infraestructura estándar de la Ingeniería de Software.

En la actualidad existen una variedad de propuestas de estimación/medición del tamaño funcional, pero sólo cinco métodos de medición están reconocidos como estándares internacionales para la FSM por el Comité ISO/IEC JTC1 SC7.

### 3.5. Conclusiones

En este capítulo fue presentada una revisión de los métodos de medición y estimación de tamaño funcional. La evaluación fue realizada considerando el dominio funcional y la fase más temprana del proceso de desarrollo en que pueden ser aplicados, el artefacto de software que es usado como entrada al proceso de medición/estimación, la disponibilidad de una herramienta de soporte para agilizar el proceso, las reglas propuestas para identificar, clasificar y cuantificar los ítems funcionales del artefacto de software, la función de medición usada para calcular el tamaño funcional, la estrategia aplicada para la validación del enfoque y la exactitud de la medición/estimación con respecto a un valor de referencia predeterminado. A continuación se presentan las principales conclusiones.

En lo referido al dominio funcional, todas las propuestas de medición y la mayoría de las de estimación están orientadas al software MIS, lo cual es razonable debido a que de un modo u otro (extensión de las reglas, modelos estadísticos) han sido derivadas del método IFPUG.

En cuanto a la fase en que puede ser aplicado el enfoque, la mayoría de las propuestas de medición y algunas de las de estimación son aplicables en la fase de análisis de requerimientos. En las restantes propuestas de estimación, en algunas se indica que son aplicables desde las etapas tempranas pero no se establece específicamente en que punto del proceso y un número menor puede ser usada durante el estudio de factibilidad. Esto permite concluir que, con excepción del enfoque de Choi *et al.*, no se encontraron otras propuestas para medir el tamaño funcional en la fase de Elicitación de Requerimientos.

La mayoría de los enfoques de medición se basa en la SRS, la que puede variar en su forma de representación (SRS tradicional, modelo en notación UML), con excepción del enfoque de Choi *et al.* que trabaja con documentos en lenguaje natural generados con la técnica de objetivos y escenarios. En cuanto a las técnicas de estimación, varias requieren el modelo de datos del sistema y aún la SRS. Dos conclusiones pueden ser extraídas de este análisis: por un lado, en general se requieren documentos con un nivel de detalle que excede largamente la escasa información disponible al comienzo de un proyecto y en algunos casos no difiere demasiado de lo requerido para la medición estándar; por otro lado, no se encontraron enfoques basados en escenarios, excepto la propuesta de Choi.

En cuanto a las herramientas de soporte para el proceso de medición/estimación, la conclusión es que son escasos los enfoques que disponen de un software automático o semiautomático. Esto evidencia la dificultad que implica desarrollar este tipo de software, debido a que se requiere procesar texto en lenguaje natural - posiblemente no estructurado -, extraer los ítems que aportan tamaño funcional al sistema, clasificarlos en una categoría y cuantificarlos, procesos que deben ser realizados de acuerdo al modelo y las reglas del método de medición/estimación.

La validación en general ha sido realizada aplicando el enfoque sobre casos de estudio o sistemas reales y comparando con el resultado de la medición realizada con el método IFPUG, aunque hay métodos que en su propuesta original no fueron validados. En lo que refiere específicamente a las propuestas de medición, en general el número de casos usados para tal verificación es muy reducido, entre 1 y 3. En cuanto a la validación de los



enfoques de estimación, varias propuestas no especifican si fue realizada alguna comprobación de los resultados mientras que en las restantes se describe la evaluación realizada y el número de casos usados, que es significativamente alto (100, 130) para las que pertenecen a grandes organizaciones mientras que en las restantes varía entre 20 y 43 proyectos. Se puede concluir que la validación de los métodos de medición en general fue realizada sobre muy pocos casos, mientras que las técnicas de estimación usaron más casos que los métodos de medición o directamente no fueron validadas.

La exactitud de la medición puede variar entre 1% y 14%, mientras que para las estimaciones el rango puede ampliarse hasta el 50% para la fase de factibilidad. Sin embargo, es destacable que aproximadamente la mitad de los enfoques evaluados no presentan información al respecto y esto está estrechamente relacionado con la observación del párrafo anterior sobre la falta de validación.

Una última conclusión, parece existir una gran preocupación y también una falta de acuerdos mínimos acerca de qué, cómo y cuándo medir el software. Esto queda evidenciado en la gran cantidad y variedad de enfoques de FSM existentes. Ante este panorama de múltiples soluciones para la medición temprana del software, es importante resaltar que, sea cual fuere la técnica elegida, siempre es mejor una estimación inexacta (que podrá ser refinada posteriormente) que ninguna estimación o el retraso de la misma hasta disponer de más información.

Tabla 3.2. Resumen de los enfoques de medición analizados

Enfoque	Dominio	Fase	Artefacto	Reglas	Función de medición	Herramienta	Validación	Exactitud
IFPUG	MIS	análisis de requerimientos	SRS	identificación, clasificación y cuantificación de los 5 componentes	$UFP = \sum_{\#i,j \in 1,3} \#ElementTypes_{ij} W_{ij}$	no	no en la propuesta original, si en la práctica	-
Fetcke <i>et al.</i>	MIS	análisis de requerimientos	modelo de UC y de objetos generados con OOSE	IFPUG adaptadas	IFPUG	no	no	no especificado
Harpur <i>et al.</i>	MIS	análisis de requerimientos	SRS OO	IFPUG adaptadas	IFPUG	sí, semiautomática	1 caso de estudio y 1 caso real	10% - 14%
Tavares <i>et al.</i>	MIS	análisis de requerimientos	SRS OO	IFPUG adaptadas	IFPUG	no	1 proyecto real	no especificado
OOMFP	MIS	análisis de requerimientos	esquema conceptual OO-Method	IFPUG adaptadas	IFPUG adaptada	si	1 caso	1% - 13%
Kusumoto <i>et al.</i>	MIS	análisis de requerimientos	SRS OO generada por REQUARIO	IFPUG adaptadas	IFPUG	si	3 sistemas reales	no especificado
Choi <i>et al.</i>	MIS	elicitación de requerimientos	texto de requerimientos basado en objetivos y escenarios	IFPUG adaptadas	IFPUG	no	no especificado	no especificado
Use Case Points	MIS	análisis de requerimientos	modelo de UC	clasificación y cuantificación de actores y UC	$UUCP = \sum_{i=1}^6 n_i * W_i$	si	no en enfoque de Karner; 23 casos, Arnold y Pedross	±10% (Arnold y Pedross)

Tabla 3.3. Resumen de los enfoques de estimación analizados

Enfoque	Dominio	Fase	Artefacto	Reglas	Función de medición	Herramienta	Validación	Exactitud
Modelo basado en ILF	no especificado	no especificado	modelo de datos	IFPUG	$UFP = \#ILF * \text{factor}$ ; factor: 30 - 35 para aprox.	no	sí, no se encontró información del número de sistemas	10%
Rule of the "Thirties"	MIS	no especificado	modelo de datos	-	$FP = \#archivos \text{ lógicos} *$ $FP/archivo \text{ lógico (entre 31 y 35)}$	no	no especificado	$\pm 30\%$
FP Prognosis	MIS	estudio de factibilidad	no especificado	IFPUG	$FP = 7.3 \times \#IO + 56$ ; $\#IO = \#EI + \#EO$	no	no	15% - 20%
NESMA Indicative FP	MIS	no especificado	modelo de datos	NESMA	$UFP = 35 \times \#ILF + 15 \times \#EIF$	no	100 proyectos	no especificado; 15% [Jones 2008b]
Propuesta de ISBSG <sup>(1)</sup>	MIS	estudio de factibilidad	modelo de datos	IFPUG	$UFP (ILF) = 8.6 \times \#ILF$ ; $UFP (total) = UFP (ILF) \times 100 / 22$	no	130 proyectos	10% - 50%
EFPM	MIS	análisis de requerimientos	SRS o documentación análoga	IFPUG	$FP = 130,327 + 15,902 * CILF$ ; $CILF: \#ILFs$ $FP = 66,905 + 13,035 * CILFEIF$ ; $CEIEOEQ: \#EIs + \#EOs + \#EQs$ $FP = 50,784 + 6,289 * CEIEOEQ$ ; $CILFEIF: \#ILFs + \#EIFs$	no	no especificado	no especificado
Propuesta de ISBSG <sup>(2)</sup>	MIS	no especificado	no especificado	IFPUG	$Est. UFP = \#EI \times 4.3 + \#EO \times 5.4 + \#EQ \times 3.8 + \#ILF \times 7.4 + \#EIF \times 5.5$	no	no se encontró información	no se encontró información

Capítulo 3. Estado del arte

Enfoque	Dominio	Fase	Artefacto	Reglas	Función de medición	Herramienta	Validación	Exactitud
NESMA Estimated FP	MIS	no especificado	no especificado	NESMA	IFPUG, valores de complejidad baja para archivos lógicos y de complejidad media para transacciones	no	100 proyectos	no especificado
Simplified Method	Web	análisis requerimientos	SRS	IFPUG	IFPUG, valores de complejidad media (IFPUG) para todas las funciones	si, tipo no especificado	20 aplicaciones reales	$\pm 1\%$ - $\pm 10\%$ .
FAST Function Points	MIS	análisis requerimientos	SRS	IFPUG	IFPUG, valores de complejidad media (propios o ISBSG) para todas las funciones	no	no especificado	$\pm 5\%$
Function Points Simplified	MIS	análisis requerimientos	SRS	IFPUG	IFPUG, valores de complejidad media (IFPUG o propios) para todas las funciones	no	43 proyectos reales	5%
E&QFP	MIS	estudio factibilidad	Requerimientos del usuario o del sistema	IFPUG y adicionales	depende del nivel	si	no se encontró información	$\pm 20\%$ - $\pm 40\%$

## Capítulo 4

### SFP: Procedimiento de estimación de tamaño funcional de Escenarios

El procedimiento *Scenario Function Points* (SFP – Puntos Función de Escenarios) permite estimar el tamaño funcional de los escenarios [Leite *et al.* 2000] generados en la fase de Elicitación de Requerimientos. Esta propuesta está basada en el método IFPUG FPA y fue diseñada siguiendo el Modelo de Proceso para los Métodos de Medición del Software de Jacquet y Abran (1997) y el estándar ISO 14143-1 (1998). La ejecución de SFP es soportada por un proceso que, partiendo de los escenarios y aplicando reglas específicas a través de una secuencia de pasos, concluye con una estimación del tamaño funcional.

En este capítulo, en primer lugar, se presenta la descripción detallada del diseño del procedimiento SFP en base al Modelo propuesto por Jacquet y Abran (1997). Luego, se describe el proceso que guía la aplicación de SFP, la comparación con otros modelos de proceso y un ejemplo ilustrativo de la aplicación de SFP.

#### 4.1. Diseño del Procedimiento SFP

El paso 1 del Modelo de Proceso para los Métodos de Medición [Jacquet y Abran 1997], [Abran y Jacquet 1999] (Capítulo 2, sección 2.2.7.3) establece los pasos para el Diseño del Método de Medición (Figura 4.1). Esta guía fue usada para el diseño de SFP.

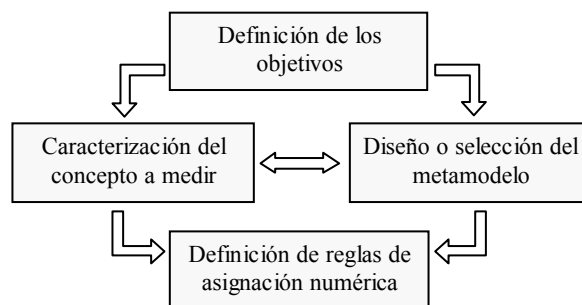


Figura 4.1. Diseño del método de medición (traducido de [Jacquet y Abran 1997])

En las siguientes secciones se presenta el detalle de cada una de estas actividades.

##### 4.1.1. Definición de los objetivos

Según Abran y Jacquet (1999), los objetivos de diseño de un método de medición deberían responder a un conjunto de preguntas, las que, asociadas con la normativa del estándar ISO 14143-1 [ISO 1998] permitieron definir los objetivos (Tabla 4.1).

Tabla 4.1. Definición de los objetivos del diseño del procedimiento de estimación

Pregunta...	El estándar ISO 14143-1...	Objetivo	Justificación
¿Qué se va a medir? (atributo, clase de software, etc.)	establece en relación a la clase de software: - un Método de Medición del Tamaño Funcional (FSMM) debe definir el dominio funcional al que puede ser aplicado.	- el atributo es el <i>tamaño funcional</i> .  - el <i>dominio funcional</i> es genérico.	Implícito en una propuesta de Medición del Tamaño Funcional (FSM).  En la bibliografía sobre el Léxico Extendido del Lenguaje (LEL) y escenarios [Leite <i>et al.</i> 1997] no se especifica ni restringe el dominio funcional de aplicación de la técnica).
¿Cuál es el punto de vista del método de medición? (usuario del software, diseñador del software, etc.)	establece: - un FSMM está basado en una representación de los requerimientos desde la perspectiva de los usuarios.	- el <i>punto de vista</i> es el del usuario.	Los escenarios “son usados para describir el comportamiento externo del sistema centrado en el punto de vista del usuario” [Rivero <i>et al.</i> 1998]. (*)
¿Cuáles son las propiedades y características del método de medición?	establece: - un FSMM puede ser aplicado tan pronto como se hayan definido los Requerimientos Funcionales del Usuario (FURS). - un FSMM deber ser tan independiente como sea posible de métodos de desarrollo de software o tecnologías particulares.	- el <i>artefacto de software</i> está formado por los escenarios del sistema de software, desarrollados según el enfoque de Leite.  - es <i>independiente</i> de un método o tecnología de desarrollo de software particular.	Ídem justificación (*)  El uso de escenarios evita “abstracciones orientadas a una solución” [Hada <i>et al.</i> 1999]).
¿Cuál es el uso previsto del método de medición?	no impone restricciones, pero requiere que sea especificado: - un FSMM debe describir los propósitos para los que puede ser usado, de modo que los usuarios del método puedan juzgar su conveniencia para su propósito.	- el <i>propósito</i> es usar el tamaño funcional como parámetro de entrada en los modelos de estimación de un proyecto de software.	El propósito de la medición determina el tipo y el alcance de la medición de Puntos Función (FP). Ambos influyen en la ubicación del límite del sistema a medir [IFPUG 2000].

**El Estándar ISO/IEC 14143-1 y los Escenarios.** Este estándar establece claramente que el tamaño funcional es “un tamaño del software derivado a partir de la cuantificación de los requerimientos funcionales del usuario”. Los escenarios no son especificaciones ni requerimientos, son descripciones auxiliares para el proceso de definición de requerimientos [Leite *et al.* 1997]. Esta limitación puede ser superada considerando los siguientes argumentos:

- las definiciones de ISO 14143-1 [ISO 1998] han sido establecidas en el contexto de la “medición” del tamaño funcional; el procedimiento propuesto en esta tesis tiene por objetivo “estimar” (no “medir”) el tamaño funcional de un proyecto (es decir, obtener una aproximación - perfectible *a posteriori* - del tamaño funcional).
- en dicho marco, como “los escenarios representan una fuente de conocimientos donde pueden ser identificados los requerimientos y en la que pueden basarse las especificaciones” [Leite *et al.* 1997], se considera que la información contenida en ellos puede ser usada como fuente para producir tal estimación.

#### 4.1.2. Caracterización del concepto a medir

Cuando fueron definidos los objetivos de diseño de SFP se estableció que el artefacto de entrada para la estimación está formado por los escenarios y que el atributo a estimar es el tamaño funcional. Para definir los subconceptos que componen el tamaño funcional de los escenarios fue necesario identificar los componentes de los escenarios relacionados a la funcionalidad del sistema y asociarlos con los del modelo de IFPUG.

A continuación se presenta el análisis de los conceptos del Modelo de Escenarios y las reglas definidas para asociarlos con los de IFPUG FPA, teniendo como marco de referencia al estándar ISO 14143-1 [ISO 1998].

##### 4.1.2.1. Clasificación de los objetos del LEL

Cada componente de los escenarios (Capítulo 2, sección 2.1.2) está formado por texto escrito en lenguaje natural que puede incluir *Símbolos del LEL* (Capítulo 2, sección 2.1.1) y palabras o frases que no se corresponden con una entrada del LEL pero que son significativas para el usuario (como fue explicado en el Capítulo 2, sección 2.1.1); a estas últimas – en el contexto de esta propuesta – se las denomina *No-Símbolos*. Considerando la noción de un *Símbolo*, es posible clasificarlo como *Simple* (su definición no involucra a otros símbolos) y *Complejo* (su definición se basa en otros *Símbolos* y/o *No-Símbolos*). La Tabla 4.2 presenta la clasificación propuesta.

Tabla 4.2. Clasificación de los objetos del LEL

<b>Objeto del LEL</b>	<b>Símbolo:</b> entrada del LEL	<i>Simple</i> : una pieza elemental de información reconocible por el usuario. Ejemplo <sup>33</sup> : <u>tarifa</u>
		<ul style="list-style-type: none"> <li>• Noción (*)             <ul style="list-style-type: none"> <li>– Es el <b>costo</b><sup>NS</sup> de la <u>habitación</u> según sus atributos.</li> </ul> </li> </ul>
		<i>Complejo</i> : un conjunto formado por uno o más <i>Símbolo Simple</i> y/o uno o más <i>No-Símbolo</i> . Ejemplo: <u>lista de precios</u> <ul style="list-style-type: none"> <li>• Noción             <ul style="list-style-type: none"> <li>– Se utiliza para almacenar la <b>tarifa</b><sup>S</sup> según el <b>tipo</b><sup>NS</sup> y <b>modalidad</b><sup>NS</sup> de la <u>habitación</u>.</li> </ul> </li> </ul>

<sup>33</sup> Los ejemplos corresponden al caso Recepción del Hotel [Bertolami y Centeno 2001], en caso contrario se indica la referencia.

	<b>No-Símbolo:</b> palabra o frase perteneciente al vocabulario mínimo. Ejemplo: número de documento.
--	--

(\*) extracto de la noción del símbolo del LEL donde se especifica la definición en términos de un Símbolo (S) y/o un No-Símbolo (NS).

#### 4.1.2.2. Punto de vista del usuario

El *punto de vista del usuario* representa una descripción formal de las necesidades de negocio del usuario en el lenguaje del usuario [IFPUG 2000]. Los componentes de software que no son visibles desde el punto de vista del usuario no son parte del modelo de medición de FP [Fetcke *et al.* 1997]. Esto es, en el FPA sólo son consideradas las funciones del software que son requeridas por el usuario.

Ahora bien, ¿qué componente de los escenarios cumple el rol de “usuario”? Esto es, se debe encontrar en el modelo de escenarios un concepto equivalente o similar al de usuario en el FPA e ISO 14143-1. Para evaluar esta cuestión fueron comparadas las definiciones de usuario y de actor.

#### Conceptos de Usuario y Actor

##### *Usuario*

Definición IFPUG: persona que especifica requerimientos funcionales y/o cualquier otra persona o cosa que se comunique o interactúe con el software [IFPUG 2000].

Definición ISO: persona o cosa que se comunica o interactúa con el software en cualquier momento durante el tiempo de vida del software [ISO 1998].

La definición de *usuario*<sup>34</sup> propuesta por Fetcke *et al.* (2001) es coincidente con la de ISO 14143-1. En el contexto de la medición de FP sobre Casos de Uso, los términos *actor* y *usuario* son considerados sinónimos [Longstreet 2001].

##### *Actor*

Definición: persona, dispositivo o estructura organizacional que cumple un rol en el escenario [Leite *et al.* 2000]. Realiza las acciones indicadas por los *verbos* sobre los *objetos* o sobre otros *sujetos*. Un actor se identifica a partir de su interacción con el sistema y por los eventos que envía hacia o recibe desde el sistema.

Las similitudes observadas entre los conceptos de usuario y actor - ambos involucran a personas o cosas comunicándose o interactuando con el software - permitieron establecer la siguiente regla:

#### Regla de asociación entre actor y usuario

**R1.** Aceptar cada *actor* como equivalente a un rol de *usuario* en IFPUG e ISO 14143.

#### 4.1.2.3. Límite del sistema

El *límite del sistema* define la interfaz conceptual entre la aplicación y el mundo externo del usuario [IFPUG 2000], [ISO 1998]. Es independiente de las consideraciones técnicas y/o de implementación y dependiente del punto de vista del usuario. Es atravesado en ambos sentidos (desde adentro hacia fuera e inversamente) por los datos procesados por las transacciones - Entrada Externa (EI – External Input), Salida Externa (EO – External Output) y Consulta Externa (EQ – External Inquiry) - y encierra los

---

<sup>34</sup> El usuario interactúa con una aplicación y no está restringido a usuarios humanos sino que puede incluir “usuarios” software y hardware.



archivos lógicos mantenidos por la aplicación (ILFs – Internal Logical File) [IFPUG 2000].

Debido a que los usuarios están representados por los actores y la aplicación por los escenarios, el límite del sistema es definido como la línea imaginaria que encierra a los escenarios y los separa del mundo externo.

**Regla para identificar el límite del sistema**

**R2.** Aceptar como *límite del sistema* al borde imaginario que encierra a todos los escenarios.

Dicho límite es atravesado en ambos sentidos por la información enviada o recibida por el actor durante su interacción con el sistema.

**4.1.2.4. Archivo lógico, BFC y recurso**

En IFPUG, las funciones de datos (también llamadas *archivos lógicos*) representan la funcionalidad proporcionada al usuario para satisfacer los requisitos de datos internos y externos. Un archivo lógico (o también llamados *datos en reposo* [Longstreet 2003]) es definido como un grupo de datos relacionados lógicamente o información de control<sup>35</sup>, identificable por el usuario<sup>36</sup> [IFPUG 2000].

En la terminología de ISO 14143-1 (1998), un Componente Funcional Básico (BFC – Base Functional Component) es una unidad elemental de FURs definida y usada por un método de FSM para propósitos de medición. En IFPUG, los BFCs están representados por los archivos lógicos y las transacciones, estas últimas serán tratadas en la sección 4.1.2.6.

En el Modelo de Escenarios, los *recursos* representan elementos físicos o información relevante que debe estar disponible en el escenario [Leite *et al.* 2000], son las entidades pasivas con las cuales los actores trabajan [Hadaad *et al.* 1999]. Los recursos reciben las *acciones* indicadas por los *verbos* y ejecutadas por los *sujetos*. Son utilizados durante la realización de un *episodio*.

Considerando la definición de recurso, pueden ser clasificados como objetos físicos y objetos lógicos. Los objetos lógicos pueden ser *Símbolos Complejos* o *No-Símbolos* (Tabla 4.3). A pesar que es recomendable que los recursos sean preferentemente símbolos del léxico [Leite *et al.* 2000], en la práctica algunos escenarios incluyen recursos *No-Símbolos*.

Tabla 4.3. Clasificación de los Recursos

<b>Recurso</b>	<i>Objeto físico</i> : entidad física necesaria para realizar el episodio. Ejemplo: teléfono, aparato de fax, etc.	
	<i>Objeto lógico</i> : entidad de soporte de información. Ejemplo: planilla, formulario, tabla, etc.	<i>Símbolo Complejo</i> Ejemplo: <u>planilla del pasajero</u>
<i>No-Símbolo</i> Ejemplo: lista de insumos faltantes		

A partir de la comparación del recurso de categoría objeto lógico con el concepto de archivo lógico se determinaron las siguientes características en común: el recurso es un

<sup>35</sup> Datos que influyen sobre un proceso elemental de la aplicación que se está midiendo. Especifica qué, cuándo o cómo van a ser procesados los datos [IFPUG 2000].

<sup>36</sup> Se refiere a requisitos definidos para procesos y/o grupos de datos que están acordados y entendidos por usuarios y desarrolladores de software [IFPUG 2000].

medio de soporte de información, pasivo e identificable por el usuario. Esto último es debido a que las descripciones del LEL están basadas en la información proporcionada por el usuario.

En base a estas consideraciones, fue establecida la siguiente regla:

### **Regla de asociación entre recurso y archivo lógico**

**R3.** Aceptar cada *recurso*<sup>37</sup> de categoría *objeto lógico* de tipo *Símbolo Complejo* o *No-Símbolo* como equivalente a un *archivo lógico* en IFPUG y un *BFC* de ISO 14143-1.

Nota: los recursos de tipo No-Símbolo son considerados, aunque se carezca de su definición, porque representan parte de la funcionalidad requerida por el usuario.

Como ya fue mencionado, los archivos lógicos son clasificados como ILFs o Archivos de Interfaz Externos (EIF – External Interface File). El propósito principal de un ILF es almacenar datos que son mantenidos a través de uno o más procesos elementales de la aplicación que se está midiendo. El propósito principal de un EIF es almacenar datos que son referenciados por uno o más procesos elementales de la aplicación que se está midiendo.

La información disponible en la etapa de construcción de los escenarios impide establecer con certeza cuáles serán los recursos mantenidos por la aplicación y cuáles sólo serán referenciados con el propósito de recuperar información. Por lo tanto, los recursos representan indistintamente datos externos e internos. Un razonamiento similar es aplicado por Meli (1997) en el contexto de Early Function Points, cuando sugiere que la diferencia entre ILF y EIF puede ser ignorada al identificar datos lógicos, dado que no siempre es clara en un nivel de análisis superficial; para distinguir entre datos externos e internos se requiere conocer que un determinado archivo estará sujeto a escritura por un proceso de la aplicación. Además, un archivo clasificado como externo debería cambiar a interno si en una etapa posterior se descubre una nueva función que altera sus datos, lo que haría que el resultado de la estimación fuera demasiado dependiente de fluctuaciones mínimas en el detalle de la información sobre el proyecto [Meli y Santillo 1998].

#### **4.1.2.5. Tipo de Dato Elemental y objeto del LEL**

El Manual IFPUG (2000) define al Tipo de Dato Elemental (DET - Data Element Type) como un campo único, no repetido, reconocible por el usuario. Por ejemplo, un número de cuenta bancaria es un DET. La condición de no repetido está relacionada a la contabilización de los DETs y significa que aunque aparezca en múltiples campos debe ser contado una sola vez. Esta cuestión será abordada posteriormente en este capítulo.

El documento de ISO 14143-1 (1998) no especifica ningún requerimiento para la identificación de entidades, así como tampoco suministra especificaciones respecto a los subconceptos que comprenden el concepto de funcionalidad. Es responsabilidad del diseñador del método proveer las reglas que permitan identificar los componentes sin ambigüedades.

De acuerdo a la clasificación de los objetos del LEL (sección 4.1.2.1), un símbolo Simple o un No-Símbolo (no identificado como recurso) representa una unidad de información que es reconocible por el usuario. La similitud de las definiciones del Símbolo Simple y No-Símbolo (no recurso) con la definición de DET en IFPUG soporta la definición de la siguiente regla:

---

<sup>37</sup> De aquí en adelante el término *recurso* se refiere exclusivamente a *objetos lógicos*.

#### Regla de asociación entre Objetos del LEL y Tipo de Dato Elemental

**R4.** Aceptar cada *Símbolo Simple* y cada *No-Símbolo* (no recurso) como equivalente a un *Tipo de Dato Elemental (DET)* en IFPUG.

Nota: esta regla incluye también al *No-Símbolo* que no es recurso y que es mencionado por el usuario como información necesaria para completar una determinada operación. Por ejemplo, el usuario requiere que sea ingresado el número de documento de la persona que solicita una reserva en el hotel.

Ejemplos

- Símbolo Simple: *tarifa*
- No-Símbolo: número de documento

#### 4.1.2.6. Tipo de Registro Elemental

En IFPUG, un Tipo de Registro Elemental (RET – Record Element Type) es definido como un subgrupo de datos elementales, reconocible por el usuario, dentro de un ILF o EIF. La definición de este concepto es similar en el marco de SFP (se dice similar y no igual porque debe ser reemplazado “ILF o EIF” por “recurso” debido a la asociación de los conceptos de archivo lógico y recurso).

#### 4.1.2.7. Proceso elemental, BFC y episodio simple

En una propuesta previa [Bertolami 2003] fue establecida la asociación conceptual entre el *episodio simple* y la *transacción lógica* del Método MkII FPA [Symons 1991]. De modo similar, en este nuevo enfoque se propone asociar el concepto de *episodio simple* con el de proceso elemental en IFPUG y de BFC de ISO 14143-1 (Capítulo 2, sección 2.2.4).

Como fue mencionado (Capítulo 2, sección 2.2.3), un proceso elemental es definido como la unidad de actividad más pequeña que es significativa para el usuario. Debe ser autosuficiente y dejar la aplicación en estado consistente. De acuerdo a su propósito principal es clasificado como EI (mantener un ILF o alterar el comportamiento del sistema), EO (presentar información al usuario; la lógica del proceso debe contener una fórmula matemática o cálculo o crear datos derivados; también puede mantener uno o más ILFs y/o modificar el comportamiento del sistema) o EQ (presentar información al usuario mediante la recuperación de datos o información de control; la lógica del proceso no contiene fórmulas matemáticas ni cálculos, no crea datos derivados y no se mantiene ningún ILF ni se modifica el comportamiento del sistema) [IFPUG 2000].

En el Modelo de Escenarios [Leite *et al.* 2000], cada escenario incluye un conjunto de episodios, que es definido como una serie de acciones que detallan al escenario y describen su comportamiento. Cada episodio representa una *acción* realizada por un *actor*, donde pueden participar otros *actores* y se utilizan *recursos* [Hadad *et al.* 1999]. Los episodios son de tipo *simple*, *condicional* y *opcional*. Independientemente de su tipo, un episodio puede ser expresado como una única acción o un escenario. El modelo de escenarios provee la descripción de comportamientos con diferentes órdenes temporales. Una secuencia de episodios implica un orden de precedencia, pero un orden no secuencial requiere el agrupamiento de dos o más episodios. Esto se usa para expresar un orden paralelo o secuencial indistinto [Leite *et al.* 2000].

En el contexto de esta tesis, se considera al episodio simple (no escenario) como la categoría básica, pues tanto los episodios condicionales como los opcionales, ambos encierran un episodio simple, como se puede observar en las siguientes descripciones en

notación BNF extraídas de la descripción de los escenarios [Leite *et al.* 2000] (en Capítulo 2, sección 2.1.2 se incluye la descripción completa):

<simple sentence> ::= <episode sentence> CR

<conditional sentence> ::= IF <condition> THEN <episode sentence> CR

<optional sentence> ::= [ <episode sentence> ] CR

Los episodios pertenecientes a un grupo no secuencial pueden incluir episodios de cualquiera de los tres tipos y puede ser aplicado el mismo análisis.

Un *episodio simple* [Leite *et al.* 2000] (no escenario) es definido como una *sentencia simple* basada en la siguiente notación:

[Actor | Resource] + Verb + Predicate (Capítulo 2, sección 2.1.2, Figura 2.2)

Como un episodio expresa una única acción y fue derivado desde las descripciones del LEL, se puede considerar que es la unidad de actividad más pequeña reconocible por (o significativa para) el usuario. En este sentido, existen similitudes entre el proceso elemental y el episodio simple.

Resta analizar otro requisito del proceso elemental: un proceso elemental debe ser autosuficiente y dejar la aplicación en estado consistente. Para aclarar estos aspectos, a continuación se presenta un ejemplo extraído del manual de IFPUG (2000).

*Un usuario requiere la capacidad de añadir un nuevo empleado a una aplicación y la definición del empleado, realizada por el usuario, incluye el salario e información de beneficiarios. Desde la perspectiva del usuario, la unidad más pequeña de actividad es añadir un nuevo empleado. Añadir una parte de la información, como el salario o los beneficiarios, no es una actividad que se pueda calificar como proceso elemental. Si se añaden a la vez el salario del empleado y la información de beneficiarios, esta unidad de actividad está completa y el proceso empresarial queda en un estado consistente.*

Como los escenarios no son una Especificación de Requerimientos (SRS – Software Requirement Specification), las descripciones carecen del nivel de detalle de tales documentos. En algunos casos la información disponible podría no ser suficiente para establecer en forma rigurosa si un episodio cumple ambas condiciones. Sin embargo, se puede lograr una aproximación suficiente, de acuerdo al objetivo de esta propuesta, en base a la información del LEL y los escenarios.

A los efectos de comparar con el ejemplo sobre el alta de un empleado del manual de IFPUG y aclarar la anterior afirmación, a continuación se presenta un episodio y un fragmento de un símbolo del LEL, extraídos del LEL y Escenarios (L&E) del Sistema de Gestión de Comercio [Bertolami y Centeno 2008].

El siguiente episodio se refiere al alta de un empleado:

- El *administrador* ingresa los datos de un nuevo *empleado* en el formulario de *empleado*.

En la noción del símbolo *empleado* se establece, entre otras descripciones, cuáles son los datos del empleado de interés para el usuario.

- Sus datos son: cuil, tipo documento, nro. documento, apellido, nombre, dirección, porcentaje de comisión, tipo de usuario (*empleado*, *administrador*), tipo teléfono, teléfono, e-mail, nro. de vendedor.

Al ejecutarse el episodio serán ingresados los datos especificados en la noción de símbolo empleado en el respectivo formulario. Al finalizar su ejecución el proceso estará completo y la aplicación queda en un estado consistente.

El análisis y consideraciones en relación al episodio simple comparado con el proceso elemental permitieron establecer la siguiente regla:

**Regla de asociación entre episodio simple y proceso elemental**

**R5.** Aceptar cada *episodio simple* (no escenario) como equivalente a un *proceso elemental* de IFPUG y a un *BFC* de ISO 14143-1.

El escaso nivel de detalle disponible en los escenarios dificulta la clasificación de los episodios como lo prescribe IFPUG, especialmente para distinguir entre las EOs y EQs. Por lo tanto, en el marco de SFP los episodios simples son clasificados como *Entradas Externas (EI)* y *Salidas Externas (EO)* de acuerdo a su propósito. De modo similar a IFPUG, son definidas como sigue:

EI: episodio simple que procesa datos o información de control que vienen desde fuera del límite de la aplicación.

EO: episodio simple que envía datos o información de control fuera del límite de la aplicación.

Las EIs tienen semántica similar a las EIs de IFPUG, pero mientras estas últimas mantienen un ILF, las EIs de SFP mantienen un recurso. Las EOs comprenden la presentación de datos al usuario, independientemente de la naturaleza de los mismos (es decir, éstos pueden ser recuperados desde un recurso o generados por un proceso), representan una combinación de las EOs y EQs de IFPUG y no deben ser confundidas con sus EOs.

**Propósito principal de un episodio simple**

EI: mantener un *recurso*.

EO: presentar información al usuario. Pueden ser datos recuperados de 1 o más *recursos* o datos derivados.

Nota: adaptado de [IFPUG 2000].

Por último, en IFPUG un Tipo de Archivo Referenciado (FTR – File Type Referenced) es un ILF leído o mantenido por una función transaccional o un EIF leído por una función transaccional. En el contexto de SFP, un Tipo de Recurso Referenciado (RTR – Resource Type Referenced) es un recurso mantenido o leído durante la realización de un episodio simple<sup>38</sup>.

**4.1.2.8. Resumen de la asociación de conceptos**

Además de establecer las asociaciones conceptuales, cada concepto fue comparado con las definiciones del estándar ISO/IEC 14143-1 (Tabla 4.4).

---

<sup>38</sup> De aquí en adelante se usará *episodio* con el mismo significado que *episodio simple*.

Tabla 4.4. Asociación de conceptos entre SFP, FPUG e ISO/IEC 14143-1

ISO 14143-1	FPUG	SFP
<p><i>Usuario:</i> cualquier persona o cosa que se comunica o interactúa con el software en cualquier momento durante el tiempo de vida del software.</p> <p><i>Alcance:</i> determina los BFCs a ser incluidos en una aplicación específica de un método de FSM.</p> <p><i>Límite:</i> interfaz conceptual entre el software bajo estudio y sus usuarios.</p> <p><i>BFC:</i> unidad elemental de Requerimientos Funcionales del usuario definido y usado por un método de FSM para propósitos de medición. Sólo expresa Requerimientos Funcionales del Usuario, no expresa Requerimientos Técnicos y de Calidad.</p>	<p><i>Usuario:</i> cualquier persona que especifica los Requisitos Funcionales de Usuario y/o cualquier persona o cosa que comunica o interactúa con el software en cualquier momento.</p> <p><i>Alcance de la medición:</i> define la funcionalidad que se incluirá en una medición de puntos función concreta.</p> <p><i>Límite del sistema:</i> borde conceptual entre el software a medir y el usuario.</p> <p><i>Proceso elemental:</i> unidad de actividad más pequeña que es significativa para el usuario. Un proceso elemental debe ser autosuficiente y dejar la aplicación en estado consistente.</p> <p><i>Archivo lógico:</i> representa la funcionalidad provista al usuario para satisfacer los requerimientos de datos externos e internos.</p>	<p><i>Actor:</i> persona, dispositivo o estructura organizacional que cumple un rol en el escenario. Cada actor representa un rol de usuario (<b>R1</b>).</p> <p><i>Alcance de la medición:</i> define la funcionalidad a incluir en una estimación de puntos función.</p> <p><i>Límite del sistema:</i> borde imaginario que encierra a todos los escenarios (<b>R2</b>).</p> <p><i>Episodio simple:</i> sentencia simple (no escenario) que representa una única acción que se ejecuta en forma completa dejando la aplicación en estado consistente (<b>R5</b>).</p> <p><i>Recurso:</i> recurso de categoría objeto lógico de tipo Símbolo Complejo o No-Símbolo (<b>R3</b>).</p>
<p><i>Tipo de BFC:</i> una categoría definida de BFC.</p>	<p><i>El:</i> <i>proceso elemental</i> que procesa datos o información de control que viene desde fuera del límite de la aplicación. El propósito principal es mantener uno o más ILF y/o modificar el comportamiento del sistema.</p> <p><i>EO:</i> <i>proceso elemental</i> que envía datos o información de control fuera del límite de la aplicación. El propósito principal es presentar información al usuario mediante una lógica de proceso distinta a, o conjuntamente con la recuperación de datos o información de control. La lógica del proceso debe contener al menos una fórmula matemática o cálculo o crear datos derivados. Una EO puede mantener uno o más ILF y/o modificar el comportamiento del sistema.</p>	<p><i>El:</i> episodio simple que procesa datos que vienen desde fuera del límite de la aplicación. El propósito principal es mantener uno o más recursos.</p> <p><i>EO:</i> episodio simple que envía datos fuera del límite de la aplicación. El propósito principal es presentar información al usuario, sean datos recuperados desde uno o más recursos o datos derivados. Puede recibir datos desde fuera del límite. Una EO no mantiene un recurso.</p>

Capítulo 4. SFP: Procedimiento de estimación de tamaño funcional de Escenarios

ISO 14143-1	IFPUG	SFP
	<p><i>EQ</i>: <i>proceso elemental</i> que envía datos o información de control fuera del límite de la aplicación. El propósito principal es presentar información al usuario mediante la recuperación de datos o información de control.</p> <p><i>ILF</i>: grupo de datos relacionados lógicamente o información de control, identificable por el usuario, mantenido dentro del límite de la aplicación. El propósito principal es almacenar datos mantenidos a través de uno o más <i>procesos elementales</i> de la aplicación que se está midiendo.</p> <p><i>EIF</i>: grupo de datos relacionados lógicamente o información de control, identificable por el usuario, referenciado por la aplicación, pero mantenido dentro del límite de otra aplicación. El propósito principal es almacenar datos referenciados mediante uno o más procesos elementales dentro del límite de la aplicación que se está midiendo.</p>	-
	<p><i>DET</i>: campo único, no repetido, identificable por el usuario.</p>	<p><i>DET</i>: unidad elemental de información, identificable por el usuario. Es un Símbolo Simple o un No-Símbolo (no recurso) (<b>R4</b>).</p>
	<p><i>FTR</i>: tipo de archivo referenciado por una función transaccional.</p>	<p><i>RTR</i>: tipo de recurso referenciado por un episodio simple.</p>
	<p><i>RET</i>: subgrupo de datos elementales, reconocible por el usuario, dentro de un ILF o EIF.</p>	<p><i>RET</i>: subgrupo de datos elementales, reconocible por el usuario, dentro de un recurso.</p>

### 4.1.3. Diseño del metamodelo

El procedimiento SFP está basado en una simplificación del metamodelo del método IFPUG FPA, tal como fue determinado a partir del análisis de conceptos y las asociaciones establecidas en la sección precedente. La Tabla 4.5 presenta la instanciación del Modelo de Referencia [NESMA 2002] (Capítulo 2, sección 2.2.7.1) para el procedimiento SFP.

Tabla 4.5. Instanciación del Modelo de Referencia para SFP

Elemento		Calificador	
	Objeto	Tipo	Descripción
	Escenarios	S	Dominio funcional: no especificado. Tipo de estimación: proyecto de desarrollo, de mejoras o de reemplazo. Alcance: las funciones identificadas en los escenarios. Límite: encierra a todos los escenarios. Propósito: estimación (costo, esfuerzo, cronograma).
Nivel	Elemento compuesto	Tipo	Descripción
-	-	-	-
Nivel	Elemento de estimación	Tipo	Descripción
1.1	Recurso		
1.2	Episodio		
Nivel	Tipo de BFC	Tipo	Descripción
1.1.1	Recurso	S Q	#DET; #RET Tabla IFPUG para ILF
1.2.1	Entrada Externa (EI)	S	#DET; #RTR
1.2.2	Salida Externa (EO)	Q	Tablas IFPUG para EI y EQ

S: especificación, Q: cuantificación

Una vez que fueron definidos los tipos de entidades que son relevantes para el tamaño funcional, deben ser descritas las relaciones entre esos tipos de entidades y las reglas para identificarlos en el artefacto de software [Abran y Jacquet 1999].

Vale destacar que los tipos de entidades que componen el metamodelo de SFP se corresponden con los conceptos definidos en el marco de la abstracción orientada a los datos propuesta por Fetcke *et al.* (2001) (Capítulo 2, sección 2.2.7.2). De acuerdo a las asociaciones conceptuales establecidas en SFP, la aplicación está representada por los escenarios, el usuario por el actor, las transacciones por los episodios y los grupos de datos por los recursos (Figura 4.2).

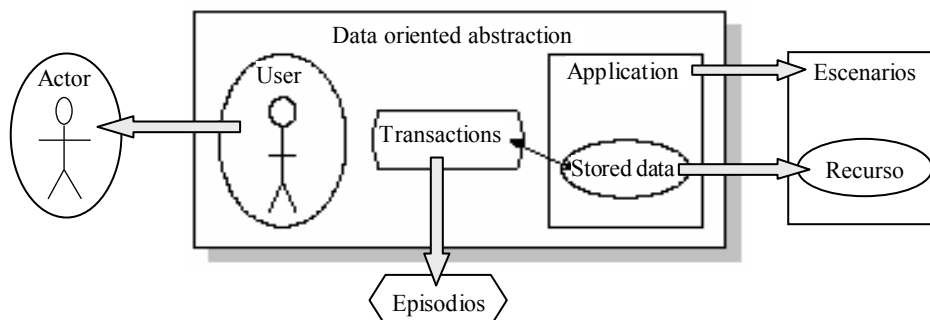


Figura 4.2. Asociación con los conceptos de la abstracción orientada a los datos de Fetcke (adaptado de [Fetcke *et al.* 2001])



#### 4.1.3.1. Relación entre los tipos de entidades y el límite

Cuando fueron definidas las EIs y EOs implícitamente fue especificada su relación con el límite del sistema. Las EIs reciben datos que vienen desde fuera del límite de la aplicación y las EOs envían datos fuera del límite de la aplicación. Esta relación no está definida para los recursos.

#### 4.1.3.2. Relación entre los tipos de entidades

El propósito principal de las EIs y EOs describe estas relaciones: un recurso debe ser mantenido por una o más EIs y puede ser referenciado, pero no mantenido, por una o más EOs.

#### 4.1.3.3. Reglas de identificación y clasificación de los tipos de entidades

En esta sección se presentan las reglas que permiten identificar los diferentes tipos de entidades - EI, EO y recurso - en los escenarios. En el caso de las EIs y EOs se requiere identificar previamente los episodios, para, posteriormente clasificarlos en una de esas categorías según su propósito.

Las reglas que se definen en esta sección son las siguientes:

- Reglas para identificar un episodio simple
- Reglas para identificar EI
- Reglas para identificar EO
- Reglas para identificar recursos

**Identificación de los Episodios.** En función de las propiedades de las diferentes clases de episodios fue establecido un conjunto de criterios (Tabla 4.6) que determinan la inclusión como *episodio candidato* (o de lo contrario su exclusión) para la estimación del tamaño funcional.

**Definición:** se denomina *episodio candidato* al episodio que está incluido en la lista de episodios inicial, pero su permanencia en la lista definitiva debe ser confirmada.

Tabla 4.6. Criterios para identificar episodios candidatos

<b>Episodios candidatos</b>
<p><b>Episodio escenario:</b> su contribución al tamaño funcional es valorada a través de sus episodios. Excluir de la lista de episodios candidatos. Ejemplo: - <b>check in</b></p>
<p><b>Excepción:</b> se refiere a una situación inusual que puede interrumpir la evolución normal del escenario. Se describe como una sentencia simple que especifica la causa de la interrupción o puede incluir el título de otro escenario que tratará la excepción. Excluir de la lista de episodios candidatos. Ejemplo: - <b>excepción:</b> Faltan tarjetas magnéticas para codificar</p>
<p><b>Restricción:</b> son aquellos aspectos que al presentarse impiden la acción. Las restricciones reflejan <i>requerimientos no funcionales</i> [Hadad <i>et al.</i> 1997] y no deben ser considerados en la medición (ver definición de <i>BFC</i> en Tabla 4.4). Excluir de la lista de episodios candidatos. Ejemplo: - <b>restricción:</b> No hay <i>disponibilidad de habitaciones</i></p>

<b>Episodios candidatos</b>
<p><b>Episodio condicional:</b> son aquéllos cuya ocurrencia depende de una condición específica. La condición puede ser interna o externa al escenario. Las condiciones internas pueden deberse a precondiciones, restricciones de actores o recursos y episodios previos [Leite <i>et al.</i> 2000]. Considerar la función requerida independientemente de la condición. Mantener en la lista de episodios candidatos.</p> <p>Ejemplo: - if existe <i>disponibilidad de habitaciones</i> y la persona estuviera de acuerdo then el <i>repcionista</i> registra los datos personales del <i>pasajero</i> en la <i>planilla de reservas</i>.</p>
<p><b>Episodio opcional:</b> son aquéllos que pueden o no tener lugar dependiendo de condiciones que pueden no estar explícitamente detalladas [Leite <i>et al.</i> 2000]. Considerar la función requerida independientemente de la condición. Mantener en la lista de episodios candidatos.</p> <p>Ejemplo: - [El <i>convocante</i> confecciona el <i>temario</i>.] [Hada <i>et al.</i> 1997]</p>
<p><b>Episodio perteneciente a un grupo no secuencial:</b> son aquéllos que tienen un orden de ejecución paralelo o no secuencial. Mantener en la lista de episodios candidatos.</p> <p>Ejemplo: - # El <i>solicitante</i> completa la <i>solicitud de adhesión</i> con sus datos personales. El <i>solicitante</i> elige un <i>bien</i> # [Mauco <i>et al.</i> 1997]</p>

Para identificar episodios simples (R5) deben ser aplicadas las siguientes reglas:

#### **Reglas para identificar un episodio simple**

- R6.** Descartar un *episodio* candidato que es un escenario.
- R7.** Descartar una sentencia que especifica una excepción.
- R8.** Descartar una sentencia que especifica una restricción.
- R9.** Aceptar un *episodio condicional* como un *episodio* candidato.
- R10.** Aceptar un *episodio opcional* como un *episodio* candidato.
- R11.** Aceptar un *episodio* de un *grupo de episodios no secuencial* como un *episodio* candidato.
- R12.** Un *episodio simple* debe referenciar uno o más *recursos* (mantener o leer).

Nota: un episodio candidato debe cumplir la Regla **R12**.

**Clasificación de los Episodios.** Los episodios simples, identificados aplicando las reglas especificadas previamente, son clasificados como EI o EO de acuerdo a su propósito principal<sup>39</sup>.

#### **Reglas para identificar EI**

- R13.** Los datos se reciben desde fuera del *límite de la aplicación*.
- R14.** Mantiene uno o más *recursos*.
- R15.** El proceso lógico es único entre los procesos lógicos realizados por otras EIs.

Nota: adaptadas de las reglas para EI de IFPUG<sup>40</sup>.

<sup>39</sup> Longstreet (2003) propone el *Vocabulario típico* que ayuda a identificar las funciones transaccionales (ver Apéndice A, sección A.2.5.2).

<sup>40</sup> En todos los casos que se indica “adaptadas”, las reglas originales pertenecen a [IFPUG 2000].

**Reglas para identificar EO**

- R16.** La función envía datos fuera del *límite de la aplicación*.
- R17.** El proceso lógico es único entre los procesos lógicos realizados por otras EOs.
- R18.** La lógica del episodio recupera datos desde un *recurso*.
- R19.** La lógica del episodio puede contener un cálculo o crear datos derivados.
- R20.** La lógica del episodio no mantiene un recurso.

Nota1: adaptadas de las reglas para EO y EQ de IFPUG.

Nota2: una EO puede incluir el ingreso de datos a través del límite si es necesario para realizar el proceso, aunque no sea el propósito principal.

Tabla 4.7. Resumen de las funciones realizadas por cada tipo de episodio (adaptada de [IFPUG 2000])

Función	Tipo de episodio	
	EI	EO
Mantener uno o más recursos	PP	N/P
Presentar información al usuario	F	PP

PP: propósito principal de un tipo de episodio

F: función de un tipo de episodio, no siendo el propósito principal y presentándose en algunas ocasiones.

N/P: función no permitida para el tipo de episodio.

**Identificación de los recursos.** En esta sección se presentan las reglas que permiten seleccionar los recursos que contribuyen al tamaño funcional. La inclusión de un recurso está determinada por el cumplimiento simultáneo de las siguientes condiciones: no duplicado, referenciado - explícitamente en el texto - en una EI o EO e incluido en la sección Recursos del escenario donde está definida la respectiva EI o EO.

**Reglas para identificar recursos**

- R21.** Aceptar como *recurso* a un *Símbolo Complejo* o *No-Símbolo* incluido en la sección Recursos del escenario y mantenido o referenciado por un *episodio simple*.
- R22.** Un *recurso* se considera una sola vez aunque sea referenciado en múltiples *episodios simples* de uno o más escenarios.

**4.1.4. Definición de las reglas de asignación numérica**

La finalidad de estas reglas es asociar valores numéricos a los objetos de software usados para caracterizar el concepto de tamaño funcional en el contexto de SFP. ISO/IEC 14143-1 (1998) establece algunos requisitos sobre la forma en que debe ser realizada la asignación numérica. Estos requisitos se refieren a [Abran y Jacquet 1999]:

1. Valoración de los BFCs: este documento especifica que el cálculo del tamaño funcional debe estar basado en la valoración de cada BFC. Por lo tanto, el método de FSM deberá:
  - definir las reglas usadas para valorar los BFCs,
  - definir cómo asignar un valor numérico a un BFC, de acuerdo a su tipo de BFC,
  - derivar el tamaño funcional a través de la valoración de los BFCs,

2. Unidades del método de medición: un método de FSM deberá definir las unidades en que es expresado el tamaño funcional.

Estas especificaciones fueron usadas como guía para la definición de las reglas de asignación numérica, las que son presentadas en la siguiente sección.

#### 4.1.4.1. Valoración de las entidades del metamodelo

En el marco de SFP las entidades que deben ser cuantificadas son las EIs, EOs y los recursos. Para ello se definen las reglas para valorar cada una de esas entidades, se establecen los criterios para asignarles un valor numérico de acuerdo a su tipo y se define una función para calcular el tamaño funcional.

Las reglas que se definen en esta sección son las siguientes:

- Reglas para contar DET en EI
  - Reglas para contar DET en EO
  - Regla para contar DET común para EI y EO
  - Reglas para contar RTR en EI
  - Reglas para contar RTR en EO
  - Reglas para contar DET en recursos
  - Regla para contar RET
  - Regla para valorar la complejidad de un recurso No-Símbolo
- **Valoración de los episodios.** En IFPUG, la contribución en FP de las funciones transaccionales depende de la complejidad (baja, media o alta), que es determinada en función del número de DETs y FTRs. De modo similar, para determinar la complejidad de las EIs y EOs en SFP, es usado el número de DETs y RTRs. En consecuencia, fue necesario definir reglas para identificar los DETs y RTRs en las EIs y EOs. Concretamente, el texto de cada EIs y EOs debe ser analizado para identificar los objetos pertenecientes a la clasificación Simple, Complejo y No-Símbolo y tipificarlos como DETs o RTRs. A continuación se presentan las reglas para identificar y contar los DETs y RTRs.

##### a. Reglas para Identificar y Contar DET en EIs y EOs

###### Reglas para contar DET en EI

**R23.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* no repetido que entra o sale del límite y es requerido para completar la EI.

**R24.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* no repetido usado en la definición de un *Símbolo Complejo* (no identificado como *recurso*) que entra o sale del límite y es requerido para completar la EI.

**R25.** No contar datos recuperados o derivados por el sistema y almacenados en un *recurso* durante un *episodio simple* si los datos no cruzaron el límite.

Nota: adaptadas de las reglas para EI de IFPUG.

Aclaración de la **R24**: los DETs de un Símbolo Complejo son determinados mediante la identificación de los Símbolos Simples y/o No-Símbolos que son usados para definirlo en

la noción (del Símbolo Complejo). Si en su definición es detectado otro Símbolo Complejo debe ser repetido el procedimiento mencionado.

Ejemplo de Símbolo Complejo

- El *repcionista* actualiza la *disponibilidad de habitaciones* en la *planilla de ocupación de habitaciones*.

La noción del símbolo *disponibilidad de habitaciones* establece: contiene la cantidad y tipo de *habitación* desocupada. Por lo tanto se deben considerar los DETs cantidad y tipo de *habitación*.

#### Reglas para contar DET en EO

**R26.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* que entra a la aplicación y es requerido para especificar cuándo, qué y cómo los datos son recuperados o generados por el *episodio simple*.

**R27.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* que sale del límite de la aplicación.

**R28.** Si un DET entra y sale se cuenta una vez.

**R29.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* usado en la definición de un *Símbolo Complejo* (no identificado como *recurso*) que sale del límite de la aplicación.

**R30.** No contar datos recuperados o derivados por el sistema durante un *episodio simple* si los datos no cruzaron el límite.

Nota: adaptadas de las reglas para EO y EQ de IFPUG.

#### Regla para contar DET común para EI y EO

**R31.** No contar como DET a un *Símbolo* que representa un *actor*.

#### b. Reglas para Identificar y Contar RTR en EIs y EOs

##### Regla para contar RTR en EI

**R32.** Contar 1 RTR por cada *recurso Símbolo Complejo* o *No-Símbolo* mantenido, leído o mantenido y leído por la EI.

Nota: adaptada de las reglas para EI de IFPUG.

##### Regla para contar RTR en EO

**R33.** Contar 1 RTR por cada *recurso Símbolo Complejo* o *No-Símbolo* leído por la EO.

Nota: adaptada de las reglas para EO y EQ de IFPUG.

c. **Evaluar la Contribución de las EIs y EOs.** Los números de DETs y RTRs de las EIs y EOs, calculados aplicando las reglas precedentes, son usados como entrada a las tablas para EI y EO (Tablas 4.8 y 4.9) para obtener la complejidad y los FP. Esas tablas están basadas en las propuestas para las EIs y EQs en el manual de IFPUG (2000).

Tabla 4.8. Complejidad y Contribución para EI (adaptada de [IFPUG 2000])

RTR	DET		
	1- 4	5 - 15	16 o más
1 <sup>(*)</sup>	Baja (3)	Baja (3)	Media (4)
2	Baja (3)	Media (4)	Alta (6)
3 o más	Media (4)	Alta (6)	Alta (6)

Tabla 4.9. Complejidad y Contribución para EO (adaptada de [IFPUG 2000])

RTR	DET		
	1 - 5	6 - 19	20 o más
1 <sup>(*)</sup>	Baja (3)	Baja (3)	Media (4)
2 o 3	Baja (3)	Media (4)	Alta (6)
4 o más	Media (4)	Alta (6)	Alta (6)

(\*) el valor original (0-1) de la Tabla de IFPUG fue modificado para adaptarlo a las reglas de SFP, ya que el valor 0 no es posible debido a que la referencia al recurso debe ser explícita y por lo tanto habrá, como mínimo, 1 RTR.

- **Valoración de los recursos.** En IFPUG, la contribución en FP de los archivos lógicos es determinada de acuerdo a su complejidad (baja, media o alta), la cual es evaluada en función del número de DETs y RETs. De modo similar, para determinar la complejidad de los recursos, en SFP es usado el número de DETs y RETs. A continuación se presentan las reglas para identificar y contar los DETs y RETs.

a. **Reglas para Identificar y Contar DET en los recursos.** El tratamiento de los recursos depende de la clase de objeto del LEL:

- Si es un Símbolo Complejo, los DETs se determinan a partir de la noción del Símbolo del LEL, es decir se obtienen los DET que forman parte de su definición. Si en su definición se detecta otro Símbolo Complejo debería repetirse el procedimiento (el tratamiento es el mismo que el indicado en la **R24**).
- Cuando se trata de objetos No-Símbolo no es posible identificar DET.

#### Reglas para contar DET en recursos

**R34.** Si el *recurso* es *No-Símbolo* no hay DET.

**R35.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* mantenido en o recuperado desde un *recurso* mediante la ejecución de un *episodio simple*.

**R36.** Contar 1 DET por cada *Símbolo Simple* o *No-Símbolo* usado en la definición del *Símbolo Complejo* identificado como *recurso*.

Nota: R35 adaptada de las reglas para ILF/EIF de IFPUG.

b. **Reglas para Identificar y Contar RET en los recursos.** El nivel de detalle disponible en el L&E no permite identificar diferentes RET en un recurso, por lo tanto se considera que todos tienen un RET. Un criterio similar es adoptado en el marco de Early Function Points; Santillo y Meli (1998) destacan la dificultad para agrupar los campos de un archivo lógico cuando no se conoce exactamente cuáles son, y en consecuencia, sugieren que cuando no hay información detallada confiable es aplicable la regla de “máxima simplicidad”, es decir, si no hay indicios de la existencia de varios RET, hay solamente un RET.

**Regla para contar RET**

**R37.** Contar 1 RET por *recurso*.

c. **Evaluar la Contribución de los Recursos.** La complejidad de los recursos depende de la clase de objeto del LEL.

- Para los recursos de tipo Símbolo Complejo es determinada en función de 1 RET y el número de DETs calculado aplicando las reglas precedentes.

- A los recursos de tipo No-Símbolo se les asigna complejidad baja. Este criterio es adecuado considerando que la aplicación del enfoque sobre un conjunto de casos de estudio arrojó como resultado intermedio que todos los recursos de tipo Símbolo Compuesto tienen complejidad baja. Por otro lado, también es consistente con las propuestas de otros autores para las etapas tempranas, cuando la información disponible es insuficiente para identificar atributos. En ese sentido, por ejemplo, Longstreet (2003) propone asignarle complejidad media a todas las funciones de datos.

**Regla para valorar la complejidad de un recurso No-Símbolo**

**R38.** Un *recurso* de tipo *No-Símbolo* siempre tiene complejidad *Baja*.

El número de DETs y 1 RET - de un recurso de tipo Símbolo Complejo - son usados como valores de entrada a una tabla para determinar los FP. Esa tabla es una simplificación de la homónima para ILF (usa sólo la entrada para 1 RET) de IFPUG (2000). Los recursos de tipo No-Símbolo siempre aportan 7 FP.

Tabla 4.10. Complejidad y Contribución de los recursos (adaptada de [IFPUG 2000])

RET	DET		
	1 a 19	20 - 50	51 o más
1	Baja (7)	Baja (7)	Media (10)

RET	DET	
	1 a 50	51 o más
1	Baja (7)	Media (10)

- **Función de medición.** El total de FPs es obtenido aplicando la siguiente fórmula:

$$FP_{SFP} = \sum_{j=1}^n FP_{EI_j} + \sum_{k=1}^p FP_{EO_k} + \sum_{i=1}^m FP_{R_i} \quad (1)$$

siendo: n: #EI, p: #EO, m: #recursos.

**4.1.4.2. Unidades del método de estimación**

La unidad es 1 Punto Función (FP). Los resultados deben ser presentados indicando el valor del tamaño funcional seguido por las unidades y el nombre del método, por ejemplo, 200 FP (SFP).

**4.2. Aplicación del Procedimiento SFP**

En esta sección se presenta el proceso SFP y un ejemplo resumido de la aplicación sobre un caso de estudio.

### 4.2.1. El Proceso SFP

El Modelo de Proceso para los Métodos de Medición del Software [Jacquet y Abran 1997] (Capítulo 2, sección 2.2.7.3) y el documento ISO 14143-1 [ISO 1998] (Capítulo 2, sección 2.2.4) especifican los pasos para aplicar un método de medición. Con el mismo objetivo, se definió un proceso que consta de varios pasos, algunos de los cuales deben ser ejecutados secuencialmente y otros soportan la ejecución en paralelo. A través de los diferentes pasos se aplican las reglas de identificación, clasificación y asignación numérica así como la función de medición presentadas en las secciones previas. A continuación se enumeran los pasos del proceso SFP junto con una breve descripción de las actividades.

1. *Identificar el límite del sistema*: construir una lista con los episodios de los escenarios del sistema.
2. *Identificar episodios*: aplicar las reglas de identificación de episodios para determinar el conjunto definitivo de episodios.
3. *Clasificar episodios*: aplicar las reglas para clasificar los episodios como EI y EO.
4. *Determinar complejidad y contribución de los episodios*: aplicar las reglas para contar DET y RTR en EI y EO y usarlos como valores de entrada en las Tablas para EI y EO.
5. *Identificar recursos*: examinar los recursos de cada episodio y construir la lista definitiva de recursos.
6. *Determinar complejidad y contribución de los recursos*: aplicar las reglas para contar DET en recursos y usar este valor y  $\#RET = 1$  como valores de entrada a la Tabla para recursos.
7. *Calcular el tamaño funcional*: aplicar la función (1) de la sección 4.1.4.1 para determinar el tamaño funcional.

Nota: los pasos 3-4 y 5-6 pueden ser realizados en paralelo.

En la Figura 4.3 se representan los pasos del proceso SFP. En dicha figura, además, haciendo un análisis comparativo de los procesos similar al realizado por Abrahão *et al.* (2004b), se muestra que este proceso puede ser asociado con el modelo de Fetcke *et al.* (2001), el paso 2 del Modelo de Proceso para los métodos de FSM de Jacquet y Abran (1997) y el proceso especificado en el estándar ISO 14143-1 (Capítulo 2, secciones 2.2.7.2, 2.2.7.3 y 2.2.4, respectivamente). Cabe señalar que esta relación entre SFP y los demás procesos se basa en la asociación de conceptos establecida en el marco de SFP y la justificación presentada en la definición de los objetivos (sección 4.1.1). Esto se debe a que los escenarios no son una SRS y en consecuencia no se pueden identificar los FURs ni los BFCs (o al menos en esa etapa no se puede asegurar que los componentes identificados vayan a serlo cuando se desarrolle la SRS) como es requerido en el contexto de una medición de tamaño funcional.



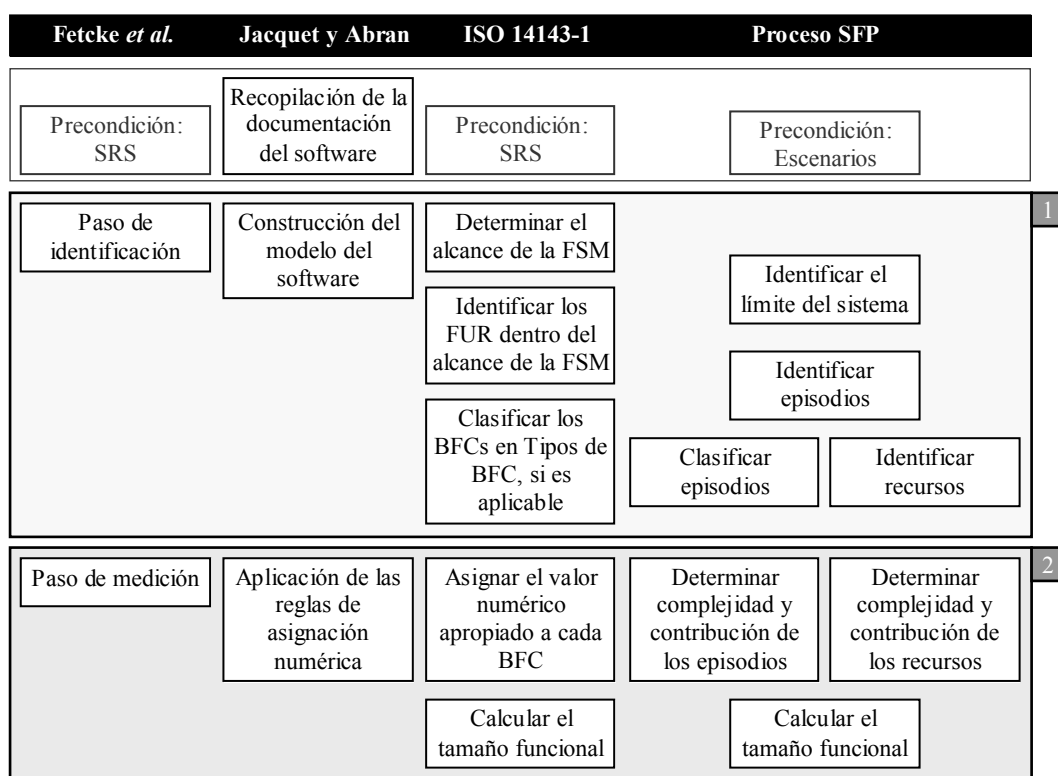


Figura 4.3. El proceso SFP comparado con otros modelos (adaptado de [Abrahão *et al.* 2004b])

Como puede observarse, los pasos han sido agrupados y etiquetados con “1” y “2”, esto fue realizado al sólo efecto de facilitar la explicación y visualización de las asociaciones sugeridas. Por ejemplo, el grupo “1” del proceso SFP contiene las actividades que permiten identificar los componentes funcionales en los escenarios y puede ser asociado con el paso de identificación del modelo de Fetcke *et al.*; el grupo “2” incluye las actividades requeridas para determinar el tamaño funcional de los escenarios y se corresponde con el paso de medición del mencionado modelo. Los tres procesos de la Figura 4.3 con los que fue comparado SFP requieren, explícita o implícitamente, la ejecución de un paso previo en el cual es obtenido el artefacto de software (por ejemplo, la SRS) al cual será aplicada la medición y que sólo en el Modelo de Jacquet y Abran es especificado como el primer paso del proceso.

En lo concerniente a la ejecución del proceso, si bien la descripción de los pasos puede servir como guía, la realización efectiva del proceso requiere de una definición del propósito de cada paso, los agentes involucrados y las habilidades demandadas, las actividades comprendidas, la secuencia en que deben ser ejecutadas, las pre y poscondiciones de cada paso, los procedimientos y las reglas que deben ser aplicados, los recursos necesarios para transformar los productos de entrada (inicial/intermedios) en nuevos productos de salida (intermedios/final) y las herramientas y medios de soporte para el manejo y almacenamiento de la información generada durante el proceso. En concreto, todo lo anterior, significa proveer una especificación del proceso, la que es presentada en el Capítulo 5.

#### 4.2.2. Un ejemplo de aplicación del proceso SFP

SFP fue diseñado para aplicarlo a los escenarios derivados desde el LEL. Esto significa que para iniciar la estimación es precondition que dichos escenarios estén disponibles. Como es de esperar, la calidad de estos documentos tiene impacto sobre los resultados a obtener. La flexibilidad que ofrece el uso del lenguaje natural para facilitar la interacción de los participantes en el proceso de captura de los requisitos, tiene como contrapartida la dificultad para su validación, ya que la presencia de errores, discrepancias u omisiones es más difícil de detectar [Doorn *et al.* 1998]. Con el objetivo de facilitar la detección de errores y mejorar la calidad de los escenarios producidos, Doorn *et al.* (1998) propusieron un mecanismo de inspección de escenarios que consiste básicamente en la realización de verificaciones cruzadas entre los escenarios y el LEL y entre los escenarios mismos. Por lo tanto, para una efectiva aplicación de SFP es aconsejable que los escenarios hayan superado dicho proceso de revisión.

En las siguientes secciones se presenta un resumen de la aplicación del proceso SFP sobre los escenarios desarrollados durante la fase de Elicitación de Requerimientos del software SFP Tool [Bertolami *et al.* 2008]. En el Apéndice D se incluye el caso completo y en el Apéndice E se presentan los detalles de esta herramienta.

Durante la primera parte del proceso (señalada con “1” en la Figura 4.3), aplicando las reglas de identificación, son reconocidos los componentes (episodios y recursos) en los escenarios del sistema. En la segunda parte (señalada con “2” en la misma figura), aplicando las reglas de asignación numérica, son cuantificados los componentes identificados y desde ellos, función de medición mediante, es calculado el tamaño funcional del sistema.

##### 4.2.2.1. Identificar el límite del sistema

Para realizar la estimación de FP, primero es necesario determinar cuales son las funciones del software que van a ser cuantificadas. De acuerdo a la regla **R2** deben ser considerados todos los escenarios del sistema bajo estudio. Por lo tanto, todos los episodios de esos escenarios son usados para construir un listado que es almacenado en la Planilla de episodios (Formulario 1). Las plantillas de los formularios que son usados de aquí en adelante son presentadas en el Apéndice C.

##### 4.2.2.2. Identificar episodios

El objetivo de este paso es determinar los episodios que cumplen las condiciones para ser usados en la estimación de tamaño funcional. Para ello son aplicadas las reglas de identificación de episodios **R5 .. R12** (sección 4.1.3.3) por cada episodio de la Planilla de episodios. Paralelamente, sobre los episodios identificados es aplicada la regla **R21** para comprobar los recursos. De este modo son seleccionados los episodios candidatos y descartados los restantes, es decir aquéllos que no cumplen las respectivas reglas (estos episodios son etiquetados con una “D” en la Planilla de episodios).

##### 4.2.2.3. Clasificar episodios

En este paso, los episodios candidatos son clasificados como EI (reglas **R13 .. R15**) o EO (reglas **R16 .. R20**) (sección 4.1.3.3). En esta fase puede ocurrir que un episodio candidato deba ser eliminado de la lista debido a que la funcionalidad por él representada ha sido considerada previamente (reglas **R15** o **R17**), esto es, son removidos los episodios duplicados). En la Tabla 4.11 se presenta un fragmento de la Planilla de episodios con los episodios clasificados como EI, EO o D y las razones que sustentan cada elección (cada

superíndice en la columna Tipo se corresponde con la respectiva explicación debajo de la tabla).

Tabla 4.11. Fragmento del Formulario 1: Identificación y clasificación de episodios

Formulario 1		SFP Planilla de episodios	
TÍTULO DE LA APLICACIÓN: SFP Tool			
AUTOR: Mabel		FECHA: 14/06/08	
Escenario	ID	Episodio	Tipo
...	...	...	...
<b>clasificar episodios</b>		El <i>autor</i> ingresa la <i>fecha de inicio</i> , <i>hora de inicio</i> y paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	D <sup>1</sup>
	5	El <i>autor</i> clasifica como <i>EI</i> o <i>EO</i> cada <i>episodio</i> con <i>ID</i> de la <i>lista de episodios</i> aplicando las <i>Reglas para clasificar episodios</i> .	EO <sup>2</sup>
	6	El <i>autor</i> almacena el <i>tipo</i> de los <i>episodios</i> ( <i>EI</i> , <i>EO</i> , <i>D</i> ) en el <i>Formulario 1</i> .	EI <sup>3</sup>
		El <i>autor</i> ingresa la <i>fecha de finalización</i> y <i>hora de finalización</i> de ese paso.	D <sup>4</sup>
		Se calcula el intervalo de tiempo transcurrido.	D <sup>5</sup>
		Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D <sup>6</sup>
		<b>restricción:</b> No se cuenta con las <i>Reglas para clasificar episodios</i> .	D <sup>7</sup>
...	...	...	...

<sup>1, 6</sup> El episodio ya fue considerado en el escenario **calcular tamaño funcional**. En este escenario debe ser descartado por estar duplicado. No cumple la regla **R15**.

<sup>2</sup> El recurso *Reglas para clasificar episodios* referenciado por este episodio cumple la regla **R21**. El texto del episodio es usado como entrada para consultar el recurso y obtener como salida el tipo de episodio. Cumple **R5** y **R12**, por lo tanto es episodio candidato. Cumple **R16**, **R17**, **R18** y **R20**, por lo tanto es EO.

<sup>3</sup> El episodio referencia el recurso *Formulario 1* que cumple la regla **R21**. El proceso modifica el recurso. Cumple **R5** y **R12**, por lo tanto es episodio candidato. Cumple **R13**, **R14** y **R15**, por lo tanto es EI.

<sup>4, 5</sup> El episodio no referencia un recurso, no cumple la regla **R12**.

<sup>7</sup> De acuerdo a **R8** las restricciones deben ser excluidas.

#### 4.2.2.4. Identificar recursos

El objetivo de este paso es determinar la lista de recursos que serán considerados en la estimación de FP. Para ello, cada recurso incluido en un episodio (regla **R12**) de la lista definitiva y que cumpla la regla **R22** es incorporado a una lista (esta regla asegura que no existirán recursos duplicados) que es almacenada en el Formulario 3.

En el cuadro siguiente se presenta el análisis detallado de la EI identificada con el ID=6 en la Tabla 4.11 y su respectivo recurso.

Episodio: El <u>autor</u> almacena el tipo de los <u>episodios</u> ( <u>EI</u> , <u>EO</u> , <u>D</u> ) en el <u>Formulario 1</u> .	
a. Identificar un <i>episodio simple</i> a partir de un episodio candidato	
El episodio candidato es un <i>episodio simple</i> porque:	
<ul style="list-style-type: none"> <li>- Cumple la regla <b>R5</b>.</li> <li>- No cumple las reglas <b>R6 .. R8</b>.</li> <li>- Cumple <b>R12</b>. Referencia el recurso <u>Formulario 1</u>.</li> </ul>	
b. Clasificar el <i>episodio simple</i> según su propósito principal	
El episodio simple es clasificado como EI porque:	
<ul style="list-style-type: none"> <li>- Cumple <b>R13</b>. Los datos se reciben desde fuera del límite de la aplicación.</li> <li>- Cumple <b>R14</b>. Mantiene el recurso <u>Formulario 1</u>.</li> <li>- Cumple <b>R15</b>. Es el único episodio que realiza este proceso lógico.</li> </ul>	
c. Identificar un recurso	
<ul style="list-style-type: none"> <li>- Cumple <b>R14</b>. Es mantenido por una EI.</li> <li>- Cumple <b>R21</b>. <u>Formulario 1</u> es un Símbolo Complejo que está incluido en la sección Recursos del escenario.</li> </ul>	

**4.2.2.5. Determinar complejidad y contribución de los episodios**

Para determinar la contribución en FP de los componentes identificados son aplicadas las reglas de asignación numérica (sección 4.1.4). En este paso es evaluada la contribución de las EIs y EOs. Para ello se requiere identificar y contabilizar los DETs y RTRs en cada EI y EO, para lo que son aplicadas las reglas para identificar y contar DETs en EIs y EOs (reglas **R23 .. R31**) y las reglas para identificar y contar RTR en EIs y EOs (reglas **R32 .. R33**) (sección 4.1.4.1). Los números de DETs y de RTRs de cada EI y EO son usados como valores de entrada a las Tablas para EI y EO (Tablas 4.8 y 4.9) respectivamente, para determinar la complejidad (baja, media, alta) y la contribución en FPs.

El Formulario 2 es usado para registrar las EIs y EOs con sus respectivos DETs y RTRs. En la Tabla 4.12 se presenta la aplicación de reglas para la misma EI que fue analizada previamente.

Tabla 4.12. Fragmento del Formulario 2: Complejidad de los episodios

<b>Formulario 2</b>	<b>SFP</b>	
	<b>Complejidad de los episodios</b>	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b>	Mabel	<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>Nº REVISIÓN:</b>
...		
<b>ESCENARIO:</b>	<b>clasificar episodios</b>	
<b>ID:</b> 6	<b>RTR</b>	<b>TIPO:</b> EI
		<b>DET</b>
	<u>Formulario 1</u>	<u>episodio</u>
		<u>tipo</u>
<b>TOTAL:</b>	1	<b>TOTAL:</b> 2
	<b>Complejidad</b>	Baja
...		

A continuación se presenta el análisis detallado de las reglas aplicadas para determinar los RTRs y DETs de la EI de la Tabla 4.12.

<p>EI: El <u>autor</u> almacena el <u>tipo</u> de los <u>episodios</u> (<u>EL</u>, <u>EQ</u>, D) en el <u>Formulario 1</u>.                  Semántica: cada <u>episodio</u> tiene asociado un tipo (<u>EL</u>, <u>EQ</u>, D) que debe ser mantenido en el <u>Formulario 1</u>.</p> <p>a. Identificar y contar RTR</p> <p><u>Formulario 1</u> es un recurso mantenido por la EI.                  - Según la regla <b>R32</b> se debe contar 1 RTR.</p> <p>b. Identificar y contar DET</p> <p><u>episodio</u> y <u>tipo</u> son DET de entrada porque:                  - pertenecen a la categoría Símbolo Simple y se reciben desde fuera del límite de la aplicación. Cumplen <b>R23</b>.</p> <p>c. Evaluar la complejidad</p> <p>Se identificaron:                  - 1 RTR, 2 DETs</p> <p>Usando este par de valores como entrada a la tabla para EI se determina complejidad <i>baja</i> y una contribución de 3 FP.</p>
--

#### 4.2.2.6. Determinar complejidad y contribución de los recursos

Para evaluar la contribución de los recursos se requiere identificar y contabilizar los DETs en cada recurso, para lo cual son aplicadas las reglas para identificar y contar DET en los recursos (reglas **R34** .. **R36**) (sección 4.1.4.1). La regla para contar RET (**R37**) establece que hay 1 RET por recurso, por lo tanto, el número de DETs y 1 RET son usados como valores de entrada a la Tabla para recursos (Tabla 4.10), para determinar la complejidad (baja, media) y la contribución en FPs.

El Formulario 3 es usado para registrar los recursos con sus respectivos DETs. En la Tabla 4.13 se presenta la aplicación de reglas para el mismo recurso que fue analizado previamente.

Tabla 4.13. Fragmento del Formulario 3: Complejidad de los recursos

Formulario 3		SFP	
Complejidad de los recursos			
TÍTULO DE LA APLICACIÓN: SFP Tool			
AUTOR: Mabel		REVISOR:	
ETAPA DEL PROYECTO: Elicitación		FECHA DE REVISIÓN:	
N° REVISIÓN:			
...			
Nombre:	<u>Formulario 1</u>	RET	1
DET			
<u>Autor</u>			
título de la aplicación			
Fecha			
nombre del <u>escenario</u>			
<u>ID</u>			
<u>tipo</u>			
<u>Episodio</u>			
...			
TOTAL	7	Complejidad	Baja
...			

A continuación se presenta el análisis detallado de las reglas aplicadas para determinar los DETs y RET del recurso de la Tabla 4.13.

<p>Recurso: <u>Formulario 1</u></p> <p>a. Contar RET</p> <ul style="list-style-type: none"> <li>- Según la regla <b>R37</b> se debe contar 1 RET por recurso.</li> </ul> <p>b. Identificar y contar DET</p> <p><u>autor</u>, <u>ID</u>, <u>tipo</u>: <u>episodio</u>, nombre del <u>escenario</u>, título de la aplicación, fecha son DET porque:</p> <ul style="list-style-type: none"> <li>- pertenecen a las categorías Símbolo Simple (<u>autor</u>, <u>ID</u>, <u>tipo</u>, <u>episodio</u>) y No-Símbolo (título de la aplicación, nombre del <u>escenario</u>, fecha). Cumplen <b>R35</b>.</li> </ul> <p>c. Evaluar la complejidad</p> <p>Se identificaron:</p> <ul style="list-style-type: none"> <li>- 1 RET, 7 DETs</li> </ul> <p>Usando este par de valores como entrada a la tabla para recursos se determina complejidad</p>
---

#### 4.2.2.7. Calcular el tamaño funcional

Una vez realizado el procesamiento de todas las EIs, EOs y recursos es aplicada la fórmula (1) (sección 4.1.4.1) para calcular el tamaño funcional. En la Tabla 4.14 se presenta el resumen de la estimación, indicando el número de EI, EOs y recursos por cada nivel de complejidad, la contribución en FP y el total de FP del sistema SFP Tool.

Tabla 4.14. Resultado de la estimación

Formulario 4	SFP			
Cálculo de FP no Ajustados				
TÍTULO DE LA APLICACIÓN: SFP Tool				
AUTOR: Mabel		REVISOR:		
ETAPA DEL PROYECTO: Elicitación		FECHA DE REVISIÓN:		
Tipo de Función	Complejidad Funcional			Totales por complejidad
				Totales por Tipo de Función
Recursos	17	Baja	x 7	119
		Media	x 10	0
				119
EI	24	Baja	x 3	72
		Media	x 4	0
		Alta	x 6	0
				72
EO	7	Baja	x 3	21
	2	Media	x 4	8
	1	Alta	x 6	6
				35
<b>Total FP no Ajustados (SFP)</b>				<b>226</b>

### **4.3. Resumen y Conclusiones**

En este capítulo fue presentado un enfoque para estimar el tamaño funcional de los escenarios. El procedimiento SFP fue desarrollado en base a la asociación del modelo de escenarios con los conceptos del Método IFPUG FPA. SFP fue diseñado siguiendo los pasos establecidos en el Modelo de Proceso para los Métodos de Medición de Jacquet y Abran y orientado a lograr la mayor adhesión posible a las definiciones, conceptos y lineamientos del estándar ISO 14143-1.

La aplicación de SFP es guiada por un proceso que consta de siete pasos a través del cual se aplican reglas de identificación de episodios y recursos, reglas de clasificación de los episodios según su propósito, reglas de asignación numérica y una función para derivar el tamaño funcional. La comparación del proceso definido con otros modelos reconocidos demostró que el mismo se encuadra en los lineamientos generales establecidos para un método de medición de tamaño funcional.





## Capítulo 5

### Especificación del Proceso SFP

Un aspecto esencial para mejorar la calidad de los procesos software en una organización es desarrollar y usar procesos definidos. Un “proceso definido” es aquél que está documentado, es soportado (por ejemplo, mediante capacitación) y es aplicado con regularidad [Olson *et al.* 1993].

En este capítulo se presenta la especificación del proceso SFP que fue desarrollada con la finalidad de facilitar el aprendizaje, el entrenamiento y la eficiente ejecución. Este último aspecto está ligado a las facilidades proporcionadas para agilizar el proceso. Con ese objetivo fue diseñado un conjunto de formularios estándar y fue desarrollada una herramienta específica que soporta el proceso.

#### 5.1. Diseño de la plantilla para especificar el proceso SFP

La plantilla propuesta para especificar el proceso SFP (Capítulo 4, sección 4.2.1) fue definida a partir de las plantillas propuestas por Olson *et al.* (1993) en el Software Process Framework (SPF) [Olson *et al.* 1993] (Capítulo 2, sección 2.3 y Apéndice B) y adecuada al proceso que se está definiendo.

ID	Nombre	
Propósito		
Roles		
Nombre	Responsabilidad	Entrenamiento
Criterios de entrada		
Estado o Condición		Desde actividad
Entradas		
Producto		Actividad fuente
Actividades		
Paso	Nombre	
Recursos		
Formularios		
Procedimientos		
Estándares		
Criterios de salida		
Estado o Condición		Hacia actividad
Salidas		
Producto		Actividad destino
Herramientas		

Mediciones	
Variable	Descripción
Observaciones	

En el siguiente cuadro se presenta la definición de los elementos de la plantilla.

<b>ID:</b> código de identificación de la actividad		<b>Nombre:</b> denominación que identifica a la actividad
<b>Propósito:</b> sentencia que especifica una finalidad concreta a alcanzar mediante la ejecución de la actividad.		
<b>Roles</b>		
<b>Nombre:</b> denominación del rol.	<b>Responsabilidad:</b> alcance de la actividad a ejecutar.	<b>Entrenamiento:</b> conocimiento y habilidades requeridas para cumplir el rol.
<b>Criterios de entrada</b>		
<b>Estado o Condición:</b> predicado simple o compuesto que establece el estado de un producto o actividad.		
<b>Entradas</b>		
<b>Producto:</b> nombre del producto o artefacto intermedio o final producido por una actividad.	<b>Actividad fuente:</b> actividad que produce este producto o artefacto.	
<b>Actividades</b>		
<b>Paso:</b> número de orden que indica la secuencia de ejecución.	<b>Nombre:</b> denominación de la subactividad.	
<b>Recursos:</b> describe las entidades necesarias para soportar la ejecución de la actividad.		
<b>Documentos / Formularios:</b> documentación que incluye la información necesaria para ejecutar la actividad y los formularios estándar (ver sección 5.4) para la recolección de los datos y las mediciones.	<b>Procedimientos:</b> conjunto de instrucciones paso a paso para ejecutar una función específica.	
<b>Estándares:</b> establecen las definiciones operacionales o criterios de aceptación para los productos intermedios y final del proceso. Describe una pauta o norma que regula la realización de una determinada acción.		
<b>Criterios de salida</b>		
<b>Estado o Condición:</b> predicado simple o compuesto que establece el estado de un producto o actividad.	<b>Hacia actividad:</b> actividad destino para este estado.	
<b>Salidas</b>		
<b>Producto:</b> nombre del producto o artefacto.	<b>Actividad destino:</b> actividad que produce este producto o artefacto.	
<b>Herramientas:</b> listado de herramientas que soportan el proceso, puede ser un software estándar de planilla de cálculo o específico para el proceso (ver sección 5.5).		
<b>Mediciones</b>		
<b>Variable:</b> identificador utilizado para representar el esfuerzo.	<b>Descripción:</b> medida del esfuerzo en horas-persona. (ver sección 5.3)	
<b>Observaciones:</b> comentarios o aclaraciones necesarios para ejecutar la actividad.		

## 5.2. Especificación del Proceso SFP

En esta sección se presenta la especificación de los pasos del proceso SFP aplicando la plantilla propuesta (sección 5.1).

ID	Nombre	
1	Identificar límite del sistema	
<b>Propósito</b>		
Determinar la lista de episodios que será considerada para la estimación de los FPs.		
<b>Roles</b>		
<b>Nombre</b>	<b>Responsabilidad</b>	<b>Entrenamiento</b>
Analista de Sistemas /Ing. de software	<ul style="list-style-type: none"> <li>• Ejecutar actividad</li> <li>• Revisar productos y procedimientos</li> </ul>	<ul style="list-style-type: none"> <li>• Manejo de L&amp;E</li> <li>• Conocimiento del método IFPUG</li> <li>• Manejo del procedimiento SFP</li> </ul>
<b>Criterios de entrada</b>		
<b>Estado o Condición</b>		
LEL y Escenarios definido y acordado con el usuario. LEL y Escenarios disponibles. Herramienta de soporte disponible. Personal entrenado en SFP.		
<b>Entradas</b>		
<b>Producto</b>	<b>Actividad fuente</b>	
Escenarios	Elicitación de Requerimientos	
<b>Actividades</b>		
<b>Paso</b>	<b>Nombre</b>	
1	Construir la lista de episodios	
<b>Recursos</b>		
<b>Formularios</b>		
Formulario 1 Formulario 5		
<b>Procedimientos</b>		
Construir la lista de episodios Almacenar fecha y hora de inicio en el Formulario 5. Seleccionar cada escenario e incorporar su nombre y todos sus episodios en la Planilla de episodios. Almacenar fecha y hora de finalización en el Formulario 5.		
<b>Estándares</b>		
<b>Reglas</b> R2. Aceptar como <i>límite del sistema</i> al borde imaginario que encierra a todos los escenarios.		
<b>Normas para completar formularios</b> Formulario 1 Formulario 5		
<b>Criterios de salida</b>		
<b>Estado o Condición</b>		
Todos los episodios fueron incorporados a la lista de episodios. La lista de episodios fue revisada.		
<b>Salidas</b>		
<b>Producto</b>	<b>Actividad destino</b>	
Lista de episodios	2 Identificar episodios	
<b>Herramientas</b>		
Software de planilla de cálculo o software SFP Tool (ver Apéndice E)		
<b>Mediciones</b>		
<b>Variable</b>	<b>Descripción</b>	
E1	Esfuerzo (horas-persona) insumido por la actividad	
<b>Observaciones</b>		

ID	Nombre	
2	<i>Identificar episodios</i>	
<b>Propósito</b>		
<i>Determinar la lista de episodios.</i>		
<b>Roles</b>		
Nombre	Responsabilidad	Entrenamiento
<i>Analista de Sistemas /Ing. de software</i>	<ul style="list-style-type: none"> <li>• <i>Ejecutar actividad</i></li> <li>• <i>Revisar productos y procedimientos</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Manejo del procedimiento SFP</i></li> </ul>
<b>Criterios de entrada</b>		
<b>Estado o Condición</b>		
<i>Lista de episodios disponible. Herramienta de soporte disponible. Personal entrenado en SFP.</i>		
<b>Entradas</b>		
Producto	Actividad fuente	
<i>Lista de episodios</i>	<i>1 Identificar límite del sistema</i>	
<b>Actividades</b>		
Paso	Nombre	
<i>1</i>	<i>Analizar episodios</i>	
<b>Recursos</b>		
<b>Formularios</b>		
<i>Formulario 1 Formulario 5</i>		
<b>Procedimientos</b>		
<u>Analizar episodios</u> Almacenar fecha y hora de inicio en el Formulario 5. Evaluar cada episodio en la Planilla de episodios y determinar su permanencia en la lista de episodios de acuerdo a lo establecido por las reglas de identificación de episodios y recursos. Almacenar fecha y hora de finalización en el Formulario 5.		
<b>Estándares</b>		
<b>Reglas</b> R6. Descartar un <i>episodio</i> candidato que es un escenario. R7. Descartar una sentencia que especifica una excepción. R8. Descartar una sentencia que especifica una restricción. R9. Aceptar un <i>episodio condicional</i> como un <i>episodio</i> candidato. R10. Aceptar un <i>episodio opcional</i> como un <i>episodio</i> candidato. R11. Aceptar un <i>episodio</i> de un <i>grupo de episodios no secuencial</i> como un <i>episodio</i> candidato. R12. Un <i>episodio simple</i> debe referenciar uno o más <i>recursos</i> (mantener o leer). R21. Aceptar como <i>recurso</i> a un <i>Símbolo Complejo</i> o <i>No-Símbolo</i> incluido en la sección Recursos del escenario y mantenido o referenciado por un <i>episodio simple</i> .		
<b>Normas para completar formularios</b> <i>Formulario 1 Formulario 5</i>		
<b>Criterios de salida</b>		
<b>Estado o Condición</b>		
<i>Se identificaron todos los episodios</i>		
<b>Salidas</b>		
Producto	Actividad destino	
<i>Lista de episodios</i>	<i>3 Clasificar episodios</i>	
<b>Herramientas</b>		
<i>Software de planilla de cálculo o software SFP Tool</i>		

Mediciones	
Variable	Descripción
E2	Esfuerzo (horas-persona) insumido por la actividad
Observaciones	

ID	Nombre	
3	Clasificar episodios	
Propósito		
Clasificar los episodios como EIs y EOs		
Rol		
Nombre	Responsabilidad	Entrenamiento
Analista de Sistemas /Ing. de software	<ul style="list-style-type: none"> <li>Ejecutar actividad</li> <li>Revisar productos y procedimientos</li> </ul>	<ul style="list-style-type: none"> <li>Manejo del procedimiento SFP</li> </ul>
Criterio de entrada		
Estado o Condición		
Lista de episodios disponible y sus episodios identificados. Herramienta de soporte disponible. Personal entrenado en SFP.		
Entrada		
Producto	Actividad fuente	
Lista de episodios	2 Identificar episodios	
Actividades		
Paso	Nombre	
1	Determinar el propósito principal de los episodios.	
Recursos		
Formularios		
Formulario 1 Formulario 5		
Procedimientos		
Determinar el propósito principal de los episodios Almacenar fecha y hora de inicio en el Formulario 5. Evaluar cada episodio en la Planilla de episodios y clasificarlo como EI o EO aplicando las reglas. Almacenar fecha y hora de finalización en el Formulario 5.		
Estándares		
Reglas		
<ul style="list-style-type: none"> <li><b>Para identificar EI</b></li> <li>R13. Los datos se reciben desde fuera del límite de la aplicación.</li> <li>R14. Mantiene uno o más recursos.</li> <li>R15. El proceso lógico es único entre los procesos lógicos realizados por otras EIs.</li> <li><b>Para identificar EO</b></li> <li>R16. La función envía datos fuera del límite de la aplicación.</li> <li>R17. El proceso lógico es único entre los procesos lógicos realizados por otras EOs.</li> <li>R18. La lógica del episodio recupera datos desde un recurso.</li> <li>R19. La lógica del episodio puede contener un cálculo o crear datos derivados.</li> <li>R20. La lógica del episodio no mantiene un recurso.</li> </ul>		
Normas para completar formularios		
Formulario 1 Formulario 5		

Criterio de salida	
<b>Estado o Condición</b>	
<i>Se clasificaron todos los episodios.</i>	
<b>Salida</b>	
<b>Nombre producto</b>	<b>Actividad destino</b>
<i>Lista de EIs y EOs</i>	<i>4 Determinar complejidad de los episodios</i>
<b>Herramientas</b>	
<i>Software de planilla de cálculo o software SFP Tool</i>	
<b>Mediciones</b>	
<b>Nombre variable</b>	<b>Descripción</b>
<i>E3</i>	<i>Esfuerzo (horas-persona) insumido por la actividad</i>
<b>Observaciones</b>	
<i>Una EI o EO que no cumple con la Regla R15 o R17 debe ser descartada de la lista de episodios.</i>	

ID	Nombre	
4	<i>Determinar complejidad de los episodios</i>	
<b>Propósito</b>		
<i>Asignar un nivel de complejidad a las EIs y EOs</i>		
<b>Rol</b>		
<b>Nombre</b>	<b>Responsabilidad</b>	<b>Entrenamiento</b>
<i>Analista de Sistemas /Ing. de software</i>	<ul style="list-style-type: none"> <li>• Ejecutar actividad</li> <li>• Revisar productos y procedimientos</li> </ul>	<ul style="list-style-type: none"> <li>• Manejo del procedimiento SFP</li> </ul>
<b>Criterio de entrada</b>		
<b>Estado o Condición</b>		
<i>Lista de EIs y EOs disponible. Herramienta de soporte disponible. Personal entrenado en SFP.</i>		
<b>Entrada</b>		
<b>Producto</b>	<b>Actividad fuente</b>	
<i>Lista de EIs y EOs</i>	<i>3 Clasificar episodios</i>	
<b>Actividades</b>		
<b>Paso</b>	<b>Nombre</b>	
1	<i>Contar DET y RTR de los episodios</i>	
2	<i>Determinar la complejidad de los episodios</i>	
<b>Recursos</b>		
<b>Documentos / Formularios</b>		
<i>Formulario 2 Formulario 4 Formulario 5</i>		
<b>Procedimientos</b>		
<u>Contar DET y RTR de los episodios</u> Almacenar fecha y hora de inicio en el Formulario 5. Incorporar cada EI y EO en el Formulario 2. Identificar y contar los DETs de cada EI y EO aplicando las reglas para EIs y EOs. Identificar y contar los RTRs de cada EI y EO aplicando las reglas para EIs y EOs. Almacenar fecha y hora de finalización en el Formulario 5.		
<u>Determinar la complejidad de los episodios</u> Almacenar fecha y hora de inicio en el Formulario 5. Usando el número de DETs y RTRs determinar la complejidad de cada EI y EO aplicando las tablas para EI y EQ. Contar las EI y EO por nivel de complejidad y almacenar en el Formulario 4.		

Almacenar fecha y hora de finalización en el Formulario 5.							
<b>Estándares</b>							
<b>Reglas</b>							
<ul style="list-style-type: none"> <li>• <b>Para contar DET en EI</b></li> </ul> <p>R23. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> no repetido que entra o sale del límite y es requerido para completar la EI.</p> <p>R24. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> no repetido usado en la definición de un <i>Símbolo Complejo</i> (no identificado como <i>recurso</i>) que entra o sale del límite y es requerido para completar la EI.</p> <p>R25. No contar datos recuperados o derivados por el sistema y almacenados en un <i>recurso</i> durante un <i>episodio simple</i> si los datos no cruzaron el límite.</p> <ul style="list-style-type: none"> <li>• <b>Para contar DET en EO</b></li> </ul> <p>R26. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> que entra a la aplicación y es requerido para especificar cuándo, qué y cómo los datos son recuperados o generados por el <i>episodio simple</i>.</p> <p>R27. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> que sale del límite de la aplicación.</p> <p>R28. Si un DET entra y sale se cuenta una vez.</p> <p>R29. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> usado en la definición de un <i>Símbolo Complejo</i> (no identificado como <i>recurso</i>) que sale del límite de la aplicación.</p> <p>R30. No contar datos recuperados o derivados por el sistema durante un <i>episodio simple</i> si los datos no cruzaron el límite.</p> <ul style="list-style-type: none"> <li>• <b>Para contar DET común para EI y EO</b></li> </ul> <p>R31. No contar como DET a un <i>Símbolo</i> que representa un <i>actor</i>.</p> <ul style="list-style-type: none"> <li>• <b>Para contar RTR en EI</b></li> </ul> <p>R32. Contar 1 RTR por cada <i>recurso Símbolo Complejo</i> o <i>No-Símbolo</i> mantenido, leído o mantenido y leído por la EI.</p> <ul style="list-style-type: none"> <li>• <b>Para contar RTR en EO</b></li> </ul> <p>R33. Contar 1 RTR por cada <i>recurso Símbolo Complejo</i> o <i>No-Símbolo</i> leído por la EO.</p>							
<b>Tablas</b>							
<b>Complejidad para EI</b>		<b>Complejidad para EO</b>					
<b>RTR</b>	<b>DET</b>			<b>RTR</b>	<b>DET</b>		
	1 - 4	5 - 15	16 o más		1 - 5	6 - 19	20 o más
1	Baja	Baja	Media	1	Baja	Baja	Media
2	Baja	Media	Alta	2 o 3	Baja	Media	Alta
3 o más	Media	Alta	Alta	4 o más	Media	Alta	Alta
<b>Normas para completar formularios</b>							
<i>Formulario 2</i>							
<i>Formulario 4</i>							
<i>Formulario 5</i>							
<b>Criterio de salida</b>							
<b>Estado o Condición</b>							
<i>Se determinó la complejidad de todas las EIs y EOs.</i>							
<b>Salida</b>							
<b>Nombre producto</b>		<b>Actividad destino</b>					
<i>Número de EIs por cada nivel de complejidad</i>		<i>7 Calcular el tamaño funcional</i>					
<i>Número de EOs por cada nivel de complejidad</i>							
<b>Herramientas</b>							
<i>Software de planilla de cálculo o software SFP Tool</i>							

Mediciones	
Nombre variable	Descripción
E4	Esfuerzo (horas-persona) insumido por las actividades
Observaciones	

ID	Nombre	
5	Identificar recursos	
Propósito		
Obtener una lista que contenga los recursos usados por los episodios excluyendo las repeticiones		
Rol		
Nombre	Responsabilidad	Entrenamiento
Analista de Sistemas /Ing. de software	<ul style="list-style-type: none"> <li>Ejecutar actividad</li> <li>Revisar productos y procedimientos</li> </ul>	<ul style="list-style-type: none"> <li>Manejo del procedimiento SFP</li> </ul>
Criterio de entrada		
Estado o Condición		
Lista de episodios disponible. Herramienta de soporte disponible. Personal entrenado en SFP.		
Entrada		
Producto	Actividad fuente	
Lista de episodios	2 Identificar episodios	
Actividades		
Paso	Nombre	
1	Analizar recursos de los episodios	
Recursos		
Formularios		
Formulario 1 Formulario 3 Formulario 5		
Procedimientos		
Analizar recursos de los episodios Almacenar fecha y hora de inicio en el Formulario 5. Evaluar el/los recursos de cada episodio no descartado en la Planilla de episodios aplicando las reglas y agregarlo/s en el Formulario 3 si no está incluido. Almacenar fecha y hora de finalización en el Formulario 5.		
Estándares		
Reglas		
R22. Un recurso se considera una sola vez aunque sea referenciado en múltiples episodios simples de uno o más escenarios.		
Normas para completar formularios		
Formulario 3 Formulario 5		
Criterio de salida		
Estado o Condición		
Se evaluaron todos los episodios. No existen recursos repetidos en la lista de recursos.		
Salida		
Nombre producto	Actividad destino	
Lista de recursos	6 Determinar complejidad y contribución de los recursos	



<b>Herramientas</b>	
<i>Software de planilla de cálculo o software SFP Tool</i>	
<b>Mediciones</b>	
<b>Nombre variable</b>	<b>Descripción</b>
<i>E5</i>	<i>Esfuerzo (horas-persona) insumido por la actividad</i>
<b>Observaciones</b>	

ID	Nombre	
6	<i>Determinar complejidad de los recursos</i>	
<b>Propósito</b>		
<i>Asignar un nivel de complejidad a cada recurso</i>		
<b>Rol</b>		
<b>Nombre</b>	<b>Responsabilidad</b>	<b>Entrenamiento</b>
<i>Analista de Sistemas /Ing. de software</i>	<ul style="list-style-type: none"> <li>• <i>Ejecutar actividad</i></li> <li>• <i>Revisar productos y procedimientos</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Manejo del procedimiento SFP</i></li> </ul>
<b>Criterio de entrada</b>		
<b>Estado o Condición</b>		
<i>Lista de episodios disponible y sus episodios clasificados como EI y EO. Herramienta de soporte disponible. Personal entrenado en SFP.</i>		
<b>Entrada</b>		
<b>Producto</b>	<b>Actividad fuente</b>	
<i>Lista de recursos</i>	<i>5 Identificar recursos</i>	
<b>Actividades</b>		
<b>Paso</b>	<b>Nombre</b>	
1	<i>Contar DET en los recursos</i>	
2	<i>Determinar la complejidad de los recursos</i>	
<b>Recursos</b>		
<b>Formularios</b>		
<i>Formulario 3 Formulario 5</i>		
<b>Procedimientos</b>		
<p><u>Contar DET en los recursos</u>                  Almacenar fecha y hora de inicio en el Formulario 5.                  Identificar y contar los DETs de cada recurso en el Formulario 3 aplicando las reglas para recursos.                  Aplicar la regla para RET a cada recurso en el Formulario 3.</p> <p><u>Determinar la complejidad de los recursos</u>                  Determinar la complejidad aplicando las tablas para EI y EQ.                  Contar los recursos por nivel de complejidad y almacenar en el Formulario 4.                  Almacenar fecha y hora de finalización en el Formulario 5.</p>		
<b>Estándares</b>		
<b>Reglas</b>		
<ul style="list-style-type: none"> <li>• <b>Para contar DET en recursos</b>                      R34. Si el <i>recurso</i> es <i>No-Símbolo</i> no hay DET.                      R35. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> mantenido en o recuperado desde un <i>recurso</i> mediante la ejecución de un <i>episodio simple</i>.                      R36. Contar 1 DET por cada <i>Símbolo Simple</i> o <i>No-Símbolo</i> usado en la definición del <i>Símbolo Complejo</i> identificado como <i>recurso</i>.</li> <li>• <b>Para contar RET en recursos</b>                      R37. Contar 1 RET por <i>recurso</i>.</li> <li>• <b>Regla para valorar la complejidad de un recurso No-Símbolo</b>                      R38. Un <i>recurso</i> de tipo <i>No-Símbolo</i> siempre tiene complejidad <i>Baja</i>.</li> </ul>		

<b>Tabla</b>			
<b>Complejidad para recursos</b>			
<b>RET</b>	<b>DET</b>		
	1 a 19	20 - 50	51 o más
1	Baja	Baja	Media
<b>Normas para completar formularios</b>			
<i>Formulario 3</i>			
<i>Formulario 4</i>			
<i>Formulario 5</i>			
<b>Criterio de salida</b>			
<b>Estado o Condición</b>			
<i>Se determinó la complejidad de todos los recursos.</i>			
<b>Salida</b>			
<b>Nombre producto</b>		<b>Actividad destino</b>	
<i>Número de recursos por cada nivel de complejidad</i>		<i>7 Calcular el tamaño funcional</i>	
<b>Herramientas</b>			
<i>Software de planilla de cálculo o software SFP Tool</i>			
<b>Mediciones</b>			
<b>Nombre variable</b>	<b>Descripción</b>		
<i>E6</i>	<i>Esfuerzo (horas-persona) insumido por las actividades</i>		
<b>Observaciones</b>			

<b>ID</b>	<b>Nombre</b>		
<i>7</i>	<i>Calcular el tamaño funcional</i>		
<b>Propósito</b>			
<i>Obtener la cuenta total de FPs</i>			
<b>Rol</b>			
<b>Nombre</b>	<b>Responsabilidad</b>	<b>Entrenamiento</b>	
<i>Analista de Sistemas /Ing. de software</i>	<ul style="list-style-type: none"> <li>• Ejecutar actividad</li> <li>• Revisar productos y procedimientos</li> </ul>	<ul style="list-style-type: none"> <li>• Manejo del procedimiento SFP</li> </ul>	
<b>Criterio de entrada</b>			
<b>Estado o Condición</b>			
<i>Se obtuvo el número de EIs, EOs y recursos por cada nivel de complejidad.</i>			
<i>Herramienta de soporte disponible.</i>			
<i>Personal entrenado en SFP.</i>			
<b>Entrada</b>			
<b>Producto</b>		<b>Actividad fuente</b>	
<i>Número de EIs por cada nivel de complejidad</i>		<i>4 Determinar complejidad de los episodios</i>	
<i>Número de EOs por cada nivel de complejidad</i>		<i>6 Determinar complejidad de los recursos</i>	
<i>Número de recursos por cada nivel de complejidad</i>			
<b>Actividades</b>			
<b>Paso</b>	<b>Nombre</b>		
<i>1</i>	<i>Determinar la contribución en FP de los episodios</i>		
<i>2</i>	<i>Determinar la contribución en FP de los recursos</i>		
<i>3</i>	<i>Calcular la cuenta de FPs</i>		
<b>Recursos</b>			
<b>Documentos / Formularios</b>			
<i>Formulario 4</i>			
<i>Formulario 5</i>			

<b>Procedimientos</b>	
<u>Determinar la contribución en FP de los episodios</u> Almacenar fecha y hora de inicio en el Formulario 5. Multiplicar el número de EIs por cada nivel de complejidad por su contribución en FPs obtenida desde las Tablas para EI y almacenar en el Formulario 4. Multiplicar el número de EOs por cada nivel de complejidad por su contribución en FPs obtenida desde las Tablas para EO y almacenar en el Formulario 4.	
<u>Determinar la contribución en FP de los recursos</u> Multiplicar el número de recursos por cada nivel de complejidad por su contribución en FPs obtenida desde las Tablas para recursos y almacenar en el Formulario 4. Almacenar fecha y hora de finalización en el Formulario 5.	
<u>Calcular la cuenta de FPs</u> Sumar los productos obtenidos en los pasos anteriores.	
<b>Estándares</b>	
<b>Tablas</b>	
<b>Contribución para EI y EO</b>	<b>Contribución para recursos</b>
Baja (3)   Media (4)   Alta (6)	Baja (7)   Media (10)
<b>Normas para completar formularios</b> Formulario 4 Formulario 5	
<b>Criterio de salida</b>	
<b>Estado o Condición</b>	
<i>Se evaluó la contribución de las EIs, EOs y los recursos.</i>	
<b>Salida</b>	
<b>Nombre producto</b>	<b>Actividad destino</b>
<i>Cuenta total de FPs</i>	
<b>Herramientas</b>	
<i>Software de planilla de cálculo o software SFP Tool</i>	
<b>Mediciones</b>	
<b>Nombre variable</b>	<b>Descripción</b>
<i>E</i>	<i>Esfuerzo total (horas-persona) insumido por el proceso. <math>E = E1 + E2 + E3 + E4 + E5 + E6</math></i>
<b>Observaciones</b>	

### 5.3. Medición del esfuerzo

La unidad de medida es la *hora de trabajo (work-hour)* definida como la hora trabajada por una persona, incluyendo breves interrupciones, pero excluyendo las más extensas. Debido a que el objetivo es medir las horas trabajadas netas y a la disparidad de criterios que puede existir en cuanto a las interrupciones, se consideró más adecuado registrar todas las entradas correspondientes a la hora de inicio y fin de un período de trabajo y asociarlas con el respectivo paso del proceso. Pueden mencionarse las siguientes ventajas:

- Favorecer el control y la gestión eficiente de recursos.
- Evaluar estrategias de mejoramiento.
- Establecer una base para la estimación de futuros proyectos.

### 5.4. Documentación de soporte

Es fundamental que el proceso de estimación esté documentado porque:

- facilita y organiza la recolección de datos y realización de cálculos.

- favorece el proceso de revisión y el control de los cambios.
- permite el seguimiento de los FP (hacia sus orígenes).

En consecuencia, es conveniente disponer de formularios estándar que permitan mantener los resultados de las actividades del proceso y adjuntarlos a la documentación del proyecto. Para registrar los datos y reportar los resultados (subpaso 1 del Paso 3 del Modelo de Jacquet y Abran (1997), Capítulo 2, sección 2.2.7.3) de cada actividad fue diseñado un juego de formularios que se puede implementar usando un software estándar de planilla de cálculo. Los formularios propuestos (en el Apéndice C se presentan las plantillas de los formularios y las normas para completarlos) son los siguientes:

- **Formulario 1:** *Planilla de episodios*  
Objetivo: almacenar y clasificar los episodios.  
Uso: durante los pasos 1, 2, 3 y 5 del proceso.
- **Formulario 2:** *Complejidad de los episodios*  
Objetivo: almacenar los DETs y RTRs y el nivel de complejidad de las EIs y EOs.  
Uso: durante el paso 4 del proceso.
- **Formulario 3:** *Complejidad de los recursos*  
Objetivo: almacenar los DETs y RET y el nivel de complejidad de los recursos.  
Uso: durante los pasos 5 y 6 del proceso.
- **Formulario 4:** *Cálculo de FP no Ajustados*  
Objetivo: almacenar el número de EIs, EOs y recursos por nivel de complejidad y la cuenta de FPs.  
Uso: durante los pasos 4, 6 y 7 del proceso.
- **Formulario 5.** *Horas de trabajo*  
Objetivo: almacenar los datos de esfuerzo por cada paso del proceso.  
Uso: durante todos los pasos del proceso.

### 5.5. Soporte automatizado

En el marco del Proyecto de Investigación “Estimaciones de Tamaño en la Etapa Inicial de un Proyecto de Desarrollo de Software”, realizado en la Universidad Nacional de la Patagonia San Juan Bosco, fue desarrollado el software “SFP Tool” [Ritter *et al.* 2009] que soporta el proceso SFP. Básicamente esta herramienta automatiza los pasos del proceso que no requieren la interacción con el usuario (pasos 1, 5 y 7) y facilita aquéllos en los cuales es necesaria su intervención (pasos 2, 3, 4 y 6). Además, maneja automáticamente el registro de tiempos y el cálculo del esfuerzo. La descripción completa de la herramienta se presenta en el Apéndice E.

Un aspecto ya señalado, pero que vale destacar en este contexto, es el referido a la necesidad de la revisión de los escenarios previo a su uso como fuente de información para la herramienta. La experiencia ha demostrado que algunos errores u omisiones en la redacción de los escenarios conducen a resultados incorrectos cuando es usada la herramienta.

### 5.6. Resumen

La especificación del proceso, aquí presentada, provee un documento estructurado, conciso y preciso cuya utilidad fue comprobada en la práctica, tanto en la fase de

capacitación del usuario como durante la ejecución de una estimación, en este último caso facilitando la organización del trabajo y ayudando a despejar dudas.

Una cuestión clave para la validación del enfoque es disponer de datos de mediciones de proyectos. La factibilidad de esa práctica se relaciona íntimamente con las facilidades provistas a los usuarios para almacenar los resultados de la estimación de FPs. Los formularios propuestos permiten la recolección de los resultados intermedios y finales de la estimación y del esfuerzo insumido por el proceso. Además, como es mantenida toda la información generada durante el proceso, también se facilita la revisión y seguimiento de una estimación desde el principio hasta el final. Estas ventajas son acrecentadas si además es usada la herramienta propuesta.



## Capítulo 6

### Evaluación del Procedimiento SFP

En este capítulo se presenta la evaluación del procedimiento SFP. Esta evaluación comprende dos partes: en primer lugar se aplica el procedimiento SFP para comprobar la factibilidad de su uso en la práctica. En segundo lugar se realiza una evaluación basada en múltiples casos de estudio enmarcados en un Caso de Estudio global siguiendo la propuesta de Yin (2003). El Caso de Estudio es usado como una “prueba de conceptos”, lo que significa que los resultados y las conclusiones no pretenden demostrar la validez definitiva de SFP, pues para esto se requiere ampliar la experimentación sobre un número mayor de casos y sobre sistemas de mayor envergadura.

#### 6.1. Evaluación de la aplicabilidad del procedimiento SFP

En esta sección se presenta la estimación del tamaño funcional de los escenarios de un grupo de sistemas no implementados, de los que sólo se dispone del LEL y Escenarios (L&E). Básicamente se persiguen los siguientes objetivos:

- comprobar la comprensibilidad y pertinencia de las reglas definidas y la coherencia de la secuencia de los pasos del proceso SFP.
- determinar la magnitud del esfuerzo requerido para ejecutar la estimación.

Vale destacar que la experiencia aquí presentada no cubre los requisitos ni tiene el objetivo de una validación del procedimiento.

Los casos considerados son los siguientes: *Plan de Ahorro* [Mauco *et al.* 1997]; *Recepción del Hotel* [Bertolami y Centeno 2001]; *Pasaporte* [Leite *et al.* 1996]; *Crédito hipotecario* [s.a. 1997]; *Agenda de Reuniones* [Hadad *et al.* 1998]; *Biblioteca* [Melo *et al.* s.f.]; *Sistema de Alumnos* [Ballesteros *et al.* 1997]; *Banco de Sangre* [Gil y Arias 2000]; *Estación de Servicio* [Esteban y Heidanowski 1998]; *Notificaciones* [Oriana *et al.* 2006]; *L&E del L&E* [García y Gentile 2000]. En la Tabla 6.1 se presentan los resultados de la estimación realizada manualmente.

Tabla 6.1. Resumen de los resultados de la estimación con SFP

Caso <sup>1</sup>	EI		EO		Recurso		FP <sup>4</sup>	Esfuerzo (hora – persona)	Esfuerzo / FP
	Nro. <sup>2</sup>	Comp. <sup>3</sup>	Nro. <sup>2</sup>	Comp. <sup>3</sup>	Nro.	Comp.			
Plan de Ahorro (3)	3 / 1	B / M	6	B	5	B	66	02:03	0,03
Recepción del Hotel (1)	9 / 1	B / M	3	B	9	B	103	02:50	0,03
Pasaporte (5)	9	B	9	B	7	B	103	02:06	0,02
Crédito hipotecario (11)	6	B	5	B	11	B	103	02:14	0,02
Agenda de Reuniones (2)	17 / 2	B / M	5	B	7	B	123	03:53	0,03
Biblioteca (8)	11	B	20	B	5	B	128	04:29	0,04
Sistema de Alumnos (7)	12 / 1	B / M	11	B	10	B	143	03:48	0,03
Banco de Sangre (6)	24	B	6	B	10	B	160	03:58	0,02
Estación de Servicio (4)	14	B	16	B	18	B	216	05:33	0,03
Notificaciones (9)	17 / 7	B / M	7/1	B / M	25	B	279	06:39	0,02
L&E del L&E (10)	36	B	25	B	21	B	330	09:41	0,03

<sup>1</sup>entre paréntesis se indica el orden de realización; <sup>2</sup>Nro. indica el número de Entradas Externas (EI) o Salidas Externas (EO) por cada nivel de complejidad; <sup>3</sup>Comp. B y M significan

Complejidad baja y media respectivamente. Por ejemplo, el *Plan de Ahorro* tiene 3 EIs de complejidad baja y 1 de complejidad media.<sup>41</sup>Puntos Función (FP).

A partir de la aplicación de SFP sobre estos casos se comprobó que las reglas propuestas junto con el proceso definido permiten la cuantificación del tamaño funcional de los escenarios. Cabe mencionar que, además, la realización de esta experiencia permitió adquirir destreza en la aplicación de la técnica.

Por otra parte, como puede observarse en la Tabla 6.1, el esfuerzo requerido para ejecutar el proceso está comprendido entre 2:03 y 9:41 hora-persona, lo que significa que, *a priori* (en la sección 6.2.7 es comparado con los valores de una medición estándar), parece ser irrelevante frente a las ventajas de disponer una estimación tan temprana. Más importante aún, el valor de la relación Esfuerzo/FP varía entre 0,02 y 0,04 hora-persona por FP estimado. El 55% de los casos insumió 0,03 hora-persona/FP. El valor 0,04 hora-persona/FP corresponde al L&E del caso *Biblioteca* que está escrito en portugués, lo que podría justificar el incremento debido a la dificultad para comprender un léxico en un lenguaje que no es el nativo del autor del proceso. Otro aspecto que se puede observar es que para las cuatro primeras aplicaciones (ver el orden de realización adjunto al nombre de cada caso en la columna 1 de la Tabla 6.1), el esfuerzo se mantuvo constante en 0,03 hora-persona/FP, pero a medida que se fue adquiriendo experiencia se produjo una reducción, de modo que para los siete casos restantes osciló entre 0,02 hora-persona/FP (4 casos) y 0,03 hora-persona/FP (2 casos), con la mencionada excepción del caso *Biblioteca*. Vale destacar que el software SFP Tool permitirá agilizar el proceso de estimación de los FPs. Actualmente no se dispone de datos definitivos al respecto, pues el uso de la herramienta también requiere entrenamiento y por lo tanto, en las primeras aplicaciones, el esfuerzo propio del proceso se ha visto incrementado por el inherente al aprendizaje.

Por último, otro detalle que puede observarse en los resultados intermedios que se presentan en la Tabla 6.1, es que la complejidad del 93% de las EIs, 99% de las EOs y 100% de los recursos es baja (considerando el promedio para todos los casos). Esto puede ser explicado porque la información disponible en la etapa de Elicitación de Requerimientos muy probablemente sea insuficiente para describir las entidades del modelo. Por lo tanto, estos resultados deberían ser ratificados o rectificadas mediante mediciones realizadas sobre los artefactos que sean producidos en las siguientes fases del desarrollo.

## 6.2. Evaluación de la exactitud de la estimación de tamaño funcional

### 6.2.1. Objetivo

El objetivo de esta evaluación es determinar la capacidad de SFP para estimar los FP<sup>41</sup> de los escenarios a los fines de dar respuesta a dos de las preguntas formuladas en el Capítulo 1 de esta tesis: ¿la estimación del tamaño funcional basada en escenarios es consistente con las que se obtienen en las etapas posteriores del proceso?, ¿es comparable la exactitud de la estimación de tamaño funcional basada en escenarios con la de otras propuestas?

La primera de ellas apunta a verificar la proximidad entre el tamaño estimado y el resultado de una medición realizada sobre un producto de una etapa posterior, por ejemplo, del software en funcionamiento. Como fue explicado en el Capítulo 3, sección 3.1, esto significa calcular la exactitud de la estimación, para lo cual el *valor de*

---

<sup>41</sup> Todas las referencias a FP corresponden a FP no Ajustados (UFP).



*referencia*, en el contexto de estudio, es el resultado de la medición con el método estándar IFPUG. La segunda se orienta a comprobar si la exactitud lograda con SFP es similar a la de otras propuestas de estimación. Es decir, en ambos casos la cuestión central es la exactitud de la estimación. Esta evaluación se corresponde con el subpaso 2 del Paso 3 del Modelo de Proceso [Jacquet y Abran 1997] (Capítulo 2, sección 2.2.7.3).

### 6.2.2. Metodología de investigación

Existen diferentes formas de hacer investigación empírica en la Ingeniería del Software, entre las principales pueden mencionarse: encuestas, experimentos formales y casos de estudio [Fenton y Pfleeger 1996]. Los diversos métodos de investigación tienen ventajas y desventajas según el tipo de problema abordado y sus circunstancias. A diferencia de las encuestas, que son estudios retrospectivos, los casos de estudio y los experimentos enfocan sobre eventos contemporáneos. La diferencia clave entre ellos radica en el nivel de control sobre los factores externos que pueden afectar los resultados. Kitchenham *et al.* (1994) definen al caso de estudio, en el contexto de la Ingeniería de Software, como una prueba del uso de un método/herramienta con el objeto de comparar el efecto de una herramienta con el de otras herramientas o prácticas actuales. El control de factores externos es mucho más difícil que en un experimento formal.

En el marco de esta evaluación es estudiado un evento contemporáneo, como lo es la aplicación de una nueva técnica de estimación de tamaño funcional y la comparación con un método estándar. Los posibles factores externos son: la destreza para desarrollar el L&E, el criterio (o el estilo) usado para las descripciones del L&E (no hay un riguroso patrón para generarlos, sólo las heurísticas propuestas para inspeccionarlos [Kaplan *et al.* 2000], [Doorn *et al.* 1998]) y la experiencia en la aplicación de ambas técnicas (SFP, IFPUG), en este caso los dos primeros no son controlados.

Existen dos tipos básicos de diseños para los casos de estudio: simple (un caso) y múltiple (varios casos), los cuales a su vez pueden ser realizados sobre una sola unidad de análisis o más de una unidad [Yin 2003].

Para llevar a cabo la evaluación fue elegido un Caso de Estudio múltiple - de acuerdo a la clasificación propuesta por Yin (2003) - como estrategia de investigación. Diferentes razones determinaron esta elección, por un lado, un caso de estudio simple es apropiado en determinadas circunstancias, como por ejemplo, cuando se trata de un caso típico (o representativo), crítico, extremo o revelador, lo cual no podía ser asegurado para los casos disponibles; por otro lado, el caso de estudio múltiple tiene la ventaja que la evidencia suele ser más convincente y por ende, el estudio es considerado más robusto. Estos casos de estudio se basan en la lógica de la replicación (es decir, la repetición del mismo estudio sobre diferentes casos) y no en la del muestreo estadístico. Esto significa que los casos deben ser seleccionados de modo que produzcan resultados similares (*replicación literal*) o resultados opuestos por razones predecibles (*replicación teórica*) [Yin 2003].

La selección de los casos estuvo determinada, en gran parte, por su disponibilidad y su tamaño. El Caso de Estudio fue ejecutado siguiendo el esquema propuesto por Yin (2003) para el Caso de Estudio múltiple (Figura 6.1) y se espera obtener resultados similares (replicación literal). La siguiente cuestión en el diseño de un caso de estudio es el número de casos necesarios o suficientes para el estudio. El número de replicaciones literales depende del grado de certeza deseado para los resultados del caso de estudio. Dos o tres replicaciones literales pueden ser suficientes cuando las hipótesis rivales son muy diferentes y el problema no requiere un excesivo grado de certeza. En caso contrario, pueden ser realizadas cinco, seis o más replicaciones [Yin 2003].

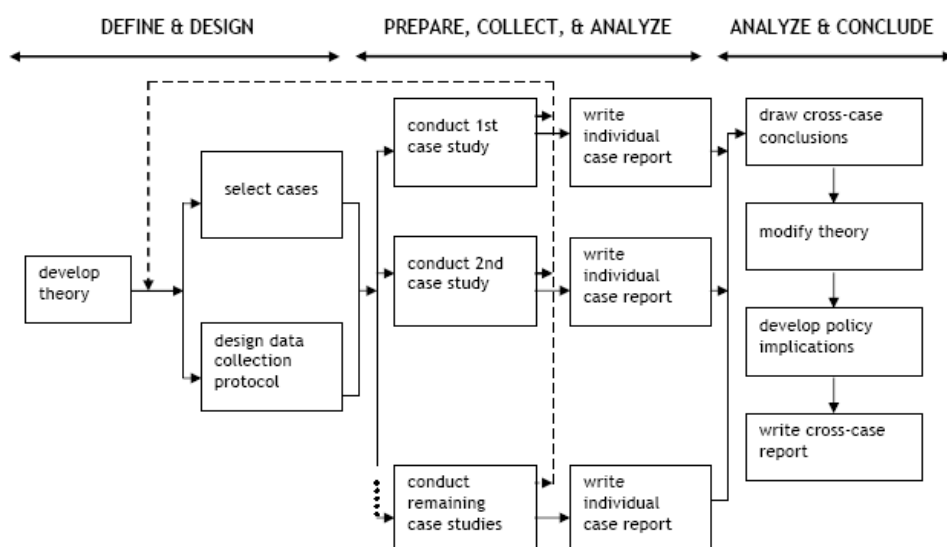


Figura 6.1. Método de Casos de Estudio [Yin 2003]

### 6.2.3. Selección de los casos

Para abordar la evaluación se adoptó la siguiente estrategia: por un lado, se seleccionaron sistemas implementados que no contaban con el L&E y por el otro, se elaboró el L&E de un sistema que posteriormente fue implementado.

#### i. Sistemas implementados:

##### a. *Sistema de Gestión de Programas de asignaturas.*

Artefactos disponibles: el software bajo estudio y el código fuente.

Fuentes de información: el programador, el administrador del sistema, el manual del usuario y los usuarios.

##### b. *Sistema de Gestión de un Comercio de indumentaria.*

Artefactos disponibles: el software bajo estudio, los Casos de Uso y el diagrama de clases.

Fuentes de información: el manual del usuario.

i.1. Construir los escenarios a partir de una aplicación implementada y aplicarlo a los sistemas mencionados en los puntos a y b, para lo que fue necesario diseñar un proceso para generar los escenarios en dichas condiciones.

i.2. Medir los IFPUG FP a partir de una aplicación implementada que no cuenta con la documentación de requerimientos tradicional y aplicarlo a los sistemas referidos en los puntos a y b, para lo que fue necesario elaborar una estrategia para medir los FP en dichas condiciones.

#### ii. Nuevo sistema implementado partiendo del L&E:

##### c. *Sistema SFP Tool.*

Artefactos disponibles: el software bajo estudio, el L&E, la Especificación de Requerimientos del Software (SRS – Software Requirement Specification) y el código fuente.

Fuentes de información: el programador, el manual del usuario y los usuarios.

- ii.1. Generar el L&E durante la fase de Elicitación de los Requerimientos del software y luego desarrollar el software siguiendo las etapas clásicas del modelo en cascada de un proyecto de desarrollo de software [Pressman 2001].

**6.2.4. Descripción del proceso de evaluación**

De acuerdo al estándar ISO 14143-4 [ISO 2002b], la conformidad con el estándar ISO 14143-1 [ISO 1998] es condición necesaria pero no suficiente para determinar si un Método de Medición del Tamaño Funcional (FSMM – Functional Size Measurement Method) es adecuado para cuantificar el tamaño funcional del software. Un proceso de evaluación de un FSMM deberá tener en cuenta las evidencias prácticas del desempeño de dicho método. Esto puede ser realizado a través de una evaluación comparativa (*benchmarking*) (Figura 6.2) de los resultados del FSMM en estudio con los obtenidos por un FSMM de referencia, ambos aplicados sobre una colección de Requerimientos de Usuario de Referencia (RUR – Reference User Requirements) conocidos.

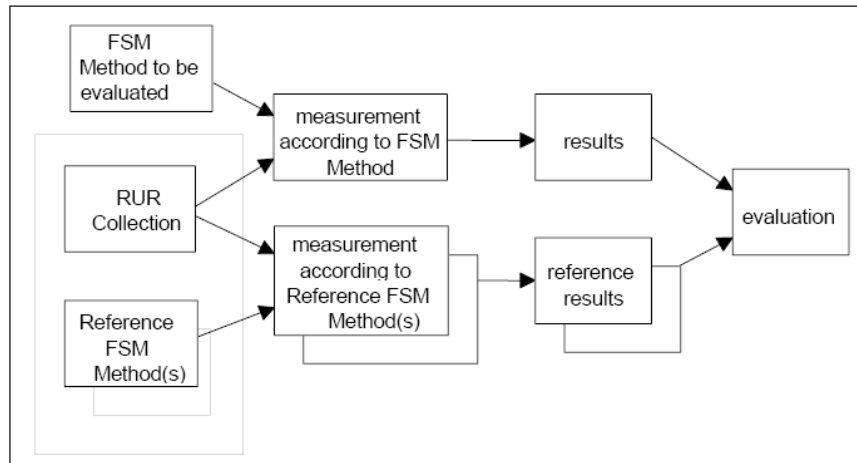


Figura 6.2. Proceso de evaluación de un FSMM [ISO 2002b]

El proceso aplicado para la evaluación de SFP fue definido a partir de una adaptación (Figura 6.3) del modelo presentado en la Figura 6.2.

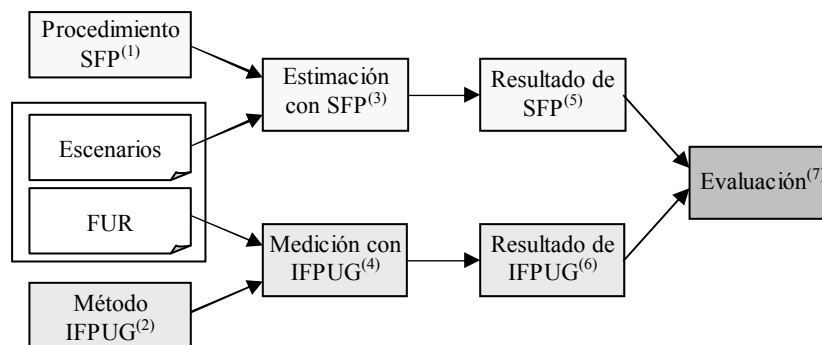


Figura 6.3. Proceso de evaluación de SFP (adaptado de [ISO 2002b])

- (<sup>1</sup>) Método a evaluar.
- (<sup>2</sup>) Método de referencia.
- (<sup>3</sup>) La estimación con SFP es aplicada sobre los escenarios.
- (<sup>4</sup>) La medición con el método IFPUG es aplicada sobre los Requerimientos Funcionales del Usuario.
- (<sup>5</sup>) Resultado de la estimación con SFP.
- (<sup>6</sup>) Resultado de referencia.
- (<sup>7</sup>) Comparación de los resultados.

En líneas generales el proceso aplicado consiste de las siguientes actividades:

- Estimar el tamaño funcional de los escenarios aplicando el procedimiento SFP. El resultado de la estimación representa el *valor estimado* (Capítulo 3, sección 3.1) para usar en el cálculo del error relativo de la estimación.
- Medir el tamaño funcional de la aplicación con el método IFPUG. El resultado de la medición es el valor de referencia para el usar en el cálculo del error relativo de la estimación.
- Calcular el error relativo de la estimación, el cual es usado como indicador de la exactitud de la estimación y en consecuencia permite determinar la capacidad de SFP como herramienta de estimación de los FP.

### 6.2.5. Caso de Estudio

Como fue mencionado, el Caso de Estudio involucra a los tres casos presentados en la sección 6.2.3:

- *Sistema de Gestión de Programas de asignaturas (A<sup>42</sup>)*
- *Sistema de Gestión de un Comercio de indumentaria (B)*
- *Sistema SFP Tool (C)*

La realización del Caso de Estudio exigió resolver dos cuestiones preliminares. En primer lugar, como los escenarios son el artefacto de entrada para el proceso SFP y no estaban disponibles para los sistemas A y B, fue necesario elaborar un proceso (sección 6.2.5.1) para construirlos a partir de la aplicación. En segundo lugar, la relevancia de la medición basada en el método IFPUG (por ser el valor de referencia) determinó la necesidad de investigar la validez de esta medición; esto fue realizado aplicando el proceso de validación de Morris y Desharnais (2001) que es resumido en la sección 6.2.5.2.

#### 6.2.5.1. Construcción del L&E desde una aplicación implementada

Debido a que se carece de un proceso definido para construir los escenarios de un sistema implementado y sin documentación, fue definida una estrategia<sup>43</sup> para construirlo. Concretamente, el proceso consta de los siguientes pasos y actividades:

1. ***Establecer el Universo de Discurso (UD)***<sup>44</sup>. A partir del análisis del contexto en que se desarrollaría el proceso fueron identificados los recursos y fuentes de información disponibles.

---

<sup>42</sup> De aquí en adelante los sistemas bajo estudio son referidos como A, B y C.

<sup>43</sup> La definición del proceso L&E fue realizada por Elena Centeno y el autor de esta tesis.

2. **Conocer las características del software.** Cuando fue posible se entrevistó al desarrollador y al administrador del software. Estas entrevistas permitieron entender el funcionamiento del sistema. A medida que surgieron dudas se hicieron nuevas consultas. En una primera etapa se registraron las palabras o frases usadas con más frecuencia o que tienen un significado relevante para la aplicación y se generó una lista de símbolos candidatos del LEL.
3. **Conocer la funcionalidad del software.** Con este fin fueron usadas todas las funciones disponibles en los diferentes modos - en el caso que existiesen - (por ejemplo, administrador / cliente), incluyendo la ayuda y fue consultado el manual del usuario. Estos últimos permitieron aclarar aspectos específicos acerca del uso y configuración del software, el llenado de los formularios y la generación de los planillas/listados requeridos por el cliente.

En este paso se amplió/modificó la lista inicial de símbolos candidatos del LEL.

4. **Generar el LEL.** La lista de símbolos del LEL fue depurada hasta obtener la lista definitiva. Luego los símbolos fueron clasificados y desarrollada la descripción de la noción e impacto de cada símbolo. Para esto fueron necesarias sendas iteraciones a través de los pasos 2 y 3 para completar o verificar dichas descripciones.
5. **Generar los escenarios.** Fue aplicada la heurística propuesta por Leite *et al.* (2000) para describir los escenarios a partir de los símbolos del LEL. Durante este paso fue necesario repetir las actividades del paso 2 y 3 para completar o verificar la descripción de los escenarios y obtener la lista de escenarios definitiva.

#### 6.2.5.2. Validación del resultado de la medición con el método IFPUG

Morris y Desharnais (2001) describen un método para verificar la validez de los datos recolectados durante una medición de Puntos Función no Ajustados (UFP – Unadjusted Function Points) aplicando las reglas de IFPUG Function Point Counting Practices Manual (CPM) Release 4.0. Si bien en el contexto de esta tesis la medición fue realizada aplicando CPM 4.1.1, se consideró adecuado usarlo para la revisión debido a que “el estudio de impacto realizado para comparar las medidas resultantes de aplicar el CPM 4.0 y el 4.1 mostró mínimas diferencias en el tamaño funcional de los proyectos medidos con ambos métodos. Los resultados de la medición indicaron que las medidas son equivalentes y que, en la mayoría de los casos, es innecesario un ajuste de las medidas realizadas previamente con el 4.0” [IFPUG 2000]. Los autores desarrollaron el método de validación sobre la base de sus experiencias en la realización del Análisis de Puntos Función (FPA – Function Points Analysis) y en la evaluación de los resultados de más de 300 aplicaciones.

El proceso consta de tres niveles de revisión y en cada nivel se examinan las propiedades de la cuenta de FP y/o los atributos del software medido.

- la revisión de *alto* nivel analiza:
  - o FP del software,
  - o Alcance de la medición de FP,
  - o límite del software medido.
- la revisión de nivel *intermedio* examina:

---

<sup>44</sup> Es el contexto en el que se desarrollarán las tareas de ingeniería, recursos disponibles, objetivos del producto y sus límites. En el UD residen todas las fuentes de información a ser utilizadas [Leite *et al.* 2000].

- la relación entre el número de archivos lógicos y los UFP del software,
  - la relación entre los números de funciones entre sí unas con otras,
  - los porcentajes de contribución al tamaño funcional de cada tipo de función,
  - la complejidad de las funciones.
- la revisión de *bajo* nivel analiza:
- la cuenta de FP de las funciones de datos seleccionadas<sup>1</sup>,
  - la cuenta de FP de las funciones transaccionales seleccionadas<sup>1</sup>.

<sup>1</sup> En este nivel se selecciona una muestra aleatoria de funciones teniendo en cuenta que la misma incluya funciones de todos los tipos y se vuelven a medir.

Según sus autores, este proceso de validación ha probado ser exitoso para detectar rápidamente las cuentas de FP incorrectas. Sin embargo, puede haber ejemplos donde una cuenta correcta podría aparecer como incorrecta. Esto puede ocurrir cuando el software no sigue los patrones de comportamiento esperado o ha sido especificado parcialmente. Por el contrario, también podría haber situaciones en que las reglas del FPA fueron aplicadas incorrectamente y la medición es inválida pero los atributos de la cuenta podrían estar dentro de los límites fijados como condición de excepción. En este caso, el resultado no es visualizado como incorrecto en los niveles alto e intermedio, sin embargo los errores deberían ser detectados en la revisión de bajo nivel.

### 6.2.5.3. Caso A: Sistema de Gestión de Programas

#### *Descripción del caso*

El *Sistema de Presentación de Programas* (SPP) [SPP 2002] era utilizado<sup>45</sup> por los profesores de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco para generar el programa de cada asignatura. El SPP forma parte de un sistema mayor que incluye otras dos aplicaciones - SPPAdmin y SPPWeb - que son usadas exclusivamente por el administrador. El sistema general consta de dos bases de datos - SPP y Programas - y el esquema de funcionamiento es el siguiente:

- Mediante la aplicación SPPAdmin se cargan los datos fijos de las asignaturas (código y nombre de la asignatura, carga horaria, tipo de cursado -anual, cuatrimestral-, asignaturas correlativas, contenidos mínimos y sede) y de los profesores (nombre, DNI y contraseña) en las bases de datos SPP o Programas, según corresponda
- Una vez instalado el software SPP, el profesor utiliza la contraseña que lo habilita para ingresar al sistema y acceder a los datos de su/s asignatura/s. Puede visualizar pero no modificar los datos fijos, generar el programa de la/s asignatura/s a su cargo e incorporar sus datos personales. Esta información es almacenada en una copia local de las bases de datos y es usada para producir las copias digital e impresa del programa que son requeridas por la Facultad.
- La aplicación SPPWeb usa la información digital generada por el SPP para actualizar - en modo *batch* - las bases de datos (SPP y Programas) del sistema general y luego publicar los programas de las asignaturas en el sitio Web de la Facultad.

Este caso se circunscribió a la aplicación SPP usada por los profesores.

---

<sup>45</sup> En el momento en que se comenzó esta investigación; posteriormente fue reemplazado por una aplicación de características y funcionalidad similares, pero en otro ambiente.

El L&E<sup>46</sup>, contruidos aplicando la estrategia descrita en la sección 6.2.5.1, consta de 41 entradas del LEL y 18 escenarios. El L&E puede ser solicitado al autor de esta tesis.

**Realización del estudio y recolección de datos**

**a. Estimar el tamaño funcional de los escenarios con el procedimiento SFP**

En esta sección se presentan los resultados de la estimación del tamaño funcional de los escenarios [Bertolami y Oliveros 2008a] del *Sistema SPP* [SPP 2002].

Los 18 escenarios incluyen un total de 112 episodios, de los cuales fueron descartados 69 por aplicación de las reglas de identificación de episodios. En la Tabla 6.2 se resumen los resultados de la estimación del sistema SPP. La misma incluye el esfuerzo requerido por el proceso de estimación.

Tabla 6.2. Resumen de la estimación del Caso A

EI			FP	EO			FP	Recurso			Total FP	Esfuerzo/FP (h.- pers./FP)
Complejidad <sup>1</sup>				Complejidad						FP		
B	M	A		B	M	A		B	M			
28	0	0	84	15	0	0	45	14	0	98	227	0,02

<sup>1</sup> Complejidad: B: baja, M: media, A: alta

**b. Medir el tamaño funcional de la aplicación con el Método IFPUG FPA**

La medición fue realizada de acuerdo al procedimiento propuesto en el manual para la Medición de Puntos Función [IFPUG 2000]. Como la aplicación existe y está en funcionamiento, el *tipo de la cuenta* es “Medición de FP de una aplicación”<sup>47</sup>. El *alcance* de la medición incluye a toda la funcionalidad suministrada al usuario por el software. El *propósito de la medición* es proporcionar un valor de referencia para compararlo con la estimación de FP basada en los escenarios. En el contexto de este estudio, SPP es considerada una aplicación independiente, por lo tanto el límite está representado por el borde que la separa de otras aplicaciones, como es indicado en la Figura 6.4.

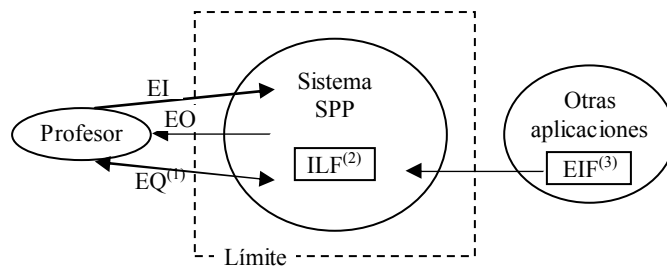


Figura 6.4. La aplicación A desde el punto de vista del usuario [IFPUG 2000]

<sup>(1)</sup>EQ: Consulta Externa; <sup>(2)</sup>ILF: Archivo Lógico Interno; <sup>(3)</sup>EIF: Archivo de Interfaz Externo

La principal dificultad encontrada para identificar las funciones transaccionales y de datos fue la falta de documentación del sistema, lo que significó realizar la medición a partir del software instalado, el código fuente y la aplicación de ayuda. Existen experiencias reportadas en la literatura en que la medición de un sistema con documentación incompleta fue realizada en base al manual del usuario y la aplicación en funcionamiento [Dekkers 2003].

<sup>46</sup> Elaborado por Elena Centeno y el autor de esta tesis.

<sup>47</sup> Se refiere a una aplicación instalada. Este cálculo ofrece una cuenta de las funciones actuales que la aplicación proporciona al usuario [IFPUG 2000].

El análisis del código fuente sirvió para aclarar la lógica de las funciones del sistema en relación con los datos almacenados en las bases de datos del sistema y otras aplicaciones. Los archivos lógicos internos y externos a la aplicación fueron identificados a partir del análisis de las tablas de las bases de datos considerando el punto de vista del usuario (en consecuencia fueron excluidos aquellos campos no significativos para el usuario); en esta aplicación, además, existe un EIF que corresponde a la ayuda en pantalla provista por una aplicación externa. La complejidad y contribución en FPs de los archivos lógicos fue evaluada del modo convencional.

Por ejemplo, el *Programa analítico* es un ILF que almacena los datos de las unidades (*número, título, contenido*) del programa de la asignatura que son mantenidos por procesos elementales de la aplicación (agregar, modificar o borrar una unidad). Esos tres datos son identificados como Tipos de Datos Elementales (DETs). Como no hay subgrupos de datos elementales, hay 1 Tipo de Registro Elemental (RET – Record Element Type). Por lo tanto, usando esos valores (3 DETs y 1 RET) como entrada a la tabla para ILF de IFPUG (2000) se determina que tiene complejidad *baja* y su contribución es de 7 FPs.

La información de la asignatura, que puede ser visualizada pero no modificada por el usuario, es almacenada en el EIF *Datos fijos* que contiene 18 DETs (*Código Asignatura, Nombre asignatura, Carga horaria, Tipo de cursado, Correlativas, etc.*), incluye dos subgrupos de datos elementales (2 RETs) y es mantenido por una aplicación externa (SPPAdmin). De acuerdo a la tabla para EIF de IFPUG (2000), tiene complejidad *baja* y contribuye con 5 FPs.

La mayoría de las transacciones fueron identificadas desde las pantallas e informes. En algunos casos la información suministrada por la aplicación de ayuda y el código permitió aclarar dudas sobre la lógica de un proceso, como por ejemplo, para identificar las EOs. La complejidad de las EIs, EQs y EOs fue determinada utilizando una estrategia similar.

Por ejemplo, el proceso para *Agregar nueva unidad al Programa analítico* (Figura 6.5) es una EI. Los DETs identificados son *Número, Título y Contenido* más los 2 DETs que corresponden a los botones *Nueva Unidad* y *Aceptar* (se debe contar un DET por la posibilidad de especificar una acción a realizar [IFPUG 2000]). Esta EI mantiene el ILF *Programa analítico*, por lo tanto hay 1 Tipo de Archivo Referenciado (FTR – File Type Referenced). Desde la Tabla para EI de IFPUG (2000) se determina que tiene complejidad *baja* y su contribución es de 3 FPs.

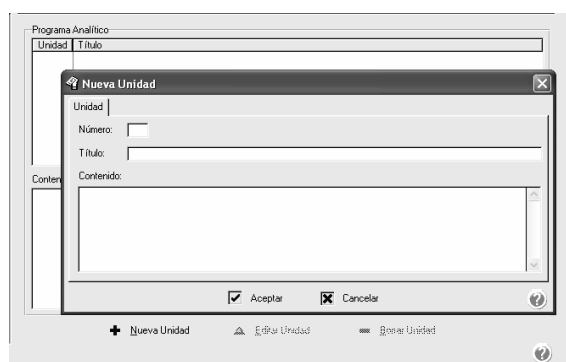


Figura 6.5. Pantalla “Nueva Unidad” del Programa Analítico

La Tabla 6.3 presenta el resumen de la medición indicando la contribución en FP por cada tipo de función (EI, EQ, EO, ILF, EIF) y cada nivel de complejidad (baja, media,



alta), el total de FP de la aplicación y los porcentajes de contribución con relación al total de FP.

Tabla 6.3. Resumen de la medición del Caso A

Función	Complejidad baja		Complejidad media		Complejidad alta		Compl. promedio	Total	%
	Nro.	FP	Nro.	FP	Nro.	FP			
EI	19	57	0	0	1	6	3	63	36%
EO	0	0	1	5	2	14	6	19	11%
EQ	12	36	1	4	0	0	3	40	23%
<b>Transacciones</b>	<b>31</b>	<b>93</b>	<b>2</b>	<b>9</b>	<b>3</b>	<b>20</b>	-	<b>122</b>	<b>70%</b>
ILF	6	42	0	0	0	0	7	42	24%
EIF	2	10	0	0	0	0	5	10	6%
<b>Archivos</b>	<b>8</b>	<b>52</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	-	<b>52</b>	<b>30%</b>
								<b>174</b>	<b>100%</b>

### b.1. Validar la medición

Los resultados de la medición fueron evaluados aplicando la propuesta de Morris y Desharnais (2001) presentada en la sección 6.2.5.2. En los siguientes párrafos se presentan los tres niveles, los aspectos analizados y los resultados de la revisión en cada nivel.

*Revisión de alto nivel.* Este nivel incluye los siguientes pasos:

- *Examinar el tamaño funcional del software.* Este paso permite verificar la conformidad del tamaño funcional en comparación con el predicho para otros atributos del software o por otras métricas. El tamaño funcional fue comparado con la estimación producida mediante *backfiring* [Jones 2006]. Esta técnica provee una tabla de conversión entre el número de líneas de código fuente (SLOC) y los FP para diferentes lenguajes de programación [Jones 1996]. Aunque es cuestionada por la potencial inconsistencia de sus resultados [Desharnais y Abran 2003], [Morris y Desharnais 2001], lo cierto es que es adoptada por muchas herramientas de estimación [Boehm *et al.* 1998], [Jones 2006]. El *backfiring* es un método efectivo cuando no se dispone de la especificación y lo único disponible es el código fuente [Jones 2007]. Conociendo el número de SLOC (8022<sup>48</sup>) y el valor tabular del lenguaje usado (Clarion) - 58 SLOC/FP - se estimaron 138 FP. El resultado de la medición (174 FP) puede considerarse admisible porque: 1) el valor SLOC/FP no fue determinado desde mediciones reales [Jones 1996]; 2) es un 26% mayor que el estimado y por lo tanto está dentro del rango sugerido ( $\pm 30\%$  del predicho por otras métricas), más aún teniendo en cuenta lo señalado en el punto 1).
- *Examinar el límite del software.* La verificación de la correcta identificación de las Els (todas actualizan ILFs) y de la contribución en FPs de los archivos externos con respecto al total de archivos (19% frente al tope sugerido de 50%) permitieron comprobar la correcta ubicación del límite del sistema.
- *Examinar el alcance de la medición de FP.* Considerando el propósito de medición fueron incluidas todas las funciones suministradas al usuario.

<sup>48</sup> Líneas de código físicas, excluyendo líneas en blanco y comentarios [Rozum 1994], contadas manual y automáticamente [GeroneSoft 2005].

*Revisión de nivel intermedio.* Este nivel incluye los siguientes pasos:

- *Examinar la relación entre el número de archivos lógicos y el tamaño funcional.* La relación entre el tamaño funcional y el número de archivos lógicos fue comparada con las establecidas a partir del análisis estadístico de proyectos [Morris y Desharnais 2001]. La diferencia entre los FP medidos (174) y los estimados es del 6% (1) y del 2% (2). Ambas son menores del 20%, lo que sugiere que la cuenta puede ser correcta.

$$FP = 30.813 * \#ILF \therefore FP = 185 \quad (1)$$

$$FP = 22.094 * (\#ILF + \#EIF) \therefore FP = 177 \quad (2)$$

- *Examinar la relación entre el número de funciones.* La proporción entre el número de funciones de entrada y salida y el número de archivos (Tabla 6.3) fue evaluada con respecto a los valores estadísticos de otras organizaciones (Tabla 6.4).

Tabla 6.4. Relación entre transacciones y archivos

Relación	EI			EO			EQ		
	Act. <sup>1</sup>	Esp. <sup>2</sup>	%	Act.	Esp.	%	Act.	Esp.	%
<b>ILF</b>	3.33	2.7	23%	0.50	1.2	-58%	2.17	1.2	81%
<b>ILF+EIF</b>	2.50	1.9	32%	0.38	0.9	-58%	1.63	0.8	103%

<sup>1</sup> Act: actual; <sup>2</sup> Esp.: esperado

El valor de la relación entre las EIs y los ILFs está dentro del rango normal para las transacciones ( $\pm 30\%$ ) y su relación con el total de archivos lógicos es muy aproximada a ese valor. Por el contrario, es menor de lo esperado para las EOs y mayor para las EQs. Esta diferencia puede ser explicada por el tipo de la aplicación medida, cuya función principal es el ingreso de datos y presentación de información, pero la mayoría de las salidas no cumple las condiciones para ser considerada EO. Debido a que el número de EOs es bajo, es razonable que exista un número mayor de EQs.

- *Examinar porcentaje de contribución a la cuenta de FP por cada tipo de función.* Los porcentajes típicos para Sistemas de Información de Gestión (MIS – Management Information System) indican que aproximadamente el 24% de los FP son aportados por los ILFs, entre el 4%-12% por los EIFs, 26%-39% por las EIs, 22%-24% por las EOs y 12%-14% por las EQs [Morris y Desharnais 2001]. Para la medición actual (Tabla 6.3) se observa que: el porcentaje de contribución de EIs, ILFs y EIFs está dentro del rango esperado; los archivos lógicos combinados (ILF+EIF) representan el 30% de los FPs, siendo el rango aceptable del 20% - 40%; la contribución de las EQs y EOs reflejan lo observado en el párrafo anterior, sin embargo, las salidas combinadas (EO+EQ) aportan el 34% de los FPs, que es muy aproximado al rango admisible del 35% - 40%; la contribución de FPs de todas las transacciones (EI+EO+EQ) es del 70%, que está en el rango esperado del 60%-80%.
- *Examinar la complejidad de las funciones.* La complejidad promedio [Morris y Desharnais 2001] de las transacciones y archivos se corresponde con la complejidad media y baja de las tablas de IFPUG (2000), respectivamente. Para la medición actual (Tabla 6.3) el 100% de los archivos lógicos es de complejidad baja. En el caso de las transacciones, la complejidad promedio de las EIs y EQs es menor que lo esperado (la mayoría tiene complejidad baja), lo que es aceptable para esta aplicación en que muchas de sus entradas y consultas manipulan pocos datos y archivos lógicos.

*Revisión de bajo nivel.* Este nivel de la revisión es realizado cuando las revisiones de los niveles previos han detectado que la cuenta es potencialmente válida, entonces la revisión detallada se usa para confirmar la evaluación, o inválida, en cuyo caso la revisión detallada es usada para investigar el origen del error a ser corregido. En este caso, los resultados de la revisión de los niveles previos sugieren que la cuenta posiblemente es válida. El análisis detallado sobre una muestra de funciones transaccionales y de datos ratificó lo expresado en el nivel intermedio, lo que permite suponer que la cuenta es válida.

**Análisis de los resultados**

Esta sección incluye el análisis de los resultados obtenidos (Tabla 6.5) y la comparación con otras propuestas.

Tabla 6.5. Resumen de la estimación/medición de tamaño funcional del Caso A

Estimación SFP	Medición IFPUG
227	174

En los siguientes párrafos se presentan las principales observaciones:

- Los resultados del proceso de validación de la medición de la aplicación con el estándar IFPUG sugieren que la cuenta es válida.
- El error relativo de la estimación con SFP es de  $\pm 30\%$ .
- El esfuerzo de la estimación fue de 0.02 hs./FP.
- Morris (2004) define diferentes “niveles” (de 1 a 6 en orden decreciente de detalle y por ende, de exactitud) para el conteo de FP, donde el nivel depende del propósito para el cual será usado el resultado y puede estar restringido por la calidad de la documentación del proyecto o el tiempo disponible. Para los niveles 5 (*Rough Count*) y 6 (*Size Approximation*) la exactitud está en el orden de  $\pm 20\%$  -  $\pm 25\%$ .
- Comparado<sup>49</sup> con otras propuestas de estimación que fueron presentadas en el Capítulo 3, sección 3.3:
  - 1) Rule of the "Thirties" [ISBSG 2006] permite estimar el tamaño funcional (con un error de  $\pm 30\%$ ) considerando que un archivo lógico aporta entre 31 y 35 FP en un proyecto de desarrollo. En este caso, a partir de las 8 entidades identificadas y el valor 31, el cálculo da como resultado 248 FP. El error relativo es de  $\pm 43\%$ .
  - 2) ISBSG [Hill 2006] propone las siguientes fórmulas:  $FP (ILF) = 8.6 \times \#ILF$ ,  $FP (total) = FP (ILF) \times 100 / 22$ . Para los ILFs identificados, el resultado es  $FP (ILF) = 52$  y  $FP (total) = 235$ . El error relativo es de  $\pm 35\%$ .
  - 3) Una de las propuestas de NESMA Early FPA [NESMA 2004] es Indicative Size  $(FP) = 35 \times \#ILF + 15 \times \#EIF$ . Para el sistema bajo estudio se obtiene Indicative Size  $(FP) = 240$ , por lo tanto, el error relativo es de  $\pm 38\%$ .
  - 4) Prognosis [Bundschuh 1998] aplica la siguiente fórmula  $FP = 7.3 \times \#IO + 56$ , siendo  $\#IO = \#EI + \#EO$ . Considerando las EI y EO identificadas, el resultado es  $FP = 224$ , por lo tanto, el error relativo es de  $\pm 29\%$ .

<sup>49</sup> El error relativo es calculado en todos los casos tomando como valor de referencia el resultado de la medición IFPUG.

**Resumen del caso A**

Los escenarios fueron contruidos desde la aplicación en funcionamiento. La medición IFPUG fue realizada a partir de las funciones identificadas desde el software en funcionamiento, el código fuente y la aplicación de ayuda. La revisión de la medición aplicando la propuesta de Morris y Desharnais (2001) sugiere que la cuenta de FP es válida.

Luego de aplicar sendos procesos, SFP e IFPUG, fue evaluada la exactitud de la estimación de tamaño funcional de los escenarios frente a la medición de la aplicación. Los resultados demuestran que la estimación basada en los escenarios difiere en un 30% de la medición con el método IFPUG y requiere un mínimo esfuerzo. En comparación con otras propuestas de estimación, en la mayoría de los casos el resultado es mejor.

Estrictamente lo establecido permite afirmar la pertinencia de la estimación de los FP de los escenarios de una aplicación en funcionamiento en relación a la medición de los FP que provienen de la aplicación en funcionamiento.

**6.2.5.4. Caso B: Sistema de Gestión de un Comercio**

**Descripción del caso**

Se trata de un *Sistema de Gestión de un comercio*<sup>50</sup> [SGC 2005] dedicado a la venta de indumentaria. La documentación disponible consiste en los diagramas de Casos de Uso (UC), descripción de los UC, diagramas de colaboración, diagrama de clases y el manual de usuario. Se dispone además del software y la base de datos del sistema. La Figura 6.6 (extraída de la documentación del sistema) presenta el diagrama de contexto del sistema.

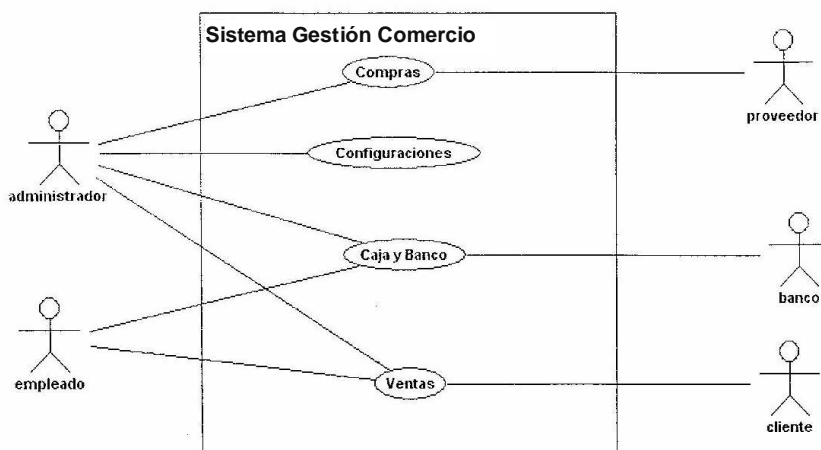


Figura 6.6. Diagrama de contexto del Sistema Gestión de un comercio [SGC 2005]

Los principales requerimientos funcionales del sistema están representados en los UC:

• Configuración	• Caja y Bancos
– Gestión Usuarios	– Vincular Medios Pago
– Gestión Configuraciones generales	– Gestión Cheques
• Compras	– Gestión Conceptos
– Gestión de Proveedores	– Gestión Tarjetas
– Gestión Artículos	– Gestión Depósito

<sup>50</sup> Por razones de confidencialidad se omite el nombre original.

- Gestión Compras	- Gestión Carteras
- Generar Listados	- Gestión de Bancos y Crédito
• Ventas	
- Gestión Ventas	
- Gestión Clientes	
- Generar Listados	

El L&E fue construido siguiendo la estrategia descrita en la sección 6.2.5.1. La versión definitiva del L&E consta de 69 símbolos del LEL y 53 escenarios [Bertolami y Centeno 2008]. El L&E puede ser solicitado al autor de esta tesis.

**Realización del estudio y recolección de datos**

**a. Estimar el tamaño funcional de los escenarios con el procedimiento SFP**

En esta sección se presentan los resultados de la estimación del tamaño funcional de los escenarios del *Sistema de Gestión de un comercio*. Los 53 escenarios incluyen un total de 259 episodios, de los cuales fueron descartados 149 por aplicación de las reglas de identificación de episodios.

La Tabla 6.6 presenta el resumen de los resultados de la aplicación del proceso SFP a los escenarios del sistema. La misma incluye el esfuerzo requerido por el proceso.

Tabla 6.6. Resumen de la estimación del Caso B

EI			FP	EO			FP	Recurso		FP	Total FP	Esfuerzo/FP (h.- pers./FP)
Complejidad <sup>1</sup>				Complejidad				B	M			
B	M	A		B	M	A						
39	5	2	194	61	2	1	152	38	0	266	612	0,02

<sup>1</sup> Complejidad: B: baja, M: media, A: alta

**b. Medir el tamaño funcional de la aplicación con el Método IFPUG FPA**

La medición del *Sistema de Gestión de un comercio* fue realizada de acuerdo al procedimiento propuesto en el manual de IFPUG (2000). Como se trata de una aplicación existente, el *tipo de la cuenta* es “Medición de FP de una aplicación”. El *alcance* de esta medición comprende toda la funcionalidad suministrada al usuario por el software. El *propósito de la cuenta* es proporcionar un valor de referencia para compararlo con el resultado de la estimación basada en los escenarios. Se trata de una aplicación independiente cuyo límite está representado en la Figura 6.7.

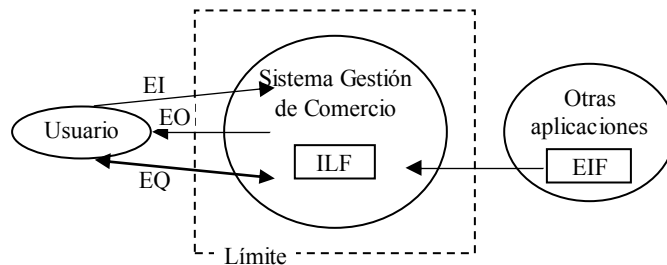


Figura 6.7. La aplicación B desde el punto de vista del usuario [IFPUG 2000]

A partir de la documentación disponible y la aplicación en funcionamiento fueron identificadas y clasificadas las funciones transaccionales y de datos, las que luego fueron medidas del modo habitual.

El análisis de la estructura de las tablas y sus relaciones en la base de datos del sistema favoreció la identificación de los archivos lógicos. En total fueron identificados 18 ILFs y 1 EIF.

Por ejemplo, *Empresas* es un ILF que contiene los datos de una empresa (*cuit*, *razonSocial*, *titular*, *tipoEmpresa*, *localidad*, etc.). Cuando se trata de la empresa local, además de la información general se incluyen los datos de configuración (*fondoCaja*, *logo*, etc.). En total se contabilizaron 15 DETs y 2 RET (el subgrupo obligatorio más el subgrupo *Configuraciones*). Desde la tabla de IFPUG (2000) se determina que este ILF tiene complejidad *baja* y su contribución es de 7 FPs.

La información de ayuda de la aplicación, que puede ser visualizada por el usuario pero es mantenida por otra aplicación, es almacenada en el EIF *Ayuda*.

Las transacciones fueron identificadas a partir de la documentación del sistema, el manual de usuario, la información de ayuda y la aplicación.

Por ejemplo, la transacción para *Agregar nuevo artículo* (Figura 6.8) es una EI. Se identificaron 13 DETs: 10 corresponden a los datos que debe ingresar el usuario para dar de alta un artículo, más 3 DETs que corresponden a los botones *Nuevo* (que inicia la transacción), *Aceptar/Cancelar* y un mensaje de error. El ILF mantenido es *Articulos* y los ILFs referenciados son *Rubros* y *Empresas* (este último se refiere a la empresa del *Proveedor*). Por lo tanto hay 3 FTR. A partir de la tabla de IFPUG se determina que esta EI tiene complejidad *alta* y su contribución es de 6 FPs.

Figura 6.8. Ventana “Crear Artículo”

La Tabla 6.7 presenta el resumen de la cuenta indicando la contribución en FP por cada tipo de función (EI, EQ, EO, ILF, EIF) y cada nivel de complejidad (baja, media, alta), el total de FP de la aplicación y los porcentajes de contribución a la cuenta de FP.

Tabla 6.7. Resumen de la medición del Caso B

Función	Complejidad baja		Complejidad media		Complejidad alta		Compl. promedio	Total	%
	Nro.	FP	Nro.	FP	Nro.	FP			
EI	18	54	11	44	24	144	5	242	45%
EO	4	16	2	10	2	14	5	40	7%
EQ	10	30	22	88	1	6	4	124	23%
<b>Transacciones</b>	<b>32</b>	<b>100</b>	<b>35</b>	<b>142</b>	<b>27</b>	<b>164</b>	-	<b>406</b>	<b>76%</b>
ILF	18	126	0	0	0	0	7	126	23%
EIF	1	5	0	0	0	0	0	5	1%
<b>Archivos</b>	<b>19</b>	<b>131</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	7	<b>131</b>	<b>24%</b>
								<b>537</b>	<b>100%</b>

Cabe mencionar que, aprovechando la documentación de requerimientos en UML [Rumbaugh *et al.* 1999] disponible y como para testear el resultado de la medición por otro camino, también fue aplicada la propuesta de Tavares *et al.* (2002). El resultado obtenido (527 FP) [Bertolami y Oliveros 2008b] sólo difiere en un 2% con el resultado de la medición estándar.

**b.1. Validar la medición**

Para validar los resultados de la medición fue aplicada la propuesta de Morris y Desharnais (2001) (sección 6.2.5.2). En los siguientes párrafos se presentan los tres niveles, los aspectos analizados y los resultados de la revisión en cada nivel.

*Revisión de alto nivel.* Este nivel incluye los siguientes pasos:

- *Examinar el tamaño funcional del software.* El tamaño funcional fue comparado con la estimación obtenida con la propuesta de Jones (2007) para la fase previa a la definición de los requerimientos<sup>51</sup>. Para calcular los FPs deben ser seleccionados en sendas tablas (Tabla 6.8) los valores numéricos correspondientes al alcance, la clase y el tipo de proyecto. Aplicando la siguiente fórmula que combina los tres valores tabulares se obtiene un valor aproximado de los FP:

$$FP = (\text{Valor}_{\text{Scope}} + \text{Valor}_{\text{Class}} + \text{Valor}_{\text{Type}})^{2.35}$$

Tabla 6.8. Ejemplos de valores de alcance, clase y tipo [Jones 2007]

**TABLE 6.7 Examples of Scope, Class, and Type Values**

Scope	Class	Type
1 Subroutine	1 Individual software	1 Nonprocedural
2 Module	2 Shareware	2 Web applet
3 Reusable module	3 Academic software	3 Batch
4 Disposable prototype	4 Single location—internal	4 Interactive
5 Evolutionary prototype	5 Multilocation—internal	5 Interactive GUI or web-based
6 Standalone program	6 Contract project—civilian	6 Batch database
7 Component of system	7 Time sharing	7 Interactive database
8 Release of system	8 Military services	8 Client/server
9 New system	9 Internet	9 Mathematical
10 Compound system	10 Leased software	10 Systems
	11 Bundled software	11 Communications
	12 Marketed commercially	12 Process control
	13 Outsource contract	13 Trusted system
	14 Government contract	14 Embedded
	15 Military contract	15 Image processing
		16 Multimedia
		17 Robotics
		18 Artificial intelligence
		19 Neural net
		20 Hybrid: mixed

En este caso, considerando que el alcance del proyecto es *New System* (9), la clase del proyecto es *Individual software* (1) y el tipo es *Interactive GUI* (5), el resultado es 581 FP. El error relativo de esta estimación con respecto a la medición de IFPUG es

<sup>51</sup> En este caso no fue posible usar *backfiring* porque no estaba disponible el código fuente para contar el número de líneas de código.

±8% y por lo tanto está dentro del rango sugerido (±30% del predicho por otras métricas).

- *Examinar el límite del software.* Se verificó que todas las EIs actualizan ILFs y que la contribución en FPs de los archivos externos con respecto al total de archivos es de 4% (frente al tope sugerido del 50%), lo que permitió comprobar la correcta ubicación del límite del sistema.
- *Examinar el alcance de la medición de FP.* Considerando el propósito de medición fueron incluidas todas las funciones suministradas al usuario.

*Revisión de nivel intermedio.* Este nivel incluye los siguientes pasos:

- *Examinar la relación entre el número de archivos lógicos y el tamaño funcional.* La relación entre el tamaño funcional y el número de archivos lógicos fue comparada con las establecidas a partir del análisis estadístico de proyectos [Morris y Desharnais 2001]. La diferencia entre los FP medidos (537) y los estimados es del -3% (1) y del 22% (2). El primero es menor y el segundo es muy aproximado al ±20%, lo que sugiere que la cuenta puede ser correcta.

$$FP = 30.813 * \#ILF \therefore FP = 555 \quad (1)$$

$$FP = 22.094 * (\#ILF + \#EIF) \therefore FP = 420 \quad (2)$$

- *Examinar la relación entre el número de funciones.* La proporción entre el número de funciones de entrada y salida y el número de archivos (Tabla 6.7) fue evaluada con respecto a los valores estadísticos de otras organizaciones (Tabla 6.9).

Tabla 6.9. Relación entre transacciones y archivos

Relación	EI			EO			EQ		
	Act. <sup>1</sup>	Esp. <sup>2</sup>	%	Act.	Esp.	%	Act.	Esp.	%
<b>ILF</b>	2,94	2.7	9%	0.44	1.2	-63%	1.83	1.2	53%
<b>ILF+EIF</b>	2.79	1.9	47%	0.42	0.9	-53%	1.74	0.8	117%

<sup>1</sup> Act: actual; <sup>2</sup> Esp.: esperado

El valor de la relación entre las EIs y los ILFs está dentro del rango normal para las transacciones (±30%) y la relación con el total de archivos lógicos excede dicho valor, lo cual es razonable pues se trata de una aplicación independiente con un único EIF. Con respecto a las salidas, se observa que es bastante menor de lo esperado para las EOs e inversamente para las EQs. Debido a que hay pocas EOs, es razonable que sea mayor el número de EQs.

- *Examinar porcentaje de contribución a la cuenta de FP por cada tipo de función.* Según los porcentajes típicos para el dominio MIS (Caso A, sección 6.2.5.3, sección Validar la medición), para la medición actual (Tabla 6.7) se observa que: el porcentaje de contribución de los ILFs (23%) está dentro del rango esperado; la baja contribución de los EIFs responde a lo explicado en el párrafo anterior; la contribución de las EIs es algo mayor de lo esperado, sin embargo, representa el 60% del total de FP para las transacciones, que es el límite superior del rango de 30%-60% para las EIs; los archivos lógicos combinados (ILF+EIF) representan el 24% de los FPs, siendo el rango aceptable del 20%-40%; la contribución de las EQs y EOs reflejan lo observado en el párrafo anterior, sin embargo, las salidas combinadas (EO+EQ) aportan el 31% de los FPs, que es aproximado al rango admisible del 35%-40%; la contribución en FPs de todas las transacciones (EI+EO+EQ) es del 76%, que está en el rango esperado del 60%-80%.



- *Examinar la complejidad de las funciones.* Para la medición actual (Tabla 6.7) el 100% de los archivos lógicos es de complejidad baja. En el caso de las transacciones, la complejidad promedio de las EIs es mayor de lo esperado, pero es aceptable porque el número de EIs cuya complejidad es mayor que la media representa el 45% del total, que es menor al valor fijado (90%) como un indicador de una posible identificación incorrecta de los DETs o de los archivos lógicos (esto último podría repercutir en el número de FTRs de la transacción); la complejidad promedio de las EQs y EOs coincide con el valor sugerido.

*Revisión de bajo nivel.* La revisión en los niveles previos sugiere la posible validez de la cuenta de FPs. El análisis detallado sobre una muestra de funciones transaccionales y de datos ratificó lo expresado en el nivel intermedio, lo que permite suponer que la cuenta es válida.

**Análisis de los resultados**

En esta sección se presenta el análisis de los resultados (Tabla 6.10) y la comparación con otras propuestas.

Tabla 6.10. Resumen de la estimación/medición de tamaño funcional del Caso B

Estimación SFP	Medición IFPUG
612	537

En los siguientes párrafos se describen las principales observaciones:

- Los resultados del proceso de validación de la medición de la aplicación con el estándar IFPUG sugieren que la cuenta es válida.
- El error relativo de la estimación es de  $\pm 14\%$ .
- El esfuerzo de la estimación fue de 0.02 hs./FP.
- El error de la estimación está dentro del porcentaje (15%) sugerido por Morris (2004) para el nivel 4 (*Default Complexity Count*).
- Comparado con otras propuestas de estimación que fueron presentadas en el Capítulo 3, sección 3.3:
  - 1) Rule of the "Thirties" [ISBSG 2006]: considerando las 19 entidades identificadas y el valor 31, se obtienen 589 FP. El error relativo es de  $\pm 10\%$ .
  - 2) ISBSG [Hill 2006]: para los ILFs identificados, el resultado es FP (ILF) = 155 y FP (total) = 704, el error relativo es de  $\pm 31\%$ .
  - 3) NESMA Indicative Size: para el sistema bajo estudio Indicative Size (FP) = 699, por lo tanto, el error relativo es de  $\pm 30\%$ .
  - 4) Prognosis [Bundschuh 1998]: considerando las EI y EO identificadas, el resultado es FP = 501, el error relativo es de  $\pm 7\%$ .

**Resumen del caso B**

Los escenarios fueron construidos desde la aplicación en funcionamiento. La medición IFPUG fue realizada a partir de las funciones identificadas desde el software instalado y la documentación disponible. Esta medición fue verificada aplicando la propuesta de Moris y Desharnais (200), concluyendo que la cuenta de FP es potencialmente válida. Una segunda comprobación con el enfoque de Tavares *et al.* (2002) permitió confirmar el

resultado (sólo difiere en el 2%), lo cual es significativo pues ambos enfoques son sustancialmente distintos, especialmente en lo referido a las entidades identificadas para producir la medición.

La exactitud de la estimación de tamaño funcional de los escenarios frente a la medición de la aplicación es de 14%. Los FP fueron obtenidos con un mínimo esfuerzo y en comparación con las técnicas de estimación de tamaño funcional, la aproximación es similar y en algunos casos mejor.

Concretamente, los resultados confirman la pertinencia de la estimación de los FP de los escenarios de una aplicación en funcionamiento en relación a la medición de FP proveniente de la aplicación en funcionamiento. Por lo tanto, se puede concluir que SFP funcionó adecuadamente como herramienta de estimación para un proyecto de software.

### 6.2.5.5. Caso C: Sistema SFP Tool

#### *Descripción del caso*

Esta herramienta CASE fue diseñada para soportar el proceso SFP. EL L&E, construido del modo habitual (es decir, para elicitar los requerimientos), consta de 70 entradas del LEL y 15 escenarios [Bertolami *et al.* 2008]. En el Apéndice D se presentan todos los detalles del caso: L&E, formularios usados para la estimación con SFP, la SRS y la medición con el método IFPUG. En el Apéndice E se presenta la descripción de la herramienta.

#### *Realización del estudio y recolección de datos*

##### a. Estimar el tamaño funcional de los escenarios con el procedimiento SFP

En esta sección se presentan los resultados de la estimación del tamaño funcional de los escenarios del *Sistema SFP Tool*. Los 15 escenarios incluyen un total de 106 episodios, de los cuales fueron descartados 72.

La Tabla 6.11 presenta el resumen de los resultados de la aplicación del proceso SFP a los escenarios del sistema. La misma incluye el esfuerzo requerido por el proceso.

Tabla 6.11. Resumen de la estimación del Caso C

EI			FP	EO			FP	Recurso		FP	Total FP	Esfuerzo/FP (h.- pers./FP)
Complejidad <sup>1</sup>				Complejidad				B	M			
B	M	A		B	M	A						
24	0	0	72	7	2	1	35	17	0	119	226	0,02

<sup>1</sup> Complejidad: B: baja, M: media, A: alta

##### b. Medir el tamaño funcional de la aplicación con el Método IFPUG FPA

La medición del Sistema SFP Tool fue realizada de acuerdo al procedimiento propuesto en el manual de IFPUG (2000). Como se trata de una aplicación instalada, el *tipo de la cuenta* es “Medición de FP de una aplicación”. El *alcance* de esta medición comprende toda la funcionalidad suministrada al usuario por el software. El *propósito de la cuenta* es proporcionar un valor de referencia para compararlo con el resultado de la estimación basada en los escenarios. Se trata de una aplicación independiente cuyo límite está representado en la Figura 6.9.

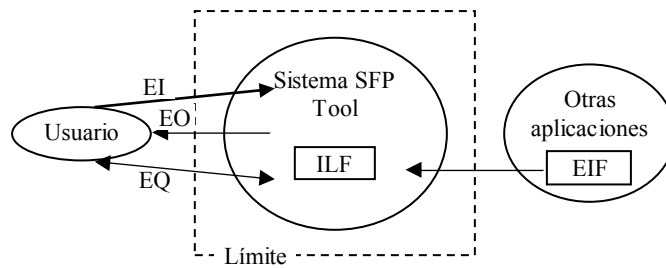


Figura 6.9. La aplicación C desde el punto de vista del usuario [IFPUG 2000]

A partir de la especificación de requerimientos fueron identificadas y clasificadas las transacciones y los archivos lógicos, que luego fueron medidos del modo convencional.

La Tabla 6.12 presenta el resumen de la cuenta indicando la contribución en FP por cada tipo de función (EI, EQ, EO, ILF, EIF) y cada nivel de complejidad (baja, media, alta), el total de FP de la aplicación y los porcentajes de contribución a la cuenta de FP.

Tabla 6.12. Resumen de la medición del Caso C

Función	Complejidad baja		Complejidad media		Complejidad alta		Compl. promedio	Total	%
	Nro.	FP	Nro.	FP	Nro.	FP			
EI	14	42	5	20	0	0	3	62	34%
EO	6	24	0	0	1	7	4	31	17%
EQ	11	33	1	4	0	0	3	37	21%
<b>Transacciones</b>	<b>31</b>	<b>99</b>	<b>6</b>	<b>24</b>	<b>1</b>	<b>7</b>	-	<b>130</b>	<b>72%</b>
ILF	5	35	0	0	0	0	7	35	19%
EIF	3	15	0	0	0	0	5	15	8%
<b>Archivos</b>	<b>8</b>	<b>50</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	-	<b>50</b>	<b>28%</b>
								<b>180</b>	<b>100%</b>

A diferencia de los casos anteriores, esta medición no fue validada aplicando la propuesta de Morris y Desharnais (2001) pues, como destacan los autores, el proceso de validación es principalmente adecuado para el dominio MIS.

### Análisis de los resultados

En esta sección se presenta el análisis de los resultados (Tabla 6.13) y la comparación con otras propuestas.

Tabla 6.13. Resumen de la estimación/medición de tamaño funcional del Caso C

Estimación SFP	Medición IFPUG
226	180

En los siguientes párrafos se describen las principales observaciones:

- El error relativo de la estimación es de  $\pm 26\%$ .
- El esfuerzo de la estimación fue de 0.02 hs./FP.
- El error de la estimación es muy aproximado al límite superior (25%) sugerido por Morris (2004) para los niveles 5 (*Rough Count*) y 6 (*Size Approximation*).
- Comparado con otras propuestas de estimación que fueron presentadas en el Capítulo 3, sección 3.3:

- 1) Rule of the "Thirties" [ISBSG 2006]: considerando las 8 entidades identificadas y el valor 31, el resultado es 248 FP. El error relativo es de  $\pm 38\%$ .
- 2) ISBSG [Hill 2006]: para los ILFs identificados, el resultado es FP (ILF) = 43 y FP (total) = 195, el error relativo es de  $\pm 9\%$ .
- 3) NESMA Indicative Size [NESMA 2004]: para el sistema bajo estudio, Indicative Size (FP) = 220, el error relativo es de  $\pm 22\%$ .
- 4) Prognosis [Bundschuh 1998]: considerando las EI y EO identificadas, el resultado es FP = 246, el error relativo es de  $\pm 37\%$ .

### Resumen del caso C

El L&E fueron generados en la fase de Elicitación de Requerimientos. La medición IFPUG fue realizada en base a la especificación de requerimientos.

El error relativo de la estimación con SFP es de  $\pm 26\%$  y el proceso fue realizado con un mínimo esfuerzo. En comparación con las otras técnicas de estimación, la aproximación es similar y en dos casos mejor.

#### 6.2.6. Alcance de la evaluación

En esta sección se presenta el alcance de la evaluación basada en el Caso de Estudio, el análisis efectuado en función del mismo y las acciones correctivas aplicadas, en el caso de ser posible, para minimizar el impacto de aquellos aspectos que pudieran ser considerados una potencial limitación.

1. El número de casos es escaso en el contexto de un experimento, sin embargo, en un Caso de Estudio, de acuerdo a lo establecido por Yin (2003), es suficiente, especialmente teniendo en cuenta el objetivo de este trabajo.
2. El tamaño de los sistemas A, B y C es pequeño considerando la clasificación propuesta por Jones (2008a) (Tabla 6.14), pues, de acuerdo a los resultados de las mediciones, sus tamaños en FP están comprendidos en la categoría de 100 a 1000 FP.

Tabla 6.14. Taxonomía para el tamaño del software [Jones 2008a]

Descriptive Term	Size in Function Points	Size in KLOC
1 = Very small applications	1–100	1–10
2 = Small applications	100–1,000	10–100
3 = Low-medium applications	1000–2,500	100–250
4 = Medium applications	2500–10,000	250–1,000
5 = Large systems	10,000–25,000	1,000–2,500
6 = Very large systems	25,000–100,000	2,500–10,000
7 = Super large systems	>100,000	>10,000

3. El L&E de los tres sistemas fue desarrollado con la colaboración de otros integrantes del grupo de investigación, las mediciones con SFP e IFPUG fueron realizadas por el autor de esta tesis, por lo que puede haberse introducido un sesgo.
4. La medición con el método IFPUG no fue realizada por un especialista certificado. Algunos estudios demuestran que, incluso las estimaciones realizadas por dos o más especialistas certificados pueden tener una variación de un 12% [Kemerer 1993].

Con respecto al primer y segundo aspecto se evaluaron dos alternativas: aplicar SFP sobre otros sistemas pequeños o sobre un sistema grande. Un análisis del costo beneficio de la adopción de alguna de las alternativas mencionadas resultó en que requerirían un esfuerzo significativo sin agregar valor, puesto que aún persistiría el potencial sesgo mencionado en el tercer punto. De todas maneras, vale mencionar que en la fase de prueba del software SFP Tool, un integrante del equipo de investigación calculó los FP de los escenarios del caso *Notificaciones a los Clientes de un Banco*. Si bien este caso no formó parte de esta evaluación, la diferencia con la estimación realizada por el autor de esta tesis fue de un 4%, que es mínima considerando el diferente grado de entrenamiento con la técnica y el paralelo aprendizaje en el uso del software.

Con respecto al cuarto punto, este aspecto no debería invalidar la propuesta pues se apunta a lograr la mejor aproximación posible del tamaño de un sistema para compararla con los FP de los escenarios. Puesto que la estimación está orientada a la etapa de Elicitación de Requerimientos, para la que la estimación de tamaño puede estar comprendida dentro de un rango muy amplio, las potenciales diferencias con la medición de uno o más expertos no resultarían relevantes en este contexto. De todas maneras la cuestión no fue soslayada, ante la falta de mediciones propias sobre proyectos similares que pudieran servir como parámetro de comparación, fue aplicada, cuando fue posible, la propuesta de Morris y Desharnais (2001) para evaluar la consistencia interna de la cuenta de IFPUG FP.

Por último, un factor que no fue considerado previamente y que podría impactar sobre los resultados produciendo estimaciones erróneas o incrementando el esfuerzo indebidamente, es el nivel de entrenamiento con SFP. Para ello se aseguró un suficiente adiestramiento antes de aplicarlo al Caso de Estudio, el que fue adquirido aplicando SFP a los casos detallados en la sección 6.1.

#### **6.2.7. La incertidumbre de la estimación**

Como fue destacado en el Capítulo 3, las estimaciones de tamaño producidas al inicio de un proyecto de desarrollo de software deberían ser revaluadas en las siguientes fases, a medida que se incorpora mayor conocimiento sobre el sistema. En los desarrollos tradicionales es recomendable estimar como mínimo en la fase de factibilidad (macro estimación), fase de requerimientos (estimación detallada) y fase de diseño (estimación refinada) [Laird y Brennan 2006].

La necesidad de exactitud de una estimación está relacionada con su propósito. Una estimación del costo (que es determinada en función del tamaño del proyecto y el costo por FP) en la fase de factibilidad de un proyecto sólo debe tener la suficiente exactitud como para soportar la decisión de avanzar hacia una definición más detallada del proyecto [Laird y Brennan 2006].

En las etapas más tempranas (antes de completar la SRS) el rango de las estimaciones de tamaño puede variar entre  $1.5x - 0.67x$  (siendo  $x$  el tamaño final del sistema) y llega hasta  $4x - 0.25x$  en la fase de factibilidad, tal como puede observarse en el “Cono de incertidumbre” propuesto por Boehm (1998) (Figura 6.10). Esto significa que el rango total es de  $16x$ . Es importante destacar que el Cono de incertidumbre representa la mejor estimación posible, en diferentes puntos del desarrollo, que pueden lograr estimadores largamente experimentados: *"It's easily possible to do worse. It isn't possible to be more accurate; it's only possible to be more lucky"*<sup>52</sup> [McConnell 2006].

---

<sup>52</sup> Es fácil hacerlo peor. No es posible ser más exacto, sólo es posible ser más afortunado [McConnell 2006]. (traducción propia).

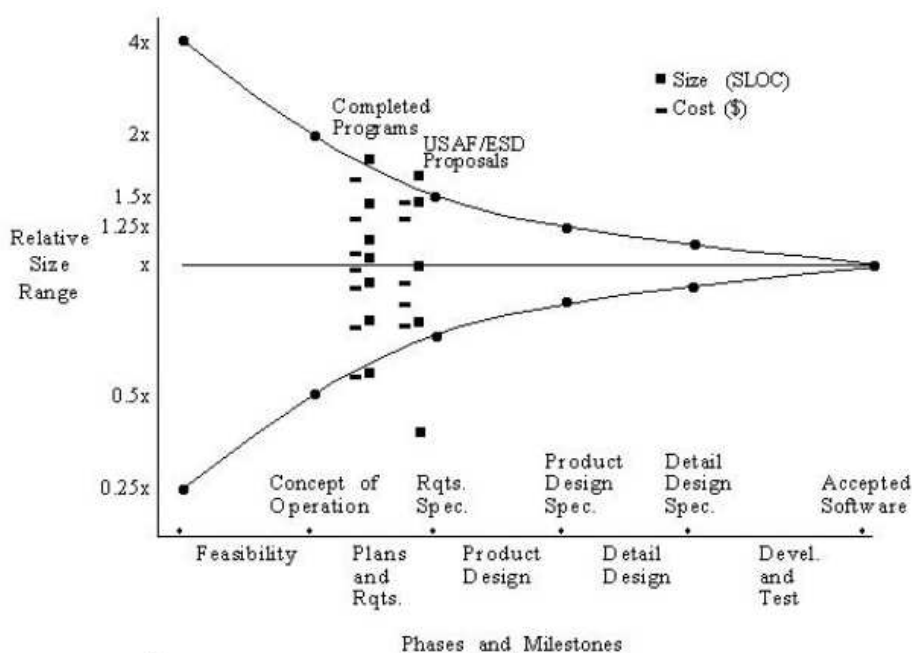


Figura 6.10. Cono de incertidumbre [Boehm *et al.* 1998]

Como puede observarse, la estimación es un tema complejo, al respecto McConnell (1998) afirma: “no sólo es difícil estimar con exactitud un proyecto en las etapas tempranas, es *teóricamente imposible*. Al final de la fase de desarrollo de los requerimientos quedan múltiples decisiones pendientes que recién van a ser tomadas durante la arquitectura, diseño detallado y construcción. La persona que afirma que es capaz de estimar el impacto de esa miríada de decisiones antes de que realmente sean un hecho, es un profeta o no está muy bien informada sobre la naturaleza intrínseca del desarrollo de software.”.

### 6.2.8. Conclusiones del Caso de Estudio

En la Tabla 6.15 se presenta el resumen con los resultados para los tres casos evaluados en el marco del Caso de Estudio.

Tabla 6.15. Resumen de los resultados de Caso de estudio

Caso	Estimación SFP	Medición IFPUG	Error relativo
Sistema de Presentación de Programas	227	174	±30%
Sistema Gestión Comercio	612	537	±14%
SFP Tool	226	180	±26%

Una vez determinada la exactitud de la estimación se debe comprobar si ese valor es aceptable. Ahora bien, ¿qué se entiende por valor aceptable?, ¿cuánto es? En general hay acuerdo que las mediciones con el método IFPUG sobre especificaciones de requerimientos (fase más temprana en que puede ser aplicado) pueden tener una desviación de ±10% (algunos autores afirman que puede ser de ±5% [Jones 2008b]) respecto a la medición sobre el producto terminado, sin embargo, no se encontró una opinión generalizada acerca del valor admisible para las estimaciones más tempranas. Analizando las técnicas de estimación (Capítulo 3, sección 3.3) puede observarse que las orientadas a la fase de factibilidad reportan una exactitud que varía entre ±10% y ±50% y otras propuestas basadas en el modelo de datos entre ±10% y ±30%. Considerando la

Figura 6.10, la estimación más optimista podría tener un error mayor que +50% (1.5x) ó -33% (0.67x), que son errores posibles cuando está completa la SRS.

Los resultados obtenidos a partir de los tres casos usados para este estudio están dentro del margen aceptable para una estimación tan temprana, de acuerdo a las consideraciones expuestas en el párrafo anterior. Como ya fue mencionado, esta evaluación tuvo por objetivo aplicar SFP como una prueba de conceptos. Los resultados no son generalizables, porque es sabido que un caso de estudio puede mostrar los efectos de una tecnología en una situación típica, pero no puede ser generalizado a cada situación posible [Kitchenham 1995]; por otro lado, se requiere extender la experimentación sobre un número mayor de casos y con aplicaciones de variado tamaño. En este contexto, los resultados son satisfactorios en relación al primer interrogante de la sección 6.2.1, debido a que se demostró que la estimación del tamaño funcional basada en escenarios es consistente con la medición del producto terminado. Sin embargo, debe quedar claro que, por las características del estudio llevado a cabo, no se pueden hacer inferencias acerca del grado de incertidumbre de la estimación, que *a priori* se podría suponer que es elevado, aunque no hay argumentos para sostener o rechazar tal afirmación.

Otra cuestión a tener en cuenta y que vale destacar, es que en este Caso de Estudio se ratificaron los valores de esfuerzo (0,02 hora-persona/FP) observados en la sección 6.1. Expresado en términos de productividad, considerando el promedio de todas las mediciones, está en el orden de los 40 FP/hora-persona. Al respecto, Jones (2008a) dice: “en general los rangos contenidos en los datos del autor son muy amplios: varían desde una muy baja, de aproximadamente 0.13 FP por persona-mes (más de 1.000 horas de trabajo por FP) hasta muy alta, con más de 200 FP por persona-mes (0.66 hora de trabajo por FP)”. Estas amplias variaciones pueden ser explicadas analizando diferentes factores que inciden en la productividad, como por ejemplo, el tamaño de las aplicaciones, el tipo de software y otros factores asociados con el desarrollo del software como herramientas, lenguajes y procesos [Jones 2008a]. Si bien estos últimos valores de productividad corresponden a las mediciones con el método estándar, representan un parámetro de referencia para justificar la viabilidad de la propuesta desde el punto de vista del esfuerzo.

Por último, la complejidad del 95% de las EIs, 88% de las EOs y 100% de los recursos es baja, considerando el promedio para los tres casos (Tablas 6.2, 6.6 y 6.11). Estos resultados son similares a los mencionados en la sección 6.1. Este análisis puede ser interesante para desarrollar, a futuro, una simplificación del proceso mediante el uso de valores de complejidad promedio para las EIs, EOs y recursos.

### 6.2.9. Comparación con otras propuestas de estimación

El análisis comparativo, basado en los criterios (Capítulo 3, sección 3.1) establecidos para el análisis de los enfoques de estimación, permite concluir:

- Su aplicabilidad es más amplia que la de otras propuestas, puesto que, como fue especificado en la definición de los objetivos (Capítulo 4, sección 4.1.1), el dominio funcional es genérico.
- De modo similar a otras propuestas es aplicable en las etapas iniciales de un proyecto de software, específicamente a partir de la Elicitación de Requerimientos. Puesto que los escenarios evolucionan a través del ciclo de desarrollo, también puede ser aplicado en las siguientes fases, representando una alternativa para determinar el tamaño de un sistema cuando es analizado su mantenimiento o reemplazo.
- A diferencia de otros enfoques, cuyo mínimo requisito puede ser, por ejemplo, el modelo de datos, SFP sólo necesita los escenarios para producir una estimación. Esto

significa que se puede obtener la estimación más temprano y con menos esfuerzo, pues, en ese caso no sería necesario esperar a tener el modelo de datos ni invertir tiempo para producirlo. Otras propuestas requieren un mayor grado de conocimiento del sistema, de modo que puedan ser identificadas algunas (por ejemplo, los archivos lógicos) o todas las funciones de IFPUG.

- La mayoría de los enfoques de estimación no incluye reglas propias para asistir el proceso. SFP cuenta con conjunto de reglas para identificar, clasificar y cuantificar los componentes funcionales definidos por el modelo.
- Al igual que en otros enfoques fue definida una función para calcular los FP; la diferencia radica en que algunos usan la propia del estándar IFPUG.
- A diferencia de las propuestas de estimación estudiadas, que en general no incluyen una herramienta de soporte, el proceso SFP es facilitado por el software SFP Tool.
- De modo similar a otras propuestas de estimación, la evaluación fue realizada sobre un caso de estudio que incluye tres sistemas reales. Con respecto al número de casos usado para la comprobación de SFP, esto puede ser analizado desde diferentes perspectivas, por un lado, el reducido número de casos es una debilidad desde el punto de vista de una rigurosa validación, pero, como ya fue señalado, no fue ése el espíritu de esta evaluación; por otro lado y a favor de esta propuesta, puede decirse que un porcentaje significativo de los enfoques de estimación revisados no fue validado o en el mejor de los casos, la verificación realizada no está disponible.
- En lo referido a la exactitud de la estimación, los resultados obtenidos se encuentran dentro del rango observado en otras propuestas y en algunos casos es mejor, más aún considerando que los resultados que están en el orden de  $\pm 10\%$  se corresponden con estimaciones basadas en Especificaciones de Requerimientos. La Tabla 6.16 incluye el error relativo calculado para las propuestas consideradas tomando como valor de referencia la respectiva medición IFPUG.

Tabla 6.16. Resumen de los resultados

Caso	Técnica	FP	Error relativo <sup>1</sup>
Sistema de Presentación de Programas	<b>SFP</b>	227	<b><math>\pm 30\%</math></b>
	Rule of Thirties	248	$\pm 43\%$
	ISBSG	235	$\pm 35\%$
	NESMA	240	$\pm 38\%$
	Prognosis	224	$\pm 29\%$
SFP Tool	<b>SFP</b>	226	<b><math>\pm 26\%</math></b>
	Rule of thirties	248	$\pm 38\%$
	ISBSG	195	$\pm 9\%$
	NESMA	220	$\pm 22\%$
	Prognosis	246	$\pm 37\%$
Sistema Gestión Comercio	<b>SFP</b>	612	<b><math>\pm 14\%</math></b>
	Rule of thirties	589	$\pm 10\%$
	ISBSG	704	$\pm 31\%$
	NESMA	645	$\pm 20\%$
	Prognosis	501	$\pm 7\%$

<sup>1</sup> Valor de referencia: resultado de la medición IFPUG.



Como puede observarse, los resultados de este análisis comparativo son alentadores en cuanto a la satisfacción de la segunda pregunta del comienzo de esta evaluación (sección 6.2.1), pues SFP produjo estimaciones de tamaño funcional cuya exactitud es comparable a la de otras propuestas de estimación.

### **6.3. Resumen y Conclusiones**

En primer lugar se comprobó la posibilidad de estimar el tamaño funcional a partir de la información contenida en los escenarios, la pertinencia de los pasos del proceso y la utilidad de las reglas propuestas. Esto puede ser afirmado desde que las mediciones, manuales o con la herramienta, fueron realizadas también por otras personas además del autor de esta tesis, quienes en términos generales no encontraron dificultades para realizar el proceso y sólo reportaron algunas dudas que sirvieron para aclarar la especificación del proceso. Paralelamente, y muy importante, se determinó que el esfuerzo del proceso es irrelevante frente a la ventaja de tener una estimación en una fase tan temprana del desarrollo. Este aspecto es fundamental debido a que no sería de mucha utilidad un proceso que, aunque bien estructurado, tuviera un costo significativo en términos de esfuerzo.

La evaluación de la exactitud de la estimación permitió comprobar que, en el contexto del Caso de Estudio, las estimaciones de SFP son compatibles con sendas mediciones del producto terminado y los resultados son similares a los obtenidos con otras propuestas de estimación. Adicionalmente, en esta evaluación se corroboraron los valores de esfuerzo obtenidos previamente, los que, por otra parte, son consistentes con los sugeridos para las mediciones con el método de medición estándar.

Desde lo antes expuesto y considerando que los resultados son prometedores, se puede concluir que es necesario aplicar SFP sobre una muestra que incluya un mayor número de sistemas y además que éstos sean de variados tamaños, con el objetivo de validar la propuesta y generalizar los resultados en caso de ratificarse los resultados existentes.



## Capítulo 7

### Conclusiones y Trabajos futuros

En este capítulo se presentan las conclusiones a las que se arribó como resultado de la investigación, se destacan las contribuciones que se espera haber aportado con esta tesis y las futuras líneas de investigación.

#### 7.1. Conclusiones

A continuación se enumeran las principales conclusiones de este trabajo de tesis, destacando, en los puntos 2 a 6, como fueron respondidas las preguntas formuladas en el Capítulo 1, sección 1.1 de esta tesis.

1. La revisión del estado del arte sobre las técnicas de estimación del tamaño funcional demostró que, en general, no se han dedicado esfuerzos orientados al desarrollo de enfoques para ser aplicados a los artefactos generados en la Elicitación de Requerimientos. En particular, no se encontró ninguno que utilice los escenarios basados en el enfoque de Leite, con excepción del propuesto por el autor de esta tesis que fue mencionado en el Capítulo 1.
2. El procedimiento SFP fue desarrollado a partir de la asociación entre los componentes de los escenarios y los del Método IFPUG FPA, la cual fue establecida a través de un análisis exhaustivo de los conceptos subyacentes a ambos modelos y descripta mediante un conjunto de reglas que conducen a la cuantificación del tamaño funcional de los escenarios. La aplicación de SFP sobre un grupo de casos de estudio - de los cuales sólo estaba disponible el L&E y no fueron implementados - permitió comprobar la pertinencia de las reglas y la secuencia especificada para el proceso definido. Por lo tanto, se puede concluir que ha sido respondida positivamente la primera de las preguntas mencionadas, es decir, es posible estimar el tamaño funcional de un sistema de software a partir de los escenarios.
3. Fue desarrollada una especificación rigurosa para el proceso SFP que facilita la capacitación del usuario y la aplicación de la técnica. Además, fue diseñado un juego de formularios para la recolección y mantenimiento de los datos y el seguimiento de la estimación desde el principio hasta el final. También, y relacionado con lo anterior, fue implementada una herramienta CASE que soporta el proceso de estimación.
4. En cuanto al esfuerzo requerido por el proceso, los resultados de la medición manual ratificaron que es irrelevante frente al requerido por una medición estándar y puede ser reducido aún más mediante la herramienta mencionada. En consecuencia, desde lo expresado en el párrafo anterior y el presente, se puede afirmar que es posible resolver el problema mediante un proceso eficiente y con un mínimo esfuerzo, de modo que se ha dado una respuesta afirmativa a la cuarta pregunta.
5. La evaluación de SFP fue realizada mediante un Caso de estudio basado en tres sistemas reales sobre los cuales fueron ejecutados sendos procesos de estimación y medición y cuyos resultados permitieron determinar la exactitud de SFP. De acuerdo a los términos en que fue abordado el caso de estudio - una prueba de conceptos - y con las limitaciones señaladas en el Capítulo 6, fue respondida afirmativamente la segunda pregunta, es decir, los resultados producidos por SFP son consistentes con los obtenidos en una etapa posterior, específicamente desde el producto implementado, es decir, su exactitud se encuentra dentro del rango aceptable para una estimación

temprana. Por último, el caso de estudio también permitió confirmar los resultados relativos al esfuerzo, de modo que fue corroborada la conclusión del párrafo anterior.

6. SFP fue comparado con otras propuestas de estimación temprana de acuerdo a un conjunto de criterios establecidos en el Capítulo 3, sección 3.1. Como resultado de esta evaluación se puede concluir que SFP: puede ser usado más temprano y con menos esfuerzo, pues puede ser aplicado desde la Elicitación de Requerimientos en adelante y sólo es necesario desarrollar los escenarios, incluye un conjunto de reglas que facilitan la identificación, clasificación y cuantificación de los componentes funcionales en los escenarios, provee una herramienta de soporte, fue evaluado mediante un caso de estudio basado en sistemas reales y por último, la exactitud está comprendida dentro del rango observado en otras propuestas y en algunos casos es mejor. Esto último permite concluir que también fue respondida en forma satisfactoria la tercera pregunta.

La conclusión general es que se ha cumplido con el objetivo de esta tesis, esto significa que se ha proporcionado un procedimiento de estimación que permite obtener el tamaño funcional de un sistema de software desde los escenarios basados en el enfoque de Leite (2000).

### 7.2. Resumen de las contribuciones

Las principales contribuciones aportadas por este trabajo de tesis son las siguientes:

1. En un contexto en que poca atención han recibido los artefactos de la Elicitación de Requerimientos como fuente de información para la estimación del tamaño funcional de un proyecto de software, se ha suministrado una propuesta concebida específicamente para tal fin.
2. Esta nueva propuesta permite estimar el tamaño funcional más temprano en el ciclo de vida del software y con una exactitud comparable a otras propuestas.
3. Se comprobó que la estimación de tamaño funcional basada en escenarios puede representar una alternativa para el conteo de los FP de una aplicación instalada, los que podrían ser usados cuando se estudia la factibilidad de un proyecto de mejoras o se evalúa el reemplazo de un sistema y no se dispone de la documentación.

Otras contribuciones que merecen ser mencionadas son las siguientes:

4. Se elaboró una estrategia para construir los escenarios a partir de un sistema implementado.
5. Se diseñó un conjunto de plantillas de formularios para facilitar el proceso de estimación y mantener la información relativa a la estimación y el esfuerzo durante todas las fases del proceso.
6. Se desarrolló una herramienta de soporte para SFP que permite agilizar la ejecución del proceso de estimación.
7. Se desarrolló la especificación del proceso SFP.

### 7.3. Trabajos futuros

El trabajo presentado en esta tesis deja abiertas algunas cuestiones que merecerían ser atendidas en el futuro trabajo de investigación. Entre ellas pueden mencionarse:

- Ampliar la experimentación sobre un mayor número de casos, incluyendo sistemas de tamaños variados, con vistas a realizar una validación empírica de los resultados. Si este proceso resultara exitoso, pueden ser investigadas dos alternativas tendientes a agilizar el proceso SFP, ambas basadas en el análisis estadístico de los datos recolectados en la fase experimental:
  - o simplificar el proceso SFP usando valores promedio de complejidad para los episodios y los recursos.
  - o describir un modelo de predicción basado en regresión y evaluar la confiabilidad del modelo mediante pruebas estadísticas.

Ambas propuestas reducirían significativamente el esfuerzo, pues sólo bastaría con identificar y contabilizar los episodios y los recursos.
- Extender la propuesta para aplicarla sobre otras especificaciones, específicamente sobre Casos de Uso.



## Referencias

- [Abrahão 2004] Abrahão, S., On The Functional Size Measurement of Object-Oriented Conceptual Schemas: Design and Evaluation Issues, PhD Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España, 2004.
- [Abrahão *et al.* 2004a] Abrahão S., Poels, G., Pastor, O., “Comparative Evaluation of Functional Size Measurement Methods: An Experimental Analysis”, Working Paper, 2004.
- [Abrahão *et al.* 2004b] Abrahão, S., Poels, G., Pastor, O. “A Functional Size Measurement Method for Object-Oriented Conceptual Schemas: Design and Evaluation Issues”, Working Paper 200/233, Faculty of Economics and Business Administration, Ghent University, Belgium, 2004, 48 p.
- [Abran *et al.* 2000] Abran, A., Oligny, S., Symons, C., St-Pierre, D., Desharnais, J., “Functional Size Measurements Methods - COSMIC-FFP: Design and Field Trials”, *FESMA-AEMES Software Conference Measurements*, 2000.
- [Abran *et al.* 2004] Abran, A., Meli, R., Symons, C., “COSMIC-FFP (ISO 19761) Software size measurement: State of the Art 2004”, 2004.
- [Abran y Jacquet 1999] Abran, A., Jacquet, J., "A Structured Analysis of the new ISO Standard on Functional Size Measurement - Definition of Concepts", (ISO/IEC 14143-1), 4th IEEE Int. Symposium and Forum on Software Engineering Standards, ISESS'99, Curitiba, Brazil, May 17-22, 1999.
- [Abran y Robillard 1994] Abran, A., Robillard, P., “Function Points: A Study of Their Measurement Processes and Scale Transformations”, *Journal of Systems Software*, Nro.25, pp. 171-184, 1994.
- [Albrecht 1979] Albrecht, A., “Measuring Application Development Productivity”, *IBM Applications Development Symposium*, Monterey, CA, pp. 83-92, 1979.
- [Anda *et al.* 2005] Anda, B., Benestad, H., Hove, S., “A Multiple-Case Study of Effort Estimation based on Use Case Points”, *Proceedings ISESE'2005* (Fourth International Symposium on Empirical Software Engineering), IEEE Computer Society, Noosa, Australia, pp. 407–416, 2005.
- [Antonelli *et al.* 1999] Antonelli, L., Rossi, G., Oliveros A., “Baseline Mentor, An Application that Derives CRC Cards from Lexicon and Scenarios”, LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Argentina, 1999.
- [Antoniol *et al.* 1999] Antoniol, G., Lokan, C., Caldiera, G., Fiutem, R., “A Function Point-Like Measure for Object-Oriented Software”, in *Empirical Software Engineering an International Journal*, vol. 4, pp. 263-287, 1999.
- [Arnold y Pedross 1998] Arnold, P., Pedross, P., “Software Size Measurement and Productivity Rating in a Large-Scale Software Development Department”, *Forging New Links*, IEEE Comput.Soc, Los Alamitos, CA, USA, pp. 490-493, 1998.
- [Asensio Monge *et al.* 2004] Asensio Monge, R., Sanchis Marco, F., Torre Cervigón, F., García García, V., Uría Paino, G., “A Preliminary Study for the development of an Early Method for the Measurement in Function Points of a Software

- Product”, 2004, <http://arxiv.org/pdf/cs.SE/0402015>, consultado en junio de 2005.
- [Ballesteros *et al.* 1997] Ballesteros, C., Minetti, G., Salto, C., “LEL y Escenarios de un Sistema de Alumnos Centro Regional de Educación Tecnológica (C.E.R.E.T.)”, caso de estudio desarrollado para el curso Tópicos I del Magister en Ingeniería de Software en la Universidad Nacional de La Plata, 1997.
- [Banerjee 2001] Banerjee, G. “Use Case Points - An Estimation Approach”, White paper, 2001.
- [Benner *et al.* 1993] Benner, K. M, Feather, M. S., Johnson, W. L., Zorman, L. A., “Utilizing Scenarios in the Software Development Process”, USC, Information Sciences Institute, Admiralty Way, Marina del Rey, USA, 1993.
- [Bertolami 2003] Bertolami, M., Una propuesta de Análisis de Puntos Función aplicado a LEL y Escenarios, Tesis del Magíster en Ingeniería de Software, Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina, 2003.
- [Bertolami *et al.* 2008], Bertolami, M., Olinik, R., Centeno, E., LEL y Escenarios de SFP Tool, Proyecto de Investigación, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina, 2008.
- [Bertolami y Centeno 2001] Bertolami, M., Centeno, E., “LEL y Escenarios de la Recepción del Hotel”, Caso de estudio desarrollado en el marco del Magister en Ingeniería de Software, Universidad Nacional de La Plata, 2001.
- [Bertolami y Centeno 2008] Bertolami, M., Centeno, E., “LEL y Escenarios del Sistema de Gestión de Comercio”, Proyecto de Investigación, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina, 2008.
- [Bertolami y Oliveros 2003] Bertolami, M., Oliveros, A., “Análisis de Puntos Función en la elicitación de requerimientos”, *Proceedings 6th Workshop on Requirements Engineering*, Piracicaba, Brasil, pp. 32-47, 2003.
- [Bertolami y Oliveros 2006] Bertolami, M., Oliveros, A., “SFP: Un Procedimiento de Estimación de Puntos Función de Escenarios”, *Proceedings 9th Workshop on Requirements Engineering*, Rio de Janeiro, Brasil, pp. 101-108, 2006.
- [Bertolami y Oliveros 2008a] Bertolami, M., Oliveros, A., “Evaluación Empírica de la Estimación de Tamaño Funcional de Escenarios”, *11th Workshop of Requirements Engineering*, Barcelona, España, pp. 25-33, sep. 2008.
- [Bertolami y Oliveros 2008b] Bertolami, M., Oliveros, A., “Medición de Puntos Función sobre especificaciones en UML, una evaluación experimental”, *Congreso Argentino de Ciencias de la Computación (CACIC 2008)*, Universidad Nacional de Chilecito, La Rioja, Argentina, 2008.
- [Boehm *et al.* 1998] Boehm, B., Abts, C., Clark, B., Deuhani-Chulani S., *COCOMO II Model Definition Manual*, University of Southern California, Los Angeles, California, 1998.
- [Bundschuh 1998] Bundschuh, M., “Function Point Prognosis”, *Proceedings FESMA 98*, May 1998.
- [Bundschuh y Dekkers 2008] Bundschuh, M., Dekkers, C., *The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement*, Springer-Verlag, Berlin, Heidelberg, 2008.



- [CAI 2006] Focus on Charles Symons, Founder of COSMIC and Creator of the Mark II Function Point, A CAI State of the Practice Interview, *An IT Metrics and Productivity Journal Special Edition*, Computer Aid Inc., Sep. 2006.
- [Cândido y Sanches 2004] Cândido, E., Sanches, R., “Estimating the size of web applications by using a simplified function point method”, *Proceedings WebMedia and LA-Web*, pp. 98-105, 2004.
- [Choi *et al.* 2006] Choi, S., Park, S., Sugumaran, V., “Function Point Extraction Method from Goal and Scenario Based Requirements Text”, NLDB 2006, LNCS 3999, Springer-Verlag, pp. 12 – 24, 2006.
- [Condori 2007] Condori, N., Un procedimiento de medición de tamaño funcional para especificaciones de requisitos, Tesis Doctoral, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2007.
- [COSMIC 2003] COSMIC-FFP, “COSMIC-FFP method approved as an ISO standard”, Press Release issued by Charles Symons, Mar. 2003.
- [Damodaran y Washington 2002] Damodaran, M., Washington, A., “Estimation Using Use Case Points”, *Proceedings ISECON 2002*, v 19, San Antonio, 2002.
- [Dekkers 2003] Dekkers, T., “Extended) Functional size measurement methods are also applicable in enhancement projects”, *IWSM2003: 13th International Workshop on Software Measurement*, Montréal, Canada, 2003.
- [Dekkers y Aguiar 2001] Dekkers, C., Aguiar, M., “Applying Function Point Analysis to Requirements Completeness”, Software Technology Support Center, *CrossTalk The Journal of Defense Software Engineering*, Feb. 2001.
- [Dekkers y Kammelar 2001] Dekkers, T., Kammelar, J., “A Functional Sizing Meta Model”, *Proceedings ESCOM-SCOPE 2001*, London, Apr 2001, FESMA 2001, Heidelberg, May 2001.
- [Desharnais y Abran 2003] Desharnais, J., Abran, A., “Approximation techniques for measuring function points”, *Proceedings 13th International Workshop on Software Measurement (IWSM 2003)*, Montréal (Canada), pp. 270-286, 2003.
- [Doom *et al.* 1998] Doorn, J., Kaplan, G., Hadad, G., Leite, J., “Inspección de Escenarios”, *Proceedings WER'98, Workshop en Engenharia do Requisitos*, Maringá, Brazil, 1998, 57-69.
- [DPO 2007] DPO, Application Experiences, Early & Quick Function Points IFPUG variant, <http://www.dpo.it/en/eqfp/esperienze.htm>, consultado en mayo de 2009.
- [DPO 2008] Early & Quick Function Points for IFPUG method release 3.0 Reference Manual v. 1.1, DPO, Jan. 2008.
- [Ebert *et al.* 2005] Ebert, C., Dumke, R., Bundschuh, M., Schmietendorf, A., *Best Practices in Software Measurement - How to use metrics to improve project and process performance*, Springer-Verlag, Berlin, Heidelberg, 2005.
- [Esteban y Heidanowski 1998] Esteban, N., Heidanowski, A., “LEL y Escenarios de una Estación de Servicio”, caso desarrollado en el curso de Tópicos I en el Magister de Ingeniería de Software, Universidad Nacional de La Plata, Argentina, 1998.
- [Fenton y Neil 1999] Fenton, N., Neil, M., “Software metrics: successes, failures and new directions”, *Journal of Systems and Software*, Vol.47, pp 149 - 157, 1999.

- [Fenton y Pfleeger 1996] Fenton, N., Pfleeger, S., *Software metrics A rigorous and Practical approach*, Second Edition, PWS Publishing Company, 1996.
- [Fetcke *et al.* 1997] Fetcke, T., Abran, A., Nguyen, T., "Mapping the OO-Jacobson Approach into Function Point Analysis", *Proceedings TOOLS '97*, Santa Barbara, CA, pp. 192-202, 1997.
- [Fetcke *et al.* 2001] Fetcke, T., Abran, A., Dumke, R., "A Generalized Representation for Selected Functional Size Measurement Methods", 11th International Workshop on Software Measurement, Montréal, Canada, 2001.
- [Fowler 2003] Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Third Edition, Addison Wesley, 2003.
- [Functional Size Measurement 2008] Functional Size Measurement, [http://www.geocities.com/lbu\\_measure/fpa/fpa.htm](http://www.geocities.com/lbu_measure/fpa/fpa.htm), última actualización 25 Jul. 2008, consultado en noviembre de 2008.
- [García y Gentile 2000] García, O., Gentile, C., LEL y Escenarios de LEL y Escenarios, Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil, Argentina, 2000, disponible en: <http://usuarios.ar.net.com.ar/ogarcia/bibliogr.htm>.
- [Gates 1997] Gates, L. P., How to Use the Software Process Framework, Special Report, CMU/SEI-97-SR-009, Oct. 1997.
- [Gencel *et al.* 2005] Gencel, C., Demirors, O., Yüceer, E., "Utilizing Functional Size Measurement Methods for Real Time Software Systems", *Proceedings 11th IEEE International Software Metrics Symposium (METRICS 2005)*, Como, Italy, Sep. 2005.
- [GeroneSoft 2005] GeroneSoft Code Counter Pro Software, <http://www.geronesoft.com/>, v1.24, Apr. 2005.
- [Gil y Arias 2000] Gil, G., Arias Figueroa, D., Oliveros, A., "Producción del LEL en un Dominio Técnico. Informe de un caso", *Workshop on Requirements Engineering*, Rio de Janeiro, Brasil, pp. 53-69, 2000.
- [Hadad 2008] Hadad, G., Uso de Escenarios en la Derivación de Software, Tesis Doctoral presentada en la Facultad de Ciencias Exactas de la Universidad Nacional de La Plata, 2008.
- [Hadad *et al.* 1997] Hadad, G., Kaplan, G., Oliveros, A., Leite, J., "Construcción de Escenarios a partir del Léxico Extendido del Lenguaje", *26 JAIIO*, Buenos Aires, Argentina, 1997.
- [Hadad *et al.* 1998] Hadad, G., Kaplan, G., Leite, J., "Léxico extendido del lenguaje y escenarios del Meeting Scheduler", Informe técnico #13, Departamento de Investigación, Universidad de Belgrano, Buenos Aires, 1998.
- [Hadad *et al.* 1999] Hadad, G., Kaplan, G., Oliveros, A., Leite, J., "Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos. Aplicación a un caso real", *Revista de Informática Teórica y Aplicada (RITA)*, Brasil, Vol.6, N°1, pp.77-103, 1999.
- [Harput *et al.* 2005] Harput, V., Kaindl, H., Kramer, S., "Extending Function Point Analysis of Object-Oriented Requirements Specifications", *11th IEEE International Software Metrics Symposium (METRICS'05)*, 2005.

- 
- [Heller 1995] Heller, R., "An Introduction to Function Point Analysis", Q/P Management Group Inc, 1995.
- [Hill 2006] Hill, P., "Software Early Lifecycle – Functional Sizing", ISBSG, Jun. 2006, Vol. 9, No. 2, [https://www.softwaretechnews.com/stn\\_view.php?stn\\_id=5&article\\_id=23](https://www.softwaretechnews.com/stn_view.php?stn_id=5&article_id=23), consultado en noviembre de 2008.
- [Hornby 2000] Hornby, A., *Oxford Advanced Learner's Dictionary of Current English*, Oxford University Press, Sixth ed., 2000.
- [Humphrey 1989] Humphrey, Watts S., *Managing the Software Process*, Addison-Wesley Publishing Company, 1989.
- [IEEE 1990] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990, Institute of Electrical and Electronic Engineers, Inc., New York, NY, Dec. 1990.
- [IEEE 1998] IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, 1998.
- [IFPUG 2000] IFPUG, Manual para la Medición de Puntos Función, Versión 4.1.1., AEMES, 2000.
- [IFPUG 2008] IFPUG, Metrics Views, Winter 2008, Vol. 2, Issue 1, [http://www.ifpug.org/members/newsletter/2008MetricViews/MV\\_Winter\\_2008.pdf](http://www.ifpug.org/members/newsletter/2008MetricViews/MV_Winter_2008.pdf), 2008.
- [Iorio *et al.* 2007] Iorio, T., Meli, R., Perna, F., "Early & Quick Function Points® v3.0: enhancements for a Publicly Available Method", *Proceedings Software Measurement European Forum (SMEF) 2007*, pp. 179-198, 2007.
- [ISBSG 2005] ISBSG, "Software Project Estimates - How accurate are they?", 2005, <http://www.isbsg.org>.
- [ISBSG 2006] Estimating Functional Size, <http://www.isbsg.org/isbsg.nsf/weben/EstimatingFunctionalSize.htm>, consultado en noviembre de 2006.
- [ISO 1993] ISO, "International Vocabulary of Basic and General Terms in Metrology", International Organization for Standardization, 1993.
- [ISO 1998] ISO, ISO/IEC 14143-1, Information technology - Software measurement - Functional size. Part 1: Definition of concepts, *International Organization for Standardization*, Geneva, 1998.
- [ISO 2002a] ISO/IEC 15939, Information Technology Software engineering - Software measurement process, *International Organization for Standardization*, 2002.
- [ISO 2002b] ISO, ISO/IEC 14143-4, Information Technology Software Measurement - Functional Size Measurement. Part 4: Reference Model, *International Organization for Standardization*, Geneva, 2002.
- [Jacobson 2003] Jacobson, I., "Use Cases - Yesterday, Today and Tomorrow", Technical Report, *The Rational Edge*, Mar. 2003, disponible en <http://www.therationaledge.com/>.
- [Jacquet y Abran 1997] Jacquet, J., Abran, A., "From Software Metrics to Software Measurement Methods: A Process Model", *3rd Int. Standard Symposium and Forum on Software Engineering Standards (ISESS'97)*, Walnut Creek, USA, 1997.

- 
- [JCGM 2008] Joint Committee for Guides in Metrology (JCGM/WG 1), "Evaluation of measurement data - Guide to the expression of uncertainty in measurement", First edition, Sep. 2008.
- [Jeffery *et al.* 1993] Jeffery, D., Low, G., Barnes, M., "A comparison of function point counting techniques", *IEEE Transactions on Software Engineering*. Vol. 19, Nro. 5, pp. 529-532, 1993.
- [Jones 1995] Jones, C.: "Backfiring: Converting lines of code to function points", *IEEE Computer*, pp. 87-88, Nov. 1995.
- [Jones 1996] Jones, C., *Programming Languages Table*, Release 8.2, Software Productivity Research, Inc., Mar. 1996.
- [Jones 2006] Jones, C., *Strengths and weaknesses of software metrics*, Software Productivity Research, Version 5, Mar. 2006.
- [Jones 2007] Jones, C., *Estimating Software Costs*, 2<sup>nd</sup> Ed., McGraw-Hill, 2007.
- [Jones 2008a] Jones, C., *Applied Software Measurement. Global Analysis of Productivity and Quality*, 3<sup>rd</sup> Ed., McGraw-Hill, 2008.
- [Jones 2008b] Jones, C., "A New Business Model for Function Point Metrics", Version 7.0, May 7, 2008.
- [Kaplan *et al.* 2000] Kaplan, G., Hadad, G., Doorn, J., Leite, J.. "Inspección del Léxico Extendido del Lenguaje", *Proceedings WER'00, Workshop de Engenharia de Requisitos*, Río de Janeiro, Brasil, pp.70-91, 2000.
- [Karner 1993] Karner, G., "Resource Estimation for Objectory Projects", *Objective Systems SF AB*, 1993.
- [Kemerer 1993] Kemerer, C., "Reliability of Function Points Measurement: A Field Experiment", *Communications of the ACM*, Vol. 36, Nro. 2, pp. 85-97, Feb. 1993.
- [Kitchenham *et al.* 1994] Kitchenham, B., Linkman, S., Law, D., "Critical review of quantitative assessment", *Software Engineering Journal*, 9(2), pp.43-53, 1994.
- [Kitchenham *et al.* 1995] Kitchenham, B., Pickard, L., Pfleeger, S., "Case studies for method and tool evaluation", *IEEE Software*, pp. 52-62, 1995.
- [Kusumoto *et al.* 2000] Kusumoto, S., Inoue, K., Kasimoto, T., Suzuki, A., Yuura, K., Tsuda, M., "Function Point Measurement for Object-Oriented Requirements Specification", *Proceedings Annual International Computer Software and Applications Conference*, IEEE, pp. 543-548, 2000.
- [Kusumoto *et al.* 2004] Kusumoto S., Matukawa F., Inoue K., Hanabusa S., Maegawa Y., "Estimating Effort by Use Case Points: Method, Tool and Case Study", *Proceedings 10th IEEE International Symposium on Software Metrics (METRICS'04)*, Chicago, IL (USA), pp. 292-299, 2004.
- [Laird y Brennan 2006] Laird, L., Brennan, C., *Software Measurement and Estimation: A Practical Approach*, New York, John Wiley & Sons, 2006.
- [Lavazza y Valetto] Lavazza, L., Valetto, G., "Requirements-based Estimation of Change Costs", *International Journal of Empirical Software Engineering*, Vol. 5, Nro. 3, p.p. 229 - 243, Nov. 2000.

- 
- [Leite *et al.* 1996] Leite, J., Oliveros, A., Rossi, G., Balaguer, F., Hadad, G., Kaplan, G., Maiorana, V., “Léxico Extendido del Lenguaje y Escenarios del sistema nacional para la obtención de pasaportes”, Informe técnico #7, Universidad de Belgrano, Buenos Aires, 1996.
- [Leite *et al.* 1997] Leite J., Rossi G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, *Proceedings RE 97' International Symposium on Requirements Engineering*, IEEE, Vol. 2, Nro. 4, pp. 184-198, 1997.
- [Leite *et al.* 2000] Leite, J., Hadad, G., Doorn, J., Kaplan, G., “A Scenario Construction Process”, *Requirements Engineering Journal*, 2000.
- [Lelli y Meli 2005] Lelli, M., Meli, R., "From narrative user requirements to Function Point", *Proceeding SMEF 2005*, pp. 235-246, 2005.
- [Longstreet 2000] Longstreet, D., OO and Function Points (Draft), Longstreet Consulting Inc., [www.softwaremetrics.com](http://www.softwaremetrics.com), consultado en octubre de 2000.
- [Longstreet 2001] Longstreet, D., “Use cases and function points”, Longstreet Consulting Inc, 2001.
- [Longstreet 2003] Longstreet, D., Function Points Analysis Training Course, Longstreet Consulting Inc., [www.SoftwareMetrics.com](http://www.SoftwareMetrics.com), 2003.
- [MacDonell 1994] MacDonell, S., “Comparative review of functional complexity assessment methods for effort estimation”, *Software Engineering Journal*, pp. 107-116, May 1994.
- [Mauco *et al.* 1997] Mauco, V., Ridao, M., del Fresno, M., Rivero, L., Doorn, J., “Proyecto Sistema de Planes de Ahorro”, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buneos Aires, Argentina, 1997.
- [McAndrews 1993] McAndrews, D., *Establishing a Software Measurement Process*, Technical Report, CMU/SEI-93-TR-16, ESC-TR-93-193, Jul. 1993.
- [McConnell 1998] McConnell, S., *Software Project Survival Guide*, Microsoft Press, 1998.
- [McConnell 2006] McConnell, S., *Software estimation. Demystifying the Black Art*, Microsoft Press, 2006.
- [McGarry *et al.* 2001] McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., Hall, F., *Practical Software Measurement: Objective Information for Decision Makers*, Addison-Wesley Professional, 2001.
- [Meli 1997] Meli, R., “Early and Extended Function Point: A New Method for Function Points Estimation”, *IFPUG-Fall Conference*, Scottsdale, Arizona, USA, Sep. 1997.
- [Meli 1998] Meli, R., “Functional Metrics: Problems And Possible Solutions”, *Proceedings FESMA'98*, Belgium, pp. 503-514, May 1998.
- [Meli 2000] Meli, R., "Functional and Technical Software Measurement: Conflict or Integration?", *3rd. European Software Measurement Conference (FESMA'00)*, Madrid, Spain, 2000.

- 
- [Meli *et al.* 2000] Meli, R., Abran, A., Ho, V., Oligny, S., “On the applicability of COSMIC-FFP for measuring software throughout its life cycle”, *ESCOM-SCOPE 2000*, Munich, Apr. 18-20, 2000.
- [Meli y Santillo 1999] Meli, R. and Santillo, L., “Function Point Estimation Methods: a Comparative Overview”, *FESMA 99*, Oct. 6-8, Amsterdam, 1999.
- [Melo *et al.* s.f.] Melo, B., Ferreira, C., Chaves D., Osório, T., Sistema de biblioteca. Sieger Solutions, <http://www.er.les.inf.puc-rio.br/biblioteca/welcome.htm>, consultado en diciembre de 2003.
- [Merriam-Webster 2009] *Merriam-Webster Online Dictionary*, 2009, <http://www.merriam-webster.com/dictionary/functionality>, consultado el 29 de marzo de 2009.
- [Morasca 2001] Morasca, S., “Software Measurement”, *Handbook of Software Engineering and Knowledge Engineering*, Vol. I, World Scientific Pub. Co., Dec. 2001.
- [Morris 2001] Morris, P., Functional Size Metrics, *International Quality and Productivity Centre*, [www.totalmetrics.com](http://www.totalmetrics.com), 2001, consultado en agosto de 2003.
- [Morris 2004] Morris, P., “Levels of Function Point Counting”, Version 1.3, *Total Metrics*, 2004.
- [Morris y Desharnais 2001] Morris, P., Desharnais, J., “Function Point Analysis - Validating The Result”, Version 1.4, *Total Metrics Pty Ltd and SELAM*, Jul. 2001.
- [Morrison y Seaver 2007] Morrison, N., Seaver, D., “Utilizing Functional Size on Enhancement Projects”, PSM Users’ Group, disponible en [http://www.psmc.com/UG2007/Presentations/14\\_Utilizing\\_Functional\\_Size\\_Morrison.pdf](http://www.psmc.com/UG2007/Presentations/14_Utilizing_Functional_Size_Morrison.pdf), consultado en mayo de 2009.
- [NESMA 2002] NESMA, “Functional Sizing in Contemporary Environments. Introduction of a Functional Sizing Reference Model”, *NESMA Workgroup NT*, Version 1.0, Oct. 2002.
- [NESMA 2004] NESMA, Early Function Point Counting, <http://www.nesma.nl/english/earlyfpa.htm>, consultado en marzo de 2005.
- [NESMA 2008] “FPA according to NESMA and IFPUG; the present situation”, ver. Jul. 2008.
- [Oligny *et al.* 1998] Oligny, S., Abran, A., Desharnais, J.-M., Morris, P., “Functional Size of Real-Time Software: Overview of Field Tests”, *Proceedings 13th International Forum on COCOMO and Software Cost Modeling*, Los Angeles, CA, 1998.
- [Olson *et al.* 1994] Olson, T., Reizer, N., Over, J., *A Software Process Framework for the SEI Capability Maturity Model*, Handbook, CMU/SEI-94-HB-01, Sep. 1994.
- [Olson *et al.* 1993] Olson, T., Gates, L. P., Mullaney, J., Over, J., Reizer, N., Kellner, M., Phillips, R., DiGennaro, S., *A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level*, Special Report, CMU/SEI-93-SR-007, Jun. 1993.

- 
- [Oriana *et al.* 2006] Oriana, G., Olinik, R., Ritter, P., “LEL y Escenarios del Sistema de Notificaciones a los Clientes de un Banco”, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina, 2006.
- [Osterweil 1987] Osterweil, L., Software Processes are Software Too, *Proceedings 9th. Intern. Conf. Software Engineering*, 2-13, IEEE Computer Society, 1987.
- [Peters 1999] Peters, K., Software Project Estimation, Software Productivity Center Inc., 1999.
- [Potts 1998] Potts, C., Antón, A., *A Representational Framework for Scenarios of System Use*, College of Computing College of Engineering, Georgia Institute of Technology, North Carolina State University, Engineering Graduate Research Center Atlanta, 1998.
- [Pressman 1993] Pressman, R., *Ingeniería del Software. Un enfoque práctico*, 3<sup>a</sup> Ed., McGraw-Hill, 1993.
- [Pressman 2001] Pressman, R., *Software Engineering: A Practitioner's Approach*, 5<sup>th</sup> Ed., McGraw-Hill, 2001.
- [Rational Corp. 1997] Rational Corp, *UML Summary*, Version 1.0, Rational Software Corporation, Sep. 1997.
- [Ritter *et al.* 2009] Ritter, P., Bertolami, M., Oriana, G., “SFP Tool: una Herramienta para Medir Puntos Función”, *Workshop de Investigadores de Ciencias de la Computación (WICC 2009)*, San Juan, Argentina, 2009.
- [Rivero *et al.* 1998] Rivero L, Doorn J, Del Fresno M, Mauco V, Ridao M, Leonardi C. “Una Estrategia de Analisis Orientada a Objetos basada en Escenarios: Aplicacion en un Caso Real”, *Proceedings of WER'98: Workshop en Engenharia do Requisitos*, Maringa, Brazil, pp. 79-90, 1998.
- [Rozum 1994] Rozum, J., “Defining and Understanding Software Measurement Data”, *Proc. of the 5th Annual Conference on Applications of Software Measurement*, Nov. 1994.
- [Rule 2001] Rule, G., “The Importance of the Size of Software Requirements”, Software Measurement Services, *NASSCOM Conference*, India, 2001.
- [Rumbaugh *et al.* 1999] Rumbaugh, J., Jacobson, I., Booch, G., “The Unified Modeling Language Reference Manual”, *Addison-Wesley*, 1999.
- [s.a. 1997] “LEL y Escenarios de un Sistema de Crédito Hipotecario”, Caso de estudio desarrollado en el marco del Magister en Ingeniería de Software, 1997.
- [Santillo *et al.* 2005] Santillo, L., Conte, M., Meli, R., “Early & Quick Function Point: Sizing More with Less”, *11th IEEE International Software Metrics Symposium (METRICS 2005)*, 2005.
- [Santillo y Meli 1998] Santillo, L., Meli, R., “Early Function Points: some practical experiences of use”, *ESCOM-ENCRESS 98*, Rome, May 1998.
- [Seaver 2000] Seaver, D., FAST Function Points, Fidelity Investments, 2000, disponible en [http://sunset.usc.edu/Activities/oct24-27-00/Presentations/Seaver\\_FAST%20Function%20Points.pdf](http://sunset.usc.edu/Activities/oct24-27-00/Presentations/Seaver_FAST%20Function%20Points.pdf), consultado en marzo de 2006.
- [SGC 2005] Sistema de Gestión Comercio [programa], Versión 1.0, Departamento de Informática, Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina, 2005.

- 
- [SPP 2002] Sistema de Presentación de Programas SPP [programa], Versión 1.1d, Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, Argentina, 2002-2005.
- [St-Pierre *et al.* 1997] St-Pierre, D., Maya, M., Abran, A., Desharnais, J. M., 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, Sep. 1997.
- [Sutcliffe 1997] Sutcliffe, A., “Workshop Exploring Scenarios in Requirements Engineering”, Centre for HCI Design, City University, Northampton Square, 1997.
- [Symons 1991] Symons, C., *Software Sizing and Estimating Mk II FPA*, John Wiley & Sons, England, 1991.
- [Tavares *et al.* 2002] Tavares, H., Carvalho, A., Castro, J., “Medição de pontos por função a partir da especificação de requisitos”, *Proceedings V Workshop on Requirements Engineering*, Valencia, España, pp. 278-298, 2002.
- [Tichenor 1997] Tichenor, C., “The Internal Revenue Service function point analysis program: a brief”, *COMPSAC '97 - 21st International Computer Software and Applications Conference*, pp. 591 -592, 1997.
- [Tichenor 2008] Tichenor, C., <http://www.ifpug.org/discus/messages/1779/10228.html?1227847628>, Posted on Thursday, November 27, 2008.
- [Torres y Calatrava 2003] Torres, I., Calatrava, F., “Automatic Function Points Counting on Conceptual Models”, Whitepaper, CARE Technologies, <http://www.caret.com/html/whitepapers.html>, 2003.
- [UKSMA 1998] UKSMA, *MkII Function Point Analysis Counting Practices Manual*, Version 1.3.1, United Kingdom Software Metrics Association, 1998.
- [STSC 2000] Guidelines for Successful Acquisition and Management of Software Intensive Systems, Version 3.0 May 2000, Chapter 13, U.S. Air Force Software Technology Support Center, [http://www.stsc.hill.af.mil/resources/tech\\_docs/gsam3.html](http://www.stsc.hill.af.mil/resources/tech_docs/gsam3.html).
- [Weidenhaupt *et al.* 1998] Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., “Scenario Usage in System Development: A Report on Current Practice”, *IEEE Software*, March, 1998, RWTH Aachen, Informatik V, Germany, 1998.
- [Yin 2003] Yin, R., *Case study research. Design and methods*, 3<sup>rd</sup> Ed., London, Sage Publications, 2003.



## Apéndice A

### **Método IFPUG FPA Versión 4.1.1.**

El presente resumen fue extraído del Manual para la Medición de Puntos Función Versión 4.1.1 [IFPUG 2000]. El Análisis de Puntos Función (FPA – Function Points Análisis) mide el software cuantificando la funcionalidad suministrada al usuario, basándose principalmente en el diseño lógico. Los objetivos del FPA son:

1. Medir la funcionalidad que el usuario pide y recibe.
2. Medir el desarrollo y mantenimiento del software independientemente de la tecnología usada para su implementación.

#### **A.1. Punto de vista del usuario**

El punto de vista del usuario representa una descripción formal de las necesidades del negocio en el lenguaje del usuario. El punto de vista del usuario:

- Es una descripción de la funciones del negocio.
- Es aprobado por el usuario.
- Puede ser utilizado para el conteo de Puntos Función (FP – Function Points).
- Puede variar su forma física (por ejemplo, catálogos de transacciones, documentos de requerimientos, especificaciones externas, especificaciones detalladas, manual de usuario).

#### **A.2. Procedimiento de Medición de Puntos Función**

El Manual para la Medición de Puntos Función especifica el siguiente procedimiento.

1. Determinar el tipo de medición de puntos función.
2. Identificar el alcance de la medición y el límite de la aplicación.
3. Contar las funciones de datos.
4. Contar las funciones transaccionales.
5. Determinar los Puntos Función no Ajustados
6. Determinar el valor del factor de ajuste.
7. Calcular la cuenta de Puntos Función Ajustados.

Para cumplir con el estándar ISO 14143-1, los pasos 8 y 9 del procedimiento no deben ser considerados.

##### **A.2.1. Determinar el Tipo de medición de Puntos Función**

Existen tres tipos de medición:

- medición de FP de un proyecto de desarrollo: este tipo de cuenta está asociada con un nuevo desarrollo.
- medición de FP de un proyecto de mejora: este tipo de cuenta está asociada con un proyecto de mejoramiento de un sistema existente.

- medición de FP de una aplicación: este tipo de cuenta está asociada con una aplicación instalada.

### **A.2.2. Identificar el Alcance de la medición y el Límite de la aplicación**

La definición del *propósito de la medición* consiste en dar respuesta a un problema específico. El propósito determina el tipo de medición de FP y el alcance de la medición. Infiere en la localización del límite entre el software a medir y otro software periférico.

El *alcance de la medición* define la funcionalidad que será incluida en una medición de FP concreta. Identifica qué funciones serán incluidas en la medición de FP para dar respuestas relacionadas con el *propósito de la medición*.

La medición de FP de un desarrollo incluye todas las funciones impactadas (construidas o personalizadas) por las actividades del proyecto.

El *límite de la aplicación* establece el borde conceptual entre el software a medir y el usuario. Al establecer el límite del sistema es importante tener en cuenta que debe fijarse desde la perspectiva del negocio y no sobre la base de consideraciones físicas o técnicas.

### **A.2.3. Calcular los FP no ajustados**

La *Medida en FP no Ajustados* refleja la funcionalidad específica medible suministrada al usuario por el proyecto o aplicación.

La funcionalidad específica del usuario de la aplicación se determina en términos de *qué* se entrega con la aplicación y no *cómo* se entrega. Se miden solamente los componentes solicitados y definidos por el usuario.

A nivel conceptual, el FPA ayuda a definir dos niveles de abstracción de datos [Longstreet 2003]:

*Datos en reposo*: los mantenidos por la aplicación que se está midiendo se clasifican como *Archivos Lógicos Internos* (ILF – Internal Logical File) y los mantenidos por otra aplicación son los *Archivos de Interfaz Externos* (EIF – External Interface File).

*Datos en movimiento*: son manejados a través de *funciones transaccionales* o *transacciones*. Todas las aplicaciones de software incluyen procesos elementales (o transacciones) para mover datos. Los procesos elementales que mueven datos desde afuera hacia adentro del límite se denominan *Entradas Externas* (EI – External Input), las que retornan datos desde un archivo hacia afuera del límite se conocen como *Salidas Externas* (EO – External Output) o *Consultas Externas* (EQ – External Inquiry).

El cálculo de los FP no Ajustados se basa en dos tipos de funciones: de *datos* y *transaccionales*.

### **A.2.4. Medir las funciones de datos**

Las *funciones de datos* representan la funcionalidad provista al usuario para satisfacer los requerimientos de datos externos e internos.

Un ILF es un grupo de datos relacionados lógicamente o información de control, identificable por el usuario, mantenido dentro del límite de la aplicación. El propósito principal de un ILF es almacenar datos mantenidos a través de uno o más *procesos elementales* de la aplicación que se está midiendo.

Un EIF es un grupo de datos relacionados lógicamente o información de control, identificable por el usuario, referenciado por la aplicación, pero mantenido dentro del

límite de otra aplicación. El propósito principal de un EIF es almacenar datos referenciados mediante uno o más procesos elementales dentro del límite de la aplicación que se está midiendo.

#### **A.2.4.1. Definiciones**

*Información de control:* son los datos que influyen sobre un proceso elemental de la aplicación que se está midiendo. Especifica qué, cuándo o cómo los datos van a ser procesados.

*Identificable por el usuario:* se refiere a los requerimientos definidos para procesos o grupos de datos que están acordados y entendidos por usuarios y desarrolladores.

*Mantenido:* se refiere a la capacidad de modificar datos a través de un proceso elemental.

*Proceso elemental:* es la unidad de actividad más pequeña que es significativa para el usuario. Debe ser autosuficiente y dejar la aplicación que está siendo contada y que soporta el proceso del negocio, en estado consistente.

#### **A.2.4.2. Reglas de medición de ILF y EIF**

El procedimiento de medición de las funciones de datos requiere dos actividades:

- Identificar los ILF y EIF.
- Determinar la complejidad de los ILF y EIF y su contribución a la cuenta de FP.

##### **A.2.4.2.1. Reglas de identificación de ILF**

Para que una información se pueda considerar ILF, debe cumplir las siguientes reglas:

- El grupo de datos o información de control es lógico e identificable por el usuario.
- El grupo de datos es mantenido por un proceso elemental dentro del límite de la aplicación que se está midiendo.

##### **A.2.4.2.2. Reglas de Identificación de EIF**

Para que una información se pueda considerar EIF, debe cumplir las siguientes reglas:

- El grupo de datos o información de control es lógico e identificable por el usuario.
- El grupo de datos es referenciado por la aplicación que se está midiendo y es externo a ella.
- El grupo de datos no está mantenido por la aplicación que se está midiendo
- El grupo de datos está mantenido en un ILF de otra aplicación.

#### **A.2.4.3. Reglas de complejidad y contribución**

El número de ILF y EIF y su complejidad determina su contribución a la cuenta de FP no ajustados. La complejidad funcional depende del número de *Tipos de Datos Elementales* (DET – Data Element Type) y *Tipos de Registros Elementales* (RET – Record Element Type) asociados a un ILF o EIF.

##### **A.2.4.3.1. Definición de DET, RET y reglas para contabilizarlos**

**DET:** es un campo único, no repetido, reconocible por el usuario.

**RET:** es un subgrupo de datos elementales reconocible por el usuario, dentro de un ILF o EIF.

**Reglas para contar DET**

- Contar 1 DET por cada campo único, no repetido, reconocible por el usuario y mantenido o recuperado desde un ILF o EIF mediante la ejecución de un proceso elemental.
- Cuando dos aplicaciones mantienen o referencian el mismo ILF/EIF pero cada una mantiene o referencia DET diferentes, contar sólo los DET que están siendo utilizados por cada aplicación.
- Contar 1 DET por cada parte de dato requerido por el usuario para establecer una relación con otro ILF o EIF.

**Reglas para contar RET**

- Contar un RET por cada subgrupo del ILF o EIF.
- Si no existen subgrupos, contar un único RET.

**Complejidad y Contribución de ILF**

RET	DET		
	1 a 19	20 - 50	51 o más
1	Baja (7)	Baja (7)	Media (10)
2 a 5	Baja (7)	Media (10)	Alta (15)
6 o más	Media (10)	Alta (15)	Alta (15)

**Complejidad y Contribución de EIF**

RET	DET		
	1 a 19	20 - 50	51 o más
1	Baja (5)	Baja (5)	Media (7)
2 a 5	Baja (5)	Media (7)	Alta (10)
6 o más	Media (7)	Alta (10)	Alta (10)

**r las funciones transaccionales**

Las *funciones transaccionales* representan la funcionalidad suministrada al usuario para el procesamiento de información.

Una EI es un proceso elemental que procesa datos o información de control que viene desde fuera del límite de la aplicación. El propósito principal de una EI es mantener uno o más ILF y/o modificar el comportamiento del sistema.

Una EO es un proceso elemental que envía datos o información de control fuera del límite de la aplicación. El propósito principal de una EO es presentar información al usuario mediante una lógica de proceso distinta a, o conjuntamente con la recuperación de datos o información de control. La lógica del proceso debe contener al menos una fórmula matemática o cálculo o crear datos derivados. Una salida externa puede mantener uno o más ILF y/o modificar el comportamiento del sistema.

Una EQ es un proceso elemental que envía datos o información de control fuera del límite de la aplicación. El propósito principal de una EQ es presentar información al usuario mediante la recuperación de datos o información de control. La lógica del proceso no contiene fórmulas matemáticas ni cálculos y no crea datos derivados. No se mantiene ningún ILF durante el proceso ni se modifica el comportamiento del sistema.

**A.2.5.1. Reglas de identificación de procesos elementales**

Un proceso elemental debe cumplir:

- es la menor unidad de actividad que tiene sentido para el usuario.
- es autosuficiente y deja la aplicación en estado consistente.

### A.2.5.2. Vocabulario típico

La siguiente lista de palabras extraída (y reducida) de Longstreet (2003) puede ayudar a identificar términos asociados con:

#### **Entradas Externas**

<i>Agregar</i>	<i>Insertar (agregar y cambiar)</i>
<i>Activar</i>	<i>Mantener (agregar, cambiar o borrar)</i>
<i>Cancelar</i>	<i>Modificar (cambiar)</i>
<i>Cambiar</i>	<i>Remover (borrar)</i>
<i>Convertir (cambiar)</i>	<i>Reemplazar (cambiar)</i>
<i>Crear (agregar)</i>	<i>Save (agregar, cambiar o borrar)</i>
<i>Borrar</i>	<i>Almacenar (agregar)</i>
<i>Editar (cambiar)</i>	<i>Actualizar (agregar, cambiar o borrar)</i>

**Salidas Externas** (algunas palabras son similares a las usadas para *Consultas Externas*)

<i>Visualizar</i>	<i>Consultar</i>
<i>Obtener</i>	<i>Retornar</i>
<i>Salida</i>	<i>Buscar</i>
<i>Imprimir</i>	<i>Seleccionar</i>

**Consultas Externas** (algunas palabras son similares a las usadas para *Salidas Externas*)

<i>Visualizar</i>	<i>Imprimir</i>
<i>Extraer</i>	<i>Consultar</i>
<i>Buscar</i>	<i>Seleccionar</i>
<i>Obtener</i>	<i>Mostrar</i>
<i>Listar</i>	<i>Ver</i>
<i>Salida</i>	

#### A.2.5.2.1. Propósito principal de un proceso elemental

- **EI:** mantener uno más ILFs o alterar el comportamiento del sistema.
- **EO:** presentar información al usuario. Se presentan datos calculados o derivados, se actualiza uno o más ILFs o se altera la información del sistema.
- **EQ:** presentar información al usuario. Se presentan datos recuperados de uno o más ILFs o EIFs.

#### A.2.5.2.2. Complejidad y contribución

Asignar a cada EI, EO y EQ identificada una complejidad funcional basada en el número de *Tipos de Archivos Referenciados* (FTR – File Type Referenced) y DET. Un FTR es un tipo de archivo referenciado (ILF leído o mantenido o un EIF leído) por una función transaccional.

#### Reglas de medición para las EI

Por cada EI deben cumplirse las reglas:

- Los datos o la información de control que entran en la aplicación.
- Al menos se actualiza un ILF si los datos introducidos no contienen información de control que altere el funcionamiento del sistema.

- Una de las tres siguientes reglas debe aplicarse:
  - o El proceso lógico es único entre los procesos lógicos realizados por otras EI.
  - o El conjunto de datos elementales es diferente de los conjuntos identificados por otras EI.
  - o Los ILF o EIF usados son diferentes de los archivos utilizados por otras EI.

**Reglas de complejidad y contribución para las EI**

**FTR**

- Contar un FTR por cada ILF mantenido por la EI.
- Contar un FTR por cada ILF o EIF leído por la EI.
- Contar un FTR por cada ILF mantenido y leído por la EI

**DET**

- Contar un DET por cada campo reconocible por el usuario que entra o sale del límite y es requerido para completar la EI.
- No contar campos que son recuperados o derivados por el sistema y almacenados en un ILF durante un proceso elemental si los campos no cruzaron el límite.
- Contar un DET por la capacidad de enviar un mensaje de respuesta del sistema fuera del límite para indicar que ha ocurrido un error durante el proceso, confirmar que se ha completado o verificar que el proceso debería continuar.
- Contar un DET por la posibilidad de especificar una acción a realizar, incluso si hay múltiples métodos para invocar el mismo proceso lógico.

**Complejidad y contribución para EI**

FTR	DET		
	1- 4	5 - 15	16 o más
0 - 1	Baja (3)	Baja (3)	Media (4)
2	Baja (3)	Media (4)	Alta (6)
3 o más	Media (4)	Alta (6)	Alta (6)

**Reglas de medición compartidas para las EO y EQ**

Por cada proceso elemental que tiene como finalidad presentar información al usuario deben cumplirse las reglas:

- La función envía datos o información de control fuera del límite de la aplicación.
- Una de las tres siguientes reglas debe aplicarse:
  - o El proceso lógico es único entre los procesos lógicos realizados por otras EO o EQ.
  - o El conjunto de datos elementales es diferente de los conjuntos identificados por otras EO o EQ.
  - o Los ILF o EIF usados son diferentes de los archivos utilizados por otras EO o EQ.

**Reglas de medición adicionales para las EO**

Además de las anteriores compartidas con las EQ, debe aplicarse una de las reglas:

- La lógica del proceso elemental contiene al menos un cálculo o una fórmula matemática.
- La lógica del proceso elemental crea datos derivados.
- La lógica del proceso elemental mantiene al menos un ILF.
- La lógica del proceso elemental altera el funcionamiento del sistema.

#### **Reglas de medición adicionales para las EQ**

Además de las anteriores compartidas con las EO, deben aplicarse todas las siguientes reglas:

- La lógica del proceso elemental recupera datos o información de control de un ILF o EIF.
- La lógica del proceso elemental no contiene un cálculo o fórmula matemática.
- La lógica del proceso elemental no crea datos derivados.
- La lógica del proceso elemental no mantiene un ILF.
- La lógica del proceso elemental no altera el funcionamiento del sistema.

#### **Reglas de complejidad y contribución comunes para las transacciones EO y EQ**

Para determinar la complejidad de una EO o EQ:

##### **FTR**

- Contar un FTR por cada ILF o EIF leído por el proceso elemental.

##### **DET**

- Contar un DET por cada campo reconocible por el usuario que entra a la aplicación y es requerido para especificar cuándo, qué y cómo los datos son recuperados o generados por el proceso elemental.
- Contar un DET por cada campo no repetido, reconocible por el usuario que sale del límite de la aplicación.
- Si un DET entra y sale se debe contar una vez.
- Contar un DET por la capacidad de enviar un mensaje de respuesta del sistema fuera del límite para indicar que ha ocurrido un error durante el proceso, confirmar que se ha completado o verificar que el proceso debería continuar.
- Contar un DET por la posibilidad de especificar una acción a realizar, incluso si hay múltiples métodos para invocar el mismo proceso lógico.
- No contar campos que son recuperados o derivados por el sistema y almacenados en un ILF durante un proceso elemental si los campos no cruzaron el límite.

#### **Reglas adicionales para las EO**

Además de las reglas enunciadas deben agregarse:

- Contar un FTR por cada ILF mantenido por el proceso elemental.
- Contar un FTR por cada ILF leído y mantenido por el proceso elemental.

**Sugerencias**

- Una EO o EQ puede ser disparada por un proceso incluido dentro del límite de la aplicación.
- Los datos derivados de una EO no tienen por qué mostrarse en la salida.

**Complejidad y contribución para EO**

FTR	DET		
	1 - 5	6 - 19	20 o más
0 - 1	Baja (4)	Baja (4)	Media (5)
2 o 3	Baja (4)	Media (5)	Alta (7)
4 o más	Media (5)	Alta (7)	Alta (7)

**Complejidad y contribución para EQ**

FTR	DET		
	1 - 5	6 - 19	20 o más
0 - 1	Baja (3)	Baja (3)	Media (4)
2 o 3	Baja (3)	Media (4)	Alta (6)
4 o más	Media (4)	Alta (6)	Alta (6)

**A.2.6. Calcular la cuenta de FP no ajustados**

El valor en *Puntos Función no Ajustados* (UFP – Unadjusted Function Points) se calcula sumando las contribuciones de cada tipo de elemento. En la siguiente fórmula, cada tipo de elemento  $i$  con un valor de complejidad  $j$  es asociado con un peso  $W_{ij}$ . El valor UFP se obtiene a partir del producto de la cantidad de tipos de elementos y sus respectivos pesos mediante la siguiente fórmula [Morasca 2001].

$$UFP = \sum_{i \in 1..5} \sum_{j \in 1..3} \#TipoElemento_{i,j} W_{i,j}$$



## Apéndice B

### Un marco para especificar procesos software

El Software Process Framework (SPF) propuesto por Olson *et al.* (1993) es un modelo estandarizado que está basado en el Capability Maturity Model (CMM) del Software Engineering Institute (SEI). El propósito del SFP es proveer los lineamientos para mejorar los procesos software de manera que sean consistentes con el CMM [Olson *et al.* 1993], [Olson *et al.* 1994], [Gates 1997]. Los dos usos principales del SPF son:

- analizar y revisar procesos software para verificar o asegurar que son consistentes con el CMM (en el contexto de la verificación de procesos).
- diseñar procesos software que sean consistentes con el CMM (en el contexto del mejoramiento de procesos software).

#### B.1. El marco operativo del SPF

Para facilitar y mejorar las operaciones, una organización debería definir, documentar y comunicar sus políticas, estándares, procesos, procedimientos, entrenamiento y herramientas a sus usuarios. Esos elementos operativos y sus relaciones constituyen el “marco operativo” (operational framework) (Figura B.1), que sirve de guía para el desarrollo de los productos. Este tipo de definición apunta hacia la forma en que trabajan u operan los diversos componentes del proceso. Es bastante frecuente que la información de políticas, estándares, procesos, procedimientos y entrenamiento estén agrupados y aún mezclados en un mismo documento, lo que resulta poco adecuado y confuso para el uso. El uso del marco operativo favorece el reuso de la documentación a largo plazo.

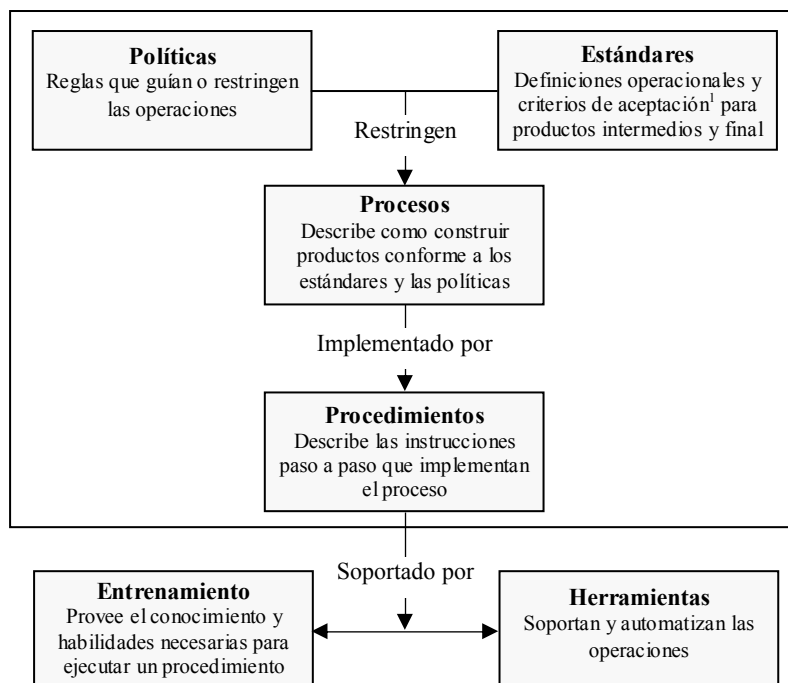


Figura B.1. Marco operativo, componentes y definiciones [Olson *et al.* 1993]

<sup>1</sup> criterios que un sistema o componente debe satisfacer para ser aceptado por el usuario, cliente o una entidad autorizada [IEEE 1990].

**Relación entre los componentes del marco operativo.** Los componentes del marco operativo están relacionados entre sí. Las políticas guían y restringen los procesos. Por ejemplo, una política puede requerir que un proceso se haya completado dentro de un período de tiempo. Los estándares también guían y restringen los procesos. Un estándar para un producto puede restringir la salida de un proceso estableciendo la estructura o criterio de aceptación para esa salida. Un estándar para un proceso puede establecer operativamente qué entrada o criterio de salida se requiere para el proceso. Además los procesos son implementados por procedimientos. Los procedimientos son la descripción paso a paso de la implementación de las partes de un proceso. Las políticas, estándares, procesos y procedimientos son soportados por entrenamiento y herramientas. El entrenamiento provee el conocimiento y las habilidades requeridas para ejecutar el proceso y los procedimientos de acuerdo a las políticas y los estándares. Por último, las herramientas brindan el soporte para realizar el proceso.

**Beneficios asociados al uso del marco operativo.** El uso del SPF permite organizar los distintos tipos de información de un proceso. Las descripciones de los elementos del proceso deben ser simples y claras, el excesivo nivel de detalle puede transformarlos en poco comprensibles e inútiles. El marco operativo ayuda a [Olson *et al.* 1993], [Olson *et al.* 1994]:

- *Separar la información según su propósito:* agrupar la información de acuerdo a su utilidad. Por ejemplo, la información relativa a los estándares que restringen un proceso debe ser accesible sin necesidad de detenerse en otra información.
- *Identificar y usar sólo la información relevante en cada parte:* cada parte del marco operativo debe incluir sólo la información relevante. Por ejemplo, la información relativa a los procesos debe incluir la descripción para construir los productos conforme a los estándares y las políticas, pero no el procedimiento paso a paso del para realizarlo.
- *Gestionar los cambios y mejoras:* la administración de los cambios y mejoras a una parte del marco operativo es facilitada porque la información está bien definida. Por ejemplo, los procesos probablemente no necesiten cambios si se modifica alguno de sus procedimientos.
- *Gestionar y mejorar la comunicación:* la comunicación es facilitada porque los usuarios saben donde buscar cierto tipo de información y conocen las relaciones entre la información. Como los cambios están acotados a la parte operativa se necesita menos comunicación y sólo se deben gestionar y comunicar los cambios relevantes.

### B.2. Plantillas del SPF

El SPF provee plantillas (*templates*) que facilitan la captura de la información necesaria para la mayoría de las representaciones de procesos. La Figura B.2 presenta un fragmento de la plantilla para especificar una actividad.

**Activity:**

**Purpose** Why is this activity performed?

**Performed by** Who is responsible for performing this activity?

Name or ID of Agent

**Entry Criteria** When can this activity begin?

State or Condition	From Activity	[and] [or]

Figura B.2. Ejemplo de plantilla del SFP



## Apéndice C

### Formularios

#### C.1. Plantillas de formularios

<b>Formulario 1</b>	<b>SFP</b> <b>Planilla de episodios</b>		
<b>TÍTULO DE LA APLICACIÓN:</b>			
<b>AUTOR:</b>		<b>FECHA:</b>	
Escenario	ID	Episodio	Tipo

<b>Formulario 2</b>	<b>SFP</b> <b>Complejidad de los episodios</b>		
<b>TÍTULO DE LA APLICACIÓN:</b>			
<b>AUTOR:</b>		<b>REVISOR:</b>	
		<b>FECHA DE REVISIÓN:</b>	
		<b>Nº REVISIÓN:</b>	
<b>ESCENARIO:</b>			
<b>ID:</b>		<b>TIPO:</b>	
<b>RTR</b>		<b>DET</b>	
<b>TOTAL</b>		<b>TOTAL</b>	
<b>Complejidad</b>			

<b>Formulario 3</b>	<b>SFP</b> <b>Complejidad de los recursos</b>		
<b>TÍTULO DE LA APLICACIÓN:</b>			
<b>AUTOR:</b>		<b>REVISOR:</b>	
		<b>FECHA DE REVISIÓN:</b>	
		<b>Nº REVISIÓN:</b>	
<b>Nombre:</b>		<b>RET</b>	<b>1</b>
<b>DET</b>			
<b>TOTAL</b>		<b>Complejidad</b>	

Formulario 4		SFP Cálculo de FP no Ajustados			
TÍTULO DE LA APLICACIÓN:					
AUTOR:					
ETAPA DEL PROYECTO:			FECHA:		
Tipo de Función	Complejidad Funcional		Totales por Complejidad		Totales por Tipo de Función
Recurso	_____	Baja	x 7	_____	_____
	_____	Media	x 10	_____	
EI	_____	Baja	x 3	_____	_____
	_____	Media	x 4	_____	
	_____	Alta	x 6	_____	
EO	_____	Baja	x 3	_____	_____
	_____	Media	x 4	_____	
	_____	Alta	x 6	_____	
<b>Total FP no Ajustados</b>					_____

Formulario 5		SFP Horas de trabajo			
TÍTULO DE LA APLICACIÓN:					
AUTOR:					
Paso	Fecha	Inicio (hora)	Fin (hora)	Intervalo (hora)	Comentarios
<b>Total</b>					

**C.1.1. Normas para completar los formularios**

En cada formulario se deben completar todos sus campos. Para todos los formularios en el encabezado se debe completar:

- a. Título de la aplicación: indicar el nombre de la aplicación.
- b. Autor: indicar el nombre del autor del proyecto.
- c. Fecha: indicar la fecha actual.

**- Formulario 1**

La planilla incluye tantas filas como *episodios* existan en los escenarios. Los campos se completan como se indica:

Ítem	Completar con...
Escenario	Nombre del escenario.
ID	Número de identificación de los episodios clasificados como EI o EO.
Episodio	Todos los episodios del escenario.
Tipo	D si el episodio es descartado. EI o EO en caso contrario.

- **Formularios 2 y 3**

El encabezado se debe completar como se indica a continuación:

- a. Revisor: indicar nombre del revisor del documento.
- b. Fecha de revisión: indicar la fecha de la revisión del documento.
- c. N° revisión: indicar el número de revisión del documento.

*Campos del formulario 2*

Ítem	Completar con...
<i>Escenario</i>	Nombre del escenario.
<i>ID</i>	Número de identificación.
<i>Tipo</i>	EI o EO
<i>RTR</i>	Lista con los nombres de los recursos referenciados.
<i>DET</i>	Lista con los nombres de los DET.
<i>Total</i>	Número de RTR y número de DET.
<i>Complejidad</i>	Complejidad (baja, media, alta).

*Campos del formulario 3*

Ítem	Completar con...
<i>Nombre</i>	Nombre del recurso.
<i>DET</i>	Lista con los nombres de los <i>DET</i> .
<i>Total</i>	Número de <i>DET</i> .
<i>Complejidad</i>	Complejidad (baja, media).

- **Formulario 4**

Completar los espacios indicados con una línea con el número de recursos, EIs y EOs de cada complejidad. Los *Totales por complejidad*, los *Totales por Tipo de Función* y el *Total FP no Ajustados* se calculan automáticamente.

- **Formulario 5**

Los campos se completan como se indica:

Ítem	Completar con...
<i>Inicio</i>	Hora de inicio de la actividad.
<i>Fin</i>	Hora de finalización de la actividad.
<i>Intervalo</i>	Diferencia entre Inicio y Fin.
<i>Total</i>	Sumatoria de los intervalos.





## Apéndice D

### Caso SFP Tool

En este apéndice son presentados el LEL y los escenarios del sistema SFP Tool, la estimación de FPs aplicando el procedimiento SFP, la Especificación de Requerimientos del sistema y la medición con el Método IFPUG.

#### D.1. LEL y Escenarios

##### LEL entries<sup>53</sup>

###### Analizar episodios

- Notion
  - Evaluar cada episodio en la Planilla de episodios y determinar su permanencia de acuerdo a lo establecido por las Reglas para identificar episodios.
- Behavioral responses
  - Permite obtener la lista de episodios para identificar episodios.

###### Calcular tamaño funcional / Calculo de FPs No Ajustados / Calcular la cuenta de FPs

- Notion
  - Obtener la cuenta total de los Puntos Función luego de evaluar la contribución de las EIs, EOs y los recursos y conocer el resultado del proceso de medición.
- Behavioral responses
  - Se registra la fecha de inicio, la hora de inicio y el paso del proceso de medición que se realiza en el Formulario 5.
  - Se multiplica el número de EIs por cada nivel de complejidad por su contribución en FPs obtenida desde la Tabla de complejidad y contribución para EI.
  - El resultado se registra en el Formulario 4.
  - Se multiplica el número de EOs de cada nivel de complejidad por su contribución en FPs obtenida desde la Tabla de complejidad y contribución para EO.
  - El resultado se registra en el Formulario 4.
  - Se multiplica el número de recursos de cada nivel de complejidad por su contribución en FPs obtenida desde la Tabla de complejidad y contribución para recursos.
  - El resultado se registra en el Formulario 4.
  - Se calcula la cuenta de FPs sumando los resultados de cada producto realizado anteriormente.
  - Se registra la cuenta total de FPs en el Formulario 4.
  - Se ingresa la fecha de finalización y la hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se registran los datos en el Formulario 5.
  - Se calcula el esfuerzo del proceso de medición.
  - El resultado se registra en el Formulario 5.

###### Clasificar episodios

- Notion
  - Paso del proceso de medición en el cual se evalúa cada episodio en la Planilla de episodios disponibles y se lo clasifica como EI o EO aplicando las Reglas para clasificar episodios.
  - Permite obtener la lista de episodios sobre la cual se trabajará en los pasos siguientes.
- Behavioral responses
  - Se registra la fecha de inicio, hora de inicio y paso del proceso de medición que se realiza en el Formulario 5.
  - Aplicando las Reglas para clasificar episodios, se identifica cada uno de los episodios que poseen ID en la lista de episodios como EI o EO según corresponda.

<sup>53</sup> Generado por BMW [Antonelli *et al.* 1999].

- Se almacenan los *episodios* clasificados como *EI* o *EO* en el *Formulario 1*.
- Se registra la *fecha de finalización* y *hora de finalización* de ese paso.
- Se calcula el intervalo de tiempo transcurrido.
- Se registran los datos en el *Formulario 5*.

---

Complejidad funcional / complejidad

- Notion
  - Es la valoración de la *complejidad* de cada *tipo de función*. Puede ser baja, media o alta.
  - Para *EIs* y *EOs* es determinada desde la *Tabla de complejidad y contribución para EI* y la *Tabla de Complejidad y Contribución para EO*, respectivamente, en función del número de *DETs* y *FTRs*. Para los *recursos* es determinada desde la *Tabla de Complejidad y Contribución para recursos* en función del número de *DETs* y *RETs*.
- Behavioral responses
  - Se utiliza para calcular la contribución en *FP* de *EIs*, *EOs* y *recursos*.

---

Contar DET en los episodios

- Notion
  - Establecer el número de *DETs* en los *episodios* en base a las *Reglas para contar DET en EIs y EOs* y su posterior uso como valor de entrada en la *Tabla de Complejidad y Contribución para EI* y *Tabla de complejidad y contribución para EO*.
- Behavioral responses
  - Permite *determinar la complejidad de los episodios*.

---

Contar DET en los recursos

- Notion
  - Establecer el número de *DET* en los *recursos* de acuerdo a las *Reglas para contar DET en los recursos*.
  - Este número obtenido, junto con el número de *RET*, es usado como valor de entrada a la *Tabla de complejidad y contribución para recursos*.
- Behavioral responses
  - Permite *determinar la complejidad de los recursos*.

---

Contar RET en los recursos

- Notion
  - Se aplica la *Regla para contar RET en los recursos*.
  - Este número junto con el número de *DET*, es usado como valor de entrada a la *Tabla de complejidad y contribución para recursos*.
- Behavioral responses
  - Permite *determinar la complejidad de los recursos*.

---

Contar RTR en los episodios

- Notion
  - Establecer el número de *RTR* en los *episodios* en base a las *Reglas para contar RTR en EIs y EOs* y su posterior uso como valor de entrada en la *Tabla de Complejidad y Contribución para EI* y la *Tabla de complejidad y contribución para EO*.
- Behavioral responses
  - Permite *determinar la complejidad de los episodios*.

---

Descartar episodios / Eliminar episodios

- Notion
  - Paso del proceso de *Analizar episodios* en el que se produce el descarte de algunos *episodios* aplicando las *Reglas para identificar episodios*.
- Behavioral responses
  - Permite determinar el conjunto definitivo de *episodios* creando la *lista de episodios*.

---

Determinar la complejidad de los episodios

- Notion
  - Paso del *proceso de medición* en el cual se efectúa la valoración de los *episodios* en función de los números de *DETs* y *RTRs* y así *determinar la contribución en FPs de los episodios*.
- Behavioral responses
  - Se registra el nombre y apellido del *autor* de la medición, la fecha, el nombre del *escenario*, el *ID* y *tipo* en el *Formulario 2*.

- Se registra la *fecha de inicio*, *hora de inicio* y paso del *proceso de medición* que se realiza en el *Formulario 5*.
- Aplicando las *Reglas para contar DET en EIs y EOs* y las *Reglas para contar RTR en EIs y EOs*, se realiza la cuenta de *DET* y *RTR* del *episodio* clasificado como *EI* o *EO* en la *Planilla de episodios*.
- El número de *DET* y *RTR* obtenido se registra en el *Formulario 2*.
- Si el *episodio* es *EI* se obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de Complejidad y Contribución para EI* los números de *DET* y *RTR* obtenidos.
- Si el *episodio* es *EO* se obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de Complejidad y Contribución para EO* los números de *DET* y *RTR* obtenidos.
- La *complejidad funcional* se registra en el *Formulario 2*.
- Se realiza la cuenta de *EIs* y *EOs* por nivel de *complejidad*.
- Los totales obtenidos se registran en el *Formulario 4*, junto con el apellido y nombre del *autor* de la medición y el título de la aplicación.
- Se registra la *fecha de finalización* y la *hora de finalización* de ese paso.
- Se calcula el intervalo de tiempo transcurrido.
- Se registran los datos en el *Formulario 5*.

Determinar la complejidad de los recursos

- Notion
  - Paso del *proceso de medición* en el cual se efectúa la valoración de los *recursos* en función de los números de *DETs* y *RETs* y así *determinar la contribución en FPs de los recursos*.
- Behavioral responses
  - Se registra la *fecha de inicio* y la *hora de inicio* del paso del *proceso de medición* que se realiza en el *Formulario 5*.
  - Se registra el nombre y apellido del *autor* de la medición, la fecha y el nombre del *recurso* en el *Formulario 3*.
  - Se identifican y se cuentan los *dets* del *recurso* en el *Formulario 3* aplicando las *Reglas para contar DET en los recursos*.
  - Se registra el número de *DET* del *recurso* en el *Formulario 3*.
  - Se aplica la *Regla para contar RET en los recursos* y se registra en número de *RET* en el *Formulario 3*.
  - Se obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de Complejidad y Contribución para recursos* los números de *DET* y *RET* obtenidos.
  - Se registra la *complejidad funcional* en el *Formulario 3*.
  - Se realiza la cuenta de *recursos* por cada nivel de *complejidad*.
  - Los totales de *recursos* por cada nivel de *complejidad* obtenidos se registran en el *Formulario 4*.
  - Se registra *fecha de finalización* y la *hora de finalización* de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se registran los datos en el *Formulario 5*.

Determinar la contribución en FPs de los episodios

- Notion
  - Paso del *proceso de medición* en el cual se realizan operaciones matemáticas que permitirán *calcular tamaño funcional*.
- Behavioral responses
  - Se multiplica el número de *EIs* de cada nivel de *complejidad* por su contribución en *FPs* obtenida desde la *Tabla de complejidad y contribución para EI*.
  - Se multiplica el número de *EOs* de cada nivel de *complejidad* por su contribución en *FPs* obtenida desde la *Tabla de complejidad y contribución para EO*.
  - Los resultados se registran en el *Formulario 4*.

Determinar la contribución en FPs de los recursos

- Notion
  - Paso del *proceso de medición* en el cual se realizan operaciones matemáticas que permitirán *calcular tamaño funcional*.
- Behavioral responses
  - Se multiplica el número de *recursos* de cada nivel de *complejidad* por su contribución en *FPs* obtenida desde la *Tabla de complejidad y contribución para recursos*.
  - Su resultado se almacena en el *Formulario 4*.

Entrada Externa / EI / EIs

- Notion
    - *episodio simple* que procesa datos que vienen desde fuera del límite de la aplicación.
  - Behavioral responses
    - Se utiliza para mantener uno o más *recursos*.
- 

Episodio / Episodios

- Notion
    - Representa una acción realizada por un actor, donde participan otros actores y se utilizan *recursos*. Puede ser un *escenario*.
  - Behavioral responses
    - Se utiliza para:
      - la elaboración de *escenarios* y de la *Planilla de episodios*,
      - *determinar la complejidad de los episodios*.
- 

Episodio candidato

- Notion
    - *Episodio* cuya inclusión en la *lista de episodios* no está determinada.
  - Behavioral responses
    - Se utiliza para generar la *Lista de Episodios disponibles*.
- 

Episodio condicional

- Notion
    - *Episodio* cuya ocurrencia depende de una condición específica.
  - Behavioral responses
    - Se utiliza para la creación de la *lista de episodios*.
- 

Episodio opcional

- Notion
    - *Episodio* que puede o no tener lugar dependiendo de ciertas condiciones.
  - Behavioral responses
    - Se utiliza para la creación de la *lista de episodios*.
- 

Episodio simple

- Notion
    - Sentencia simple (no *escenario*) que representa una acción atómica que se ejecuta en forma completa e independiente de cualquier otra acción dejando la aplicación en estado consistente.
  - Behavioral responses
    - Se utiliza para la elaboración de la *lista de episodios*.
- 

Episodios descartados

- Notion
    - *Episodios* que fueron eliminados de la *lista de episodios disponibles* al aplicar las *Reglas para identificar episodios*.
  - Behavioral responses
    - No serán utilizados en los pasos siguientes del *proceso de medición*.
- 

Escenario / Escenarios

- Notion
    - Constituye una descripción de los aspectos relevantes en cuanto al comportamiento y al ambiente de un sistema, usando lenguaje natural.
    - Incluye: nombre del *escenario*, objetivo, contexto, actores, *recursos* y *episodios*.
  - Behavioral responses
    - Se utiliza para construir la *lista de episodios disponibles*.
- 

Esfuerzo

- Notion
    - Cantidad de horas netas trabajadas por persona para realizar el *proceso de medición*.
    - Se almacena en el *Formulario 5*.
  - Behavioral responses
    - Se calcula sumando las horas de trabajo que demandó el *proceso de medición*.
-

Fecha de finalización

- Notion
    - Fecha que indica el final de un paso del proceso de medición.
  - Behavioral responses
    - Se utiliza en el Formulario 5.
- 

Fecha de inicio

- Notion
    - Fecha que indica el inicio de un paso del proceso de medición.
  - Behavioral responses
    - Se utiliza en el Formulario 5.
- 

Fecha de revisión

- Notion
    - Fecha en que se inicia una revisión a una medición existente.
  - Behavioral responses
    - Se utiliza para registrar una revisión en el conjunto de formularios de la medición.
- 

Formulario 2

- Notion
    - Contiene: autor, título de la aplicación, revisor, fecha de revisión, nro. de revisión, nombre del escenario, ID, tipo de cada episodio (EI / EO), dets, rtrs, número de dets y rtrs y nivel de complejidad.
  - Behavioral responses
    - Se utiliza para determinar la complejidad de los episodios.
- 

Formulario 3

- Notion
    - Contiene: autor, título de la aplicación, revisor, fecha de revisión, nro. de revisión, nombre del recurso, DETs, número de DETs y RETs y complejidad de los recursos.
  - Behavioral responses
    - Se utiliza para determinar la complejidad de los recursos.
- 

Formulario 4

- Notion
    - Contiene: autor, título de aplicación, etapa del proyecto, fecha, número de EIs, EOs y recursos por cada nivel de complejidad, complejidad de EIs y EOs (baja, media, alta) y de recursos (baja, media) y la cuenta de FPs.
  - Behavioral responses
    - Se utiliza para calcular tamaño funcional.
- 

Formulario 5

- Notion
    - Contiene: título de la aplicación, autor, paso del proceso de medición, hora de inicio, hora de finalización, intervalo de tiempo, fecha de inicio, fecha de finalización y total en horas.
  - Behavioral responses
    - Se utiliza para determinar el esfuerzo del proceso de medición.
- 

Hora de finalización

- Notion
    - Registro de la hora en que se finaliza (o se interrumpe) el trabajo durante el proceso de medición.
  - Behavioral responses
    - Se utiliza en el Formulario 5.
- 

Hora de inicio

- Notion
    - Registro de la hora en que se inicia (o se retoma) el trabajo de un paso del proceso de medición.
  - Behavioral responses
    - Se utiliza en el Formulario 5.
-

ID

- Notion
  - Número de identificación de un *episodio* clasificado como *EI* o *EO*.
- Behavioral responses
  - Se utiliza en el *Formulario 1* y en el *Formulario 2*.

Identificar el límite del sistema

- Notion
  - Paso del *proceso de medición* en el que se determina el conjunto de *episodios* que debe ser considerado para la medición, obteniéndose la *lista de episodios disponibles*.
- Behavioral responses
  - Se registra el nombre y apellido del *autor* de la medición, la fecha y el título de la aplicación en el *Formulario 1*.
  - Se registra la *fecha de inicio*, paso del *proceso de medición* que se realiza y la *hora de inicio* en el *Formulario 5*.
  - Se confecciona la *lista de episodios disponibles* con todos los *episodios* de todos los *escenarios* obtenidos desde el *LEL & escenarios*.
  - Los *episodios* de la *lista de episodios disponibles* se almacenan en el *Formulario 1*.
  - Se ingresa la *fecha de finalización* y la *hora de finalización* de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se registran los resultados en el *Formulario 5*.

Identificar episodios

- Notion
  - Paso del *proceso de medición* en el cual se determina el conjunto definitivo de *episodios* (*lista de episodios*) aplicando las *Reglas para identificar episodios* a los *episodios* de la *lista de episodios disponibles*.
- Behavioral responses
  - Se registra la *fecha de inicio*, *hora de inicio* y paso del *proceso de medición* que se realiza en el *Formulario 5*.
  - Se analiza cada uno de los *episodios* de la *lista de episodios disponibles* y se determina su permanencia aplicando las *Reglas para identificar episodios*.
  - Se identifican con "D" aquellos *episodios descartados*.
  - Se asigna el número *ID* a cada uno de los *episodios* de la *lista de episodios* que no fue descartado.
  - Se almacenan los *episodios* clasificados en el *Formulario 1*.
  - Se registra la *fecha de finalización* y *hora de finalización* de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se registran los datos en el *Formulario 5*.

Identificar recursos

- Notion
  - Paso del *proceso de medición* en el que cada *episodio* es examinado y sus *recursos* son identificados usando las *Reglas para identificar recursos*.
  - Permite crear la *lista de recursos*.
- Behavioral responses
  - Se registra el nombre y apellido del *autor* de la medición y el título de la aplicación en el *Formulario 3*.
  - Se registra la *fecha de inicio*, *hora de inicio* y el paso del *proceso de medición* que se está realizando en el *Formulario 5*.
  - Se evalúa el/los *recursos* de cada *episodio* no descartado en la *Planilla de episodios* aplicando las *Reglas para identificar recursos* junto con el *LEL & escenarios*.
  - Se verifica si el *recurso* se encuentra incluido en el *Formulario 3*.
  - Si el *recurso* no figura en el *Formulario 3*, se lo incluye.
  - Se ingresa la *fecha de finalización* y la *hora de finalización* de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se registran los datos en el *Formulario 5*.

Ingeniero de Requerimientos / Analista de Sistemas / Autor

- Notion
  - Persona entrenada en el manejo del procedimiento *SFP* que realiza la medición.

- Behavioral responses
  - Participa en:
  - *identificar el límite del sistema.*
  - *identificar episodios.*
  - *clasificar episodios.*
  - *determinar la complejidad de los episodios.*
  - *identificar recursos.*
  - *determinar la complejidad de los recursos.*
  - *calcular tamaño funcional.*

---

LEL & Escenarios

- Notion
  - Contiene el *LEL* y los *escenarios* acordados con el usuario de la aplicación de software.
- Behavioral responses
  - Se utiliza en los diferentes pasos del *proceso de medición* del procedimiento *SFP*.

---

Lista de Episodios / Planilla de Episodios / Formulario 1

- Notion
  - Contiene *autor*, título de la aplicación, fecha, nombre del *escenario*, *episodios*, *ID* y *tipo* de cada *episodio*.
- Behavioral responses
  - Se utiliza para *identificar episodios* y *clasificar episodios*.

---

Lista de episodios disponibles

- Notion
  - Contiene el conjunto de todos los *episodios* de todos los *escenarios*.
- Behavioral responses
  - Se utiliza para *identificar episodios*.

---

Lista de recursos

- Notion
  - Contiene el conjunto de los *recursos* no repetidos de todos los *episodios*.
- Behavioral responses
  - Se utiliza para *determinar la complejidad de los recursos*.

---

Léxico Extendido del Lenguaje / LEL

- Notion
  - Es un meta-modelo diseñado para ayudar a capturar el vocabulario de la aplicación, que utiliza el lenguaje natural.
  - Incluye: nombre del *símbolo*, noción e impacto.
- Behavioral responses
  - Se utiliza para representar los *símbolos* de la aplicación en la etapa de elicitación de requerimientos.

---

No Símbolo / No Símbolos

- Notion
  - palabra o frase perteneciente al vocabulario mínimo.
- Behavioral responses
  - Se utiliza para *identificar recursos* e identificar *DET* en *EL*, *EO* y *recursos*.

---

nro. de revisión / N° de Revisión

- Notion
  - Indica la cantidad de veces que se ha efectuado una *revisión* al conjunto de formularios de una medición.
- Behavioral responses
  - Se utiliza en el *Formulario 2* y en el *Formulario 3*.

---

Proceso de medición

- Notion
  - Consiste en varios pasos que se deben seguir en un orden determinado para calcular el *tamaño funcional*.

- Ellos son: *identificar el límite del sistema, identificar episodios, clasificar episodios, determinar la complejidad de los episodios, identificar recursos, determinar la complejidad de los recursos, calcular la cuenta de FPs.*
  - Están especificados en el procedimiento *SFP*.
  - Es realizado por el *ingeniero de requerimientos*.
  - Behavioral responses
    - Se aplica para estimar el *tamaño funcional* de los *escenarios*.
- 

Puntos Función / FP / FPs

- Notion
    - Unidad de medida lógica de las funciones del sistema de software desde el punto de vista del usuario. Miden el tamaño del software cuantificando la funcionalidad que brinda al usuario.
  - Behavioral responses
    - Se utiliza en el *proceso de medición* del *SFP*.
- 

Recurso / Recursos

- Notion
    - Representan elementos físicos o información relevante que debe estar disponible en el *escenario*. Identifica las entidades pasivas con las cuales los actores trabajan. Reciben las acciones indicadas por los verbos y ejecutadas por los sujetos. Se pueden clasificar en objetos físicos (computadora, teléfono, fax ) y objetos lógicos (formularios, planillas, etc.). Los objetos lógicos pueden ser *Símbolos Complejos* o *no símbolos*.
    - En el procedimiento *SFP* se trabaja con objetos lógicos únicamente.
  - Behavioral responses
    - Se utilizan :
    - durante la realización de un *episodio* y
    - para *Identificar recursos*
- 

Referencia a recurso / RTR / RTRs

- Notion
    - *recurso* referenciado por un *episodio simple*.
  - Behavioral responses
    - Se utiliza para *contar RTR en los episodios*.
- 

Regla para contar RET en los recursos

- Notion
    - Establece que se debe contar 1 (un) *RET* por *recurso*.
  - Behavioral responses
    - Se utiliza para *contar RET en los recursos*.
- 

Reglas para clasificar episodios

- Notion
    - Contiene las reglas que establecen como clasificar un *episodio simple* como *EI* o *EO* de acuerdo a su propósito principal.
  - Behavioral responses
    - Se utilizan para *clasificar episodios* en el procedimiento *SFP*.
- 

Reglas para contar DET en Els y EOs

- Notion
    - Son reglas que indican el criterio que se debe seguir al analizar el texto de un *episodio* para identificar los objetos que pertenecen a la clasificación *símbolo simple*, *Símbolo complejo* o *No Símbolo* y asignarles un valor numérico.
  - Behavioral responses
    - Se utilizan para *Contar DET en los episodios*.
- 

Reglas para contar DET en los recursos

- Notion
    - Son reglas que indican el criterio que se debe seguir para identificar los *DET* en los *recursos* de acuerdo a su clasificación: *símbolo complejo* o *No símbolo* y asignarles un valor numérico.
  - Behavioral responses
    - Se utilizan para *contar DET en los recursos*.
-



Reglas para contar RTR en EIs y EOs

- Notion
  - Son reglas que indican el criterio que se debe seguir al analizar el texto de un *episodio* para identificar los *recursos* que pertenecen a la clasificación *Símbolo complejo* o *No Símbolo* y asignarles un valor numérico.
- Behavioral responses
  - Se utilizan para *contar RTR en los episodios*.

Reglas para identificar episodios

- Notion
  - Contiene reglas específicas que determinan la inclusión (o exclusión) de cada *episodio* en la estimación del *tamaño funcional*.
- Behavioral responses
  - Se utilizan para *identificar episodios* en el procedimiento *SFP*.

Reglas para identificar recursos

- Notion
  - Contiene reglas que establecen las condiciones que deben cumplir los *recursos* para que sean considerados como tal en la medición.
- Behavioral responses
  - Se utilizan para *identificar recursos* en el procedimiento *SFP*.

Revisiones / Revisión

- Notion
  - Actividad en la cual el *revisor* evalúa y hace las correcciones necesarias a una medición existente.
  - Se identifica con el *n° de revisión*, el nombre del *revisor* y la *fecha de revisión*.
  - Se realiza sobre el *Formulario 2* y el *Formulario 3*.
- Behavioral responses
  - Se registra el nombre y apellido del *revisor*, la *fecha de revisión* y el *n° de revisión* que se realiza a la medición en el *Formulario 2* y el *Formulario 3*.
  - Se analizan los resultados del *Formulario 1* con el *LEL* y *Escenarios* y aplicando las *Reglas para identificar episodios* y las *Reglas para clasificar episodios*.
  - Se analizan los resultados del *Formulario 2* aplicando las *Reglas para contar DET en EIs y EOs* y las *Reglas para contar DET en los recursos* y se verifica cada nivel de *complejidad* con la *Tabla de complejidad y Contribución para EI* y la *Tabla de complejidad y contribución para EO*.
  - Se verifican los totales de *EI* y *EO* por cada nivel de *complejidad* del *Formulario 4*.
  - Se analiza el *Formulario 3* verificando con el *LEL & Escenarios* de la aplicación los *recursos* detallados aplicando las *Reglas para identificar recursos* y las *Reglas para contar DET en los recursos* y *Regla para contar RET en los recursos*.
  - Se verifica la *complejidad funcional* con la *Tabla de complejidad y contribución para recursos*.
  - Se verifican los totales de *recursos* por cada nivel de *complejidad* del *Formulario 4*.
  - Se verifican los cálculos del *Formulario 4* y la cuenta total de *FPS*.
  - Si existen diferencias con los valores de la medición existente se registran los resultados en una nueva *versión* de los formularios.

Revisor

- Notion
  - Persona entrenada en el procedimiento *SFP* que revisa una medición existente.
- Behavioral responses
  - Realiza la *revisión* utilizando la herramienta de soporte disponible.

Salida Externa / EO / EOs

- Notion
  - *episodio simple* que envía datos fuera del límite de la aplicación.
- Behavioral responses
  - Se utiliza para presentar información al usuario, sean datos recuperados desde uno o más *recursos* o datos derivados.

SFP

- Notion
    - Sigla que identifica al procedimiento *Puntos Función* de *escenarios* (Scenario Function Point).
  - Behavioral responses
    - Se utiliza para estimar el *tamaño funcional* de los *escenarios*.
- 

Símbolo / Símbolos

- Notion
    - Son palabras o frases utilizadas por el usuario y que repite con más frecuencia. También se incluyen aquellas palabras o frases que son relevantes para el dominio del problema más allá de su frecuencia de repetición.
    - Pueden ser objetos, acciones, sujetos o estados.
  - Behavioral responses
    - Se utiliza para la construcción del *LEL*.
- 

Símbolo complejo / Símbolos complejos

- Notion
    - Entrada del *LEL* formada por uno o más *símbolos Simples* y/o uno o más *No símbolo*.
    - Puede ser *recurso*.
  - Behavioral responses
    - Se utiliza para *identificar recursos* o para identificar *DETs* en *EIs* y *EOs*.
- 

Símbolo Simple / Símbolos simples

- Notion
    - Ítem atómico de información reconocible por el usuario.
    - Puede ser *DET* en *EI*, *EO* y *recurso*.
  - Behavioral responses
    - Se utiliza para identificar *DETs* en *EIs*, *EOs* y *recursos*.
- 

Tabla de complejidad y Contribución para EI

- Notion
    - Tabla de IFPUG que contiene el nivel de *complejidad* de los *episodios* clasificados como *EIs* de acuerdo a su número de *DETs* y *RTRs* y el valor numérico correspondiente.
  - Behavioral responses
    - Se utiliza para *determinar la complejidad de los episodios*.
- 

Tabla de Complejidad y Contribución para EO

- Notion
    - Tabla de IFPUG que contiene el nivel de *complejidad* de los *episodios* clasificados como *EO* de acuerdo a su número de *DETs* y *RTRs* y el valor numérico correspondiente.
  - Behavioral responses
    - Se utiliza para *determinar la complejidad de los episodios*.
- 

Tabla de Complejidad y Contribución para recursos

- Notion
    - Tabla de IFPUG que contiene el nivel de *complejidad* de los *recursos* de acuerdo a sus números de *DETs* y *RETs* y el valor numérico correspondiente.
  - Behavioral responses
    - Se utiliza para *determinar la complejidad de los recursos*.
- 

Tamaño funcional

- Notion
    - Medida de la funcionalidad que implementa una aplicación de software, desde el punto de vista del usuario. Es independiente de cualquier aspecto técnico y de cualquier decisión de implementación.
    - Se expresa en *Puntos Función*.
  - Behavioral responses
    - Se determina mediante el *proceso de medición*.
- 

Tipo

- Notion
  - Clasificación que identifica a un *episodio* en la *Planilla de Episodios*

- Comprende: *EI*, *EO* o *D* (*episodios descartados*).
- Behavioral responses
  - Se utiliza para *clasificar episodios* de la *lista de episodios*.

Tipo de Dato Elemental / DET / DETs

- Notion
  - Ítem atómico de información identificable por el usuario. Es un *símbolo Simple* o un *No Símbolo* (no *recurso*).
- Behavioral responses
  - Se utiliza para:
  - *contar DET en los episodios*,
  - *contar DET en los recursos*.

Tipo de Función

- Notion
  - Clasificación que se refiere a *EI*, *EO* o *recurso* de la aplicación.
- Behavioral responses
  - Se utiliza en el *Formulario 4*.

Tipo de registro elemental / RET / RETs

- Notion
  - En el contexto de *SFP* representa un grupo de datos elementales incluidos en un *recurso*.
- Behavioral responses
  - Se utiliza para *contar RET en los recursos*.

Versión

- Notion
  - Contiene los datos de una medición existente que, luego del proceso de *revisión*, puede arrojar diferencias con los resultados obtenidos inicialmente.
- Behavioral responses
  - Permite almacenar los resultados de una medición efectuada y revisada, sin perder los valores originales.

## Scenarios

**Calcular el esfuerzo del proceso de medición**

- Goal
  - Determinar el *esfuerzo* total del *proceso de medición*.
- Context
  - El *ingeniero de requerimientos* ha finalizado todos los pasos del *proceso de medición*.
- Resources
  - *Formulario 5*
- Actors
  - *autor*
- Episodes
  - El *autor* realiza la suma de las horas de trabajo que demandó el *proceso de medición*.
  - El resultado obtenido (*esfuerzo* del *proceso de medición*) se almacena en el *Formulario 5*.

**calcular tamaño funcional**

- Goal
  - Obtener la cuenta total de *FPs*
- Context
  - El *ingeniero de requerimientos* cuenta con el *Formulario 4* con el detalle del número de *EIs*, *EOs* y *recursos* por cada nivel de *complejidad*
- Resources
  - *Formulario 4*
  - *Formulario 5*

- Actors
  - Autor
- Episodes
  - El autor ingresa su nombre, el título de la aplicación, la fecha de inicio, la hora de inicio y el paso del proceso de medición en el Formulario 5.
  - **Determinar contribución en FPs de los episodios**
  - **Determinar contribución en FPs de los recursos**
  - Se calcula la cuenta de FPs sumando los resultados obtenidos anteriormente.
  - Se almacena la cuenta total de FPs en el Formulario 4.
  - El autor ingresa la fecha de finalización y la hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la fecha de finalización, la hora de finalización y el intervalo en el Formulario 5.
  - **Calcular el esfuerzo del proceso de medición**

---

**clasificar episodios**

- Goal
  - Clasificar los episodios de la Lista de Episodios como EI o EO aplicando las Reglas para clasificar episodios
- Context
  - El Ingeniero de requerimientos dispone de la Lista de Episodios y de las Reglas para clasificar episodios.
- Resources
  - Formulario 1
  - Formulario 5
  - Reglas para clasificar episodios
- Actors
  - autor
- Episodes
  - El autor ingresa la fecha de inicio, hora de inicio y paso del proceso de medición en el Formulario 5.
  - El autor clasifica como EI o EO cada episodio con ID de la lista de episodios aplicando las Reglas para clasificar episodios.
  - El autor almacena el tipo de los episodios (EI, EO, D) en el Formulario 1
  - El autor ingresa la fecha de finalización y hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la fecha de finalización, la hora de finalización y el intervalo en el Formulario 5.
  - **restricción:** No se cuenta con las Reglas para clasificar episodios.

---

**Contar EIs y EOs por nivel de complejidad**

- Goal
  - Establecer el número total de EIs y de EOs por cada nivel de complejidad.
- Context
  - El ingeniero de requerimientos cuenta con el Formulario 2 detallando cada EI y EO con su nivel de complejidad.
- Resources
  - Formulario 2
  - Formulario 4
- Actors
  - Autor
- Episodes
  - El autor ingresa su nombre y apellido y el título de la aplicación en el Formulario 4.
  - El autor analiza el Formulario 2 y realiza la cuenta de EIs y EOs por cada nivel de complejidad.
  - El total de EIs por cada nivel de complejidad obtenido se almacena en el Formulario 4.
  - El total de EOs por cada nivel de complejidad obtenido se almacena en el Formulario 4.

---

**Contar recursos por nivel de complejidad**

- Goal
  - Establecer el número total de recursos por cada nivel de complejidad.

- Context
  - El *Ingeniero de Requerimientos* cuenta con el *Formulario 3* detallando los *recursos* y su nivel de *complejidad*.
- Resources
  - *Formulario 3*
  - *Formulario 4*
- Actors
  - *autor*
- Episodes
  - El *autor* analiza el *Formulario 3* y realiza la cuenta de *recursos* por cada nivel de *complejidad*.
  - Los totales de *recursos* por cada nivel de *complejidad* obtenidos se almacenan en el *Formulario 4*.

---

#### **Crear una medición**

- Goal
  - Iniciar un *proceso de medición* para *calcular tamaño funcional* con el método *SFP*.
- Context
  - Se realiza una vez que se cuenta con el *LEL & escenarios* acordados con el usuario de la aplicación de software que se está relevando.
- Resources
- Actors
- Episodes
  - **identificar el límite del sistema**
  - **identificar episodios**
  - **clasificar episodios**
  - **Determinar complejidad de los episodios**
  - **identificar recursos**
  - **Determinar complejidad de los recursos**
  - **calcular tamaño funcional**

---

#### **Determinar complejidad de los episodios**

- Goal
  - Asignar un nivel de *complejidad* a las *EIs* y *EOs* de los *episodios*.
- Context
  - El *Ingeniero de requerimientos* cuenta con la *Planilla de episodios* clasificados como *EI* o *EO* y el *Formulario 2* para cada uno de los *episodios* (*EIs* y *EOs*).
- Resources
  - *Planilla de episodios*
  - *Formulario 2*
  - *Formulario 5*
  - *Reglas para contar DET en EIs y EOs*
  - *Reglas para contar RTR en EIs y EOs*
  - *Tabla de complejidad y contribución para EI*
  - *Tabla de complejidad y contribución para EO*
- Actors
  - *autor*
- Episodes
  - El *autor* ingresa su nombre y apellido, la fecha, el nombre del *escenario*, *ID* y *tipo* en el *Formulario 2*.
  - El *autor* ingresa la *fecha de inicio*, *hora de inicio* y paso del *proceso de medición* en el *Formulario 5*.
  - El *autor*, aplicando las *Reglas para contar DET en EIs y EOs* y las *Reglas para contar RTR en EIs y EOs*, realiza la cuenta de *DET* y *RTR* del *episodio* clasificado como *EI* o *EO* en la *Planilla de episodios*.
  - El número de *DET* y *RTR* obtenido se almacena en el *Formulario 2*.
  - **if** el *episodio* es *EI* **then** el *autor* obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de Complejidad y Contribución para EI* los números de *DET* y *RTR* obtenidos.

- **if** el *episodio* es *EO* **then** el *autor* obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de complejidad y contribución para EO* los números de *DET* y *RTR* obtenidos.
- La *complejidad funcional* se almacena en el *Formulario 2*.
- **Contar EIs y EOs por nivel de complejidad**
- El *autor* ingresa la *fecha de finalización* y la *hora de finalización* de ese paso.
- Se calcula el intervalo de tiempo transcurrido.
- Se almacenan la *fecha de finalización*, la *hora de finalización* y el intervalo en el *Formulario 5*.
- **restriction:** No se cuenta con las *Reglas para contar DET en EIs y EOs*
- No se cuenta con las *Reglas para contar RTR en EIs y EOs*
- No se cuenta con la *Tabla de Complejidad y Contribución para EI*
- No se cuenta con la *Tabla de complejidad y contribución para EO*

---

#### **Determinar complejidad de los recursos**

- Goal
  - Asignar un nivel de *complejidad* a cada *recurso*.
- Context
  - El *ingeniero de requerimientos* cuenta con el detalle de *recursos* en el *Formulario 3*
- Resources
  - *Formulario 3*
  - *Reglas para contar DET en los recursos*
  - *Regla para contar RET en los recursos*
  - *Tabla de complejidad y contribución para recursos*
- Actors
  - *Autor*
- Episodes
  - El *autor* ingresa la *fecha de inicio* y la *hora de inicio* del paso del *proceso de medición* en el *Formulario 5*.
  - El *autor* identifica y cuenta los *dets* de cada *recurso* del *Formulario 3* aplicando las *Reglas para contar DET en los recursos*.
  - Se almacena el número de *DET* de cada *recurso* en el *Formulario 3*.
  - El *autor* aplica la *Regla para contar RET en los recursos*.
  - Se almacena el número de *RET* en el *Formulario 3*.
  - El *autor* obtiene la *complejidad funcional* utilizando como valores de entrada a la *Tabla de complejidad y contribución para recursos* los números de *DET* y *RET* obtenidos.
  - La *complejidad funcional* se almacena en el *Formulario 3*.
  - **Contar recursos por nivel de complejidad**
  - El *autor* ingresa la *fecha de finalización* y la *hora de finalización* de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la *fecha de finalización*, la *hora de finalización* y el intervalo en el *Formulario 5*.
  - **restriction:** No se cuenta con las *Reglas para contar DET en los recursos*
  - No se cuenta con la *Regla para contar RET en los recursos*
  - No se cuenta con la *Tabla de complejidad y contribución para recursos*.

---

#### **Determinar contribución en FPs de los episodios**

- Goal
  - Realizar el cálculo que permita obtener la contribución en *FPs* de los *episodios* clasificados como *EI* y *EO* para calcular el *tamaño funcional*.
- Context
  - El *ingeniero de requerimientos* cuenta en el *Formulario 4* con el detalle del número de *EIs* y *EOs* por cada nivel de *complejidad*.
- Resources
  - *Formulario 4*
  - *Tabla de Complejidad y Contribución para EI*
  - *Tabla de complejidad y contribución para EO*
- Actors
  - *Autor*
- Episodes
  - El *autor* multiplica el número de *EIs* de cada nivel de *complejidad* por su contribución en *FPs* obtenida desde la *Tabla de complejidad y contribución para EI*.

- El resultado se almacena en el Formulario 4.
- El autor multiplica el número de EOs de cada nivel de complejidad por su contribución en FPs obtenida desde la Tabla de complejidad y contribución para EO.
- El resultado se almacena en el Formulario 4.
- **restricción:** No se cuenta con la Tabla de complejidad y contribución para EI
- No se cuenta con la Tabla de complejidad y contribución para EO

---

#### **Determinar contribución en FPs de los recursos**

- Goal
  - Realizar el cálculo que permita obtener la contribución en FPs de los recursos y calcular el tamaño funcional.
- Context
  - El ingeniero de requerimientos cuenta en el Formulario 4 con el detalle del número de recursos por nivel de complejidad.
- Resources
  - Formulario 4
  - Tabla de complejidad y contribución para recursos
- Actors
  - Autor
- Episodes
  - El autor multiplica el número de recursos de cada nivel de complejidad por su contribución en FPs obtenida desde la Tabla de complejidad y contribución para recursos.
  - Su resultado se almacena en el Formulario 4.
  - **restricción:** No se cuenta con la Tabla de complejidad y contribución para recursos

---

#### **identificar el límite del sistema**

- Goal
  - Determinar el conjunto de escenarios que debe ser considerado para la medición.
- Context
  - El Ingeniero de Requerimientos dispone del LEL & escenarios de la aplicación que desea medir y del conjunto de formularios para el proceso de medición SFP.
- Resources
  - LEL & escenarios
  - Formulario 1
  - Formulario 5
- Actors
  - autor
- Episodes
  - El autor ingresa su nombre y apellido, fecha y el título de la aplicación en el Formulario 1.
  - El autor ingresa la fecha de inicio, paso del proceso de medición que realiza y la hora de inicio en el Formulario 5.
  - El autor confecciona la lista de episodios disponibles con todos los episodios de todos los escenarios obtenidos desde el LEL & escenarios.
  - Los episodios de la lista de episodios disponibles se almacenan en el Formulario 1.
  - El autor ingresa la fecha de finalización y la hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la fecha de finalización, la hora de finalización y el intervalo en el Formulario 5.

---

#### **identificar episodios**

- Goal
  - Determinar el conjunto definitivo de episodios aplicando las Reglas para identificar episodios a los episodios de la lista de episodios disponibles.
- Context
  - El Ingeniero de requerimientos posee la lista de episodios disponibles y las Reglas para identificar episodios.
- Resources
  - lista de episodios disponibles
  - Formulario 1
  - Formulario 5
  - Reglas para identificar episodios

- Actors
  - autor
- Episodes
  - El autor ingresa la fecha de inicio, hora de inicio y paso del proceso de medición en el Formulario 5.
  - El autor analiza cada uno de los episodios de la lista de episodios disponibles y determina su permanencia aplicando las Reglas para identificar episodios.
  - El autor identifica con "D" aquellos episodios descartados.
  - Se asigna el número ID a cada episodio que no fue descartado.
  - Se almacena el ID de los episodios en el Formulario 1.
  - El autor ingresa la fecha de finalización y hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la fecha de finalización, la hora de finalización y el intervalo en el Formulario 5.
  - **restricción:** No se cuenta con las Reglas para identificar episodios

---

#### **identificar recursos**

- Goal
  - Obtener la lista de recursos usados por los episodios.
- Context
  - El ingeniero de requerimientos cuenta con la Lista de Episodios y las Reglas para identificar recursos.
- Resources
  - Lista de Episodios
  - LEL & escenarios
  - Reglas para identificar recursos
  - Formulario 3
  - Formulario 5
- Actors
  - Autor
- Episodes
  - El autor ingresa su nombre y apellido y el título de la aplicación en el Formulario 3.
  - El autor ingresa la fecha de inicio, hora de inicio y el paso del proceso de medición en el Formulario 5.
  - El autor evalúa el/los recursos de cada episodio no descartado en la Planilla de episodios aplicando las Reglas para identificar recursos junto con el LEL & escenarios.
  - El autor verifica si el recurso se encuentra incluido en el Formulario 3.
  - **if** el recurso no está en incluido en el Formulario 3 **then** se almacena el nombre del recurso en el Formulario 3.
  - El autor ingresa la fecha de finalización y la hora de finalización de ese paso.
  - Se calcula el intervalo de tiempo transcurrido.
  - Se almacenan la fecha de finalización, la hora de finalización y el intervalo en el Formulario 5.
  - **restricción:** No se cuenta con las Reglas para identificar recursos

---

#### **Revisar una medición**

- Goal
  - Verificar la estimación del tamaño funcional.
- Context
  - El revisor cuenta con los formularios iniciales de la medición (y de otras revisiones si se hubieran hecho).
- Resources
  - Formulario 1
  - Formulario 2
  - Formulario 3
  - Formulario 4
  - LEL & escenarios
  - Regla para contar RET en los recursos
  - Reglas para clasificar episodios
  - Reglas para identificar episodios
  - Reglas para identificar recursos
  - Reglas para contar DET en Els y EOs



- Reglas para contar DET en los recursos
- Reglas para contar RTR en EIs y EOs
- Tabla de complejidad y Contribución para EI
- Tabla de Complejidad y Contribución para EO
- Tabla de Complejidad y Contribución para recursos
- Actors
  - revisor
- Episodes
  - El revisor ingresa su nombre y apellido, la fecha de revisión y el Nº de revisión en el Formulario 2 y el Formulario 3.
  - El revisor analiza los resultados del Formulario 1 con el LEL y Escenarios y aplicando las Reglas para identificar episodios y las Reglas para clasificar episodios.
  - El revisor analiza los resultados del Formulario 2 aplicando las Reglas para contar DET en EIs y EOs y las Reglas para contar DET en los recursos y verifica cada nivel de complejidad con la Tabla de complejidad y Contribución para EI y la Tabla de complejidad y contribución para EO.
  - El revisor verifica los totales de EI y EO por cada nivel de complejidad del Formulario 4.
  - El revisor analiza el Formulario 3 verificando con el LEL & Escenarios de la aplicación los recursos detallados aplicando las Reglas para identificar recursos y las Reglas para contar DET en los recursos y Regla para contar RET en los recursos y verifica la complejidad funcional con la Tabla de complejidad y contribución para recursos.
  - El revisor verifica los totales de recursos por cada nivel de complejidad del Formulario 4.
  - El revisor verifica los cálculos del Formulario 4 y la cuenta total de FPS.
  - **if** existen diferencias con la medición existente **then** el revisor almacena los resultados en una nueva versión del Formulario 1 y/o Formulario 2 y/o Formulario 3 y/o Formulario 4.
  - **restriction:** No se cuenta con los formularios de la medición para hacer la revisión.

---

#### Usar SFP

- Goal
    - Realizar (o revisar) el cálculo del tamaño funcional de una aplicación de software utilizando el procedimiento SFP.
  - Context
    - Se cuenta con el LEL & Escenarios de la aplicación a medir.
    - Se cuenta con la medición existente para hacer la revisión.
  - Resources
  - Actors
  - Episodes
    - **Crear una medición**
    - **Revisar una medición**
    - **restriction:** el LEL & Escenarios no fue generado con la herramienta Baseline Mentor Workbench (BMW).
-

D.2. Estimación SFP

Formulario 1		SFP PLANILLA DE EPISODIOS	
TÍTULO DE LA APLICACIÓN: SFP Tool			
AUTOR: Mabel		FECHA: 14/06/08	
Escenario	ID	Episodio	Tipo
<u>Calcular el esfuerzo del proceso de medición</u>		- El <i>autor</i> realiza la suma de las horas de trabajo que demandó el <i>proceso de medición</i> .	D
	1	- El resultado obtenido ( <i>esfuerzo</i> del <i>proceso de medición</i> ) se almacena en el <i>Formulario 5</i> .	EI
<u>Calcular tamaño funcional</u>	2	- El <i>autor</i> ingresa su nombre, el título de la aplicación, la <i>fecha de inicio</i> , la <i>hora de inicio</i> y el paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	EI
		- <b>Determinar contribución en FPs de los episodios</b>	D
		- <b>Determinar contribución en FPs de los recursos</b>	D
		- Se calcula la cuenta de <i>FPs</i> sumando los resultados obtenidos anteriormente.	D
	3	- Se almacena la cuenta total de <i>FPs</i> en el <i>Formulario 4</i> .	EI
		- El <i>autor</i> ingresa la <i>fecha de finalización</i> y la <i>hora de finalización</i> de ese paso.	D
		- Se calcula el intervalo de tiempo transcurrido.	D
	4	- Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	EI
	- <b>Calcular el esfuerzo del proceso de medición</b>	D	
<u>Clasificar episodios</u>		- El <i>autor</i> ingresa la <i>fecha de inicio</i> , <i>hora de inicio</i> y paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	D
	5	- El <i>autor</i> clasifica como <i>EI</i> o <i>EO</i> cada episodio con <i>ID</i> de la <i>lista de episodios</i> aplicando las <i>Reglas para clasificar episodios</i> .	EO
	6	- El <i>autor</i> almacena los <i>episodios</i> clasificados en el <i>Formulario 1</i> .	EI
		- El <i>autor</i> ingresa la <i>fecha de finalización</i> y <i>hora de finalización</i> de ese paso.	D
		- Se calcula el intervalo de tiempo transcurrido.	D
		- Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
	- <b>restricción:</b> No se cuenta con las <i>Reglas para clasificar episodios</i> .	D	
<u>Contar EIs y EOs por nivel de complejidad</u>	7	- El <i>autor</i> ingresa su nombre y apellido y el título de la aplicación en el <i>Formulario 4</i> .	EI
		- El <i>autor</i> analiza el <i>Formulario 2</i> y realiza la cuenta de <i>EIs</i> y <i>EOs</i> por cada <i>nivel de complejidad</i> .	D
	8	- El total de <i>EIs</i> por cada <i>nivel de complejidad</i> obtenido se almacena en el <i>Formulario 4</i> .	EI
	9	- El total de <i>EOs</i> por cada <i>nivel de complejidad</i> obtenido se almacena en el <i>Formulario 4</i> .	EI
<u>Contar recursos por nivel de complejidad</u>		- El <i>autor</i> analiza el <i>Formulario 3</i> y realiza la cuenta de <i>recursos</i> por cada <i>nivel de complejidad</i> .	D
	10	- Los totales de <i>recursos</i> por cada <i>nivel de complejidad</i> obtenidos se almacenan en el <i>Formulario 4</i> .	EI
<u>Crear una medición</u>		- <b>identificar el límite del sistema</b>	D
		- <b>identificar episodios</b>	D
		- <b>clasificar episodios</b>	D
		- <b>Determinar complejidad de los episodios</b>	D
		- <b>identificar recursos</b>	D
		- <b>Determinar complejidad de los recursos</b>	D
	- <b>calcular tamaño funcional</b>	D	
<u>Determinar complejidad</u>	11	- El <i>autor</i> ingresa su nombre y apellido, la fecha, el <i>escenario</i> , <i>ID</i> y <i>tipo</i> en el <i>Formulario 2</i> .	EI

Formulario 1	SFP PLANILLA DE EPISODIOS		
TÍTULO DE LA APLICACIÓN: SFP Tool			
AUTOR: Mabel		FECHA: 14/06/08	
Escenario	ID	Episodio	Tip o
<b>de los episodios</b>		– El <i>autor</i> ingresa la <i>fecha de inicio</i> , <i>hora de inicio</i> y paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	D
		– El <i>autor</i> , aplicando las <i>Reglas para contar DET en EIs y EOs</i> y las <i>Reglas para contar RTR en EIs y EOs</i> , realiza la cuenta de <i>DET</i> y <i>RTR</i> del <i>episodio</i> clasificado como <i>EI</i> o <i>EO</i> en la <i>Planilla de episodios</i> .	D
	1 2	– El número de <i>DET</i> y <i>RTR</i> obtenido se almacena en el <i>Formulario 2</i> .	EI
		– <b>if</b> el <i>episodio</i> es <i>EI</i> <b>then</b> el <i>autor</i> obtiene la <i>complejidad funcional</i> utilizando como valores de entrada a la <i>Tabla de Complejidad y Contribución para EI</i> los números de <i>DET</i> y <i>RTR</i> obtenidos.	D
		– <b>if</b> el <i>episodio</i> es <i>EO</i> <b>then</b> el <i>autor</i> obtiene la <i>complejidad funcional</i> utilizando como valores de entrada a la <i>Tabla de complejidad y contribución para EO</i> los números de <i>DET</i> y <i>RTR</i> obtenidos.	D
	1 3	– La <i>complejidad funcional</i> se almacena en el <i>Formulario 2</i> .	EI
		– <b>Contar EIs y EOs por nivel de complejidad</b>	D
		– El <i>autor</i> ingresa la <i>fecha de finalización</i> y la <i>hora de finalización</i> de ese paso.	D
		– Se calcula el intervalo de tiempo transcurrido.	D
		– Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
		– <b>restriction:</b> No se cuenta con las <i>Reglas para contar DET en EIs y EOs</i>	D
		– No se cuenta con las <i>Reglas para contar RTR en EIs y EOs</i> .	D
		– No se cuenta con la <i>Tabla de Complejidad y Contribución para EI</i>	D
		– No se cuenta con la <i>Tabla de complejidad y contribución para EO</i>	D
	<b>Determinar complejidad de los recursos</b>		– El <i>autor</i> ingresa la <i>fecha de inicio</i> y la <i>hora de inicio</i> del paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .
		– El <i>autor</i> identifica y cuenta los <i>DETs</i> de cada <i>recurso</i> del <i>Formulario 3</i> aplicando las <i>Reglas para contar DET en los recursos</i> .	D
1 4		– Se almacena el número de <i>DET</i> de cada <i>recurso</i> en el <i>Formulario 3</i> .	EI
		– El autor aplica la <i>regla para contar RET en los recursos</i> .	D
1 5		– Se almacena el número de <i>RET</i> en el <i>Formulario 3</i> .	EI
		– El <i>autor</i> obtiene la <i>complejidad funcional</i> utilizando como valores de entrada a la <i>Tabla de complejidad y contribución para recursos</i> los números de <i>DET</i> y <i>RET</i> obtenidos.	D
1 6		– La <i>complejidad funcional</i> se almacena en el <i>Formulario 3</i> .	EI
		– <b>Contar recursos por nivel de complejidad</b>	D
		– El <i>autor</i> ingresa la <i>fecha de finalización</i> y la <i>hora de finalización</i> de ese paso.	D
		– Se calcula el intervalo de tiempo transcurrido.	D
		– Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
		– <b>restriction:</b> No se cuenta con las <i>Reglas para contar DET en los recursos</i>	D
		– No se cuenta con la <i>regla para contar RET en los recursos</i>	D
		– No se cuenta con la <i>Tabla de complejidad y contribución para recursos</i> .	D

Formulario 1		SFP PLANILLA DE EPISODIOS	
TÍTULO DE LA APLICACIÓN: SFP Tool			
AUTOR: Mabel		FECHA: 14/06/08	
Escenario	ID	Episodio	Tip o
<b>Determinar contribución en FPs de los episodios</b>		– El <i>autor</i> multiplica el número de <i>EIs</i> de cada <i>nivel de complejidad</i> por su contribución en <i>FPs</i> obtenida desde la <i>Tabla de complejidad y contribución para EI</i> .	D
	17	– El resultado se almacena en el <i>Formulario 4</i> .	EI
		– El <i>autor</i> multiplica el número de <i>EOs</i> de cada <i>nivel de complejidad</i> por su contribución en <i>FPs</i> obtenida desde la <i>Tabla de complejidad y contribución para EO</i> .	D
		– El resultado se almacena en el <i>Formulario 4</i> .	D
		– <b>restricción:</b> No se cuenta con la <i>Tabla de complejidad y contribución para EI</i>	D
		– No se cuenta con la <i>Tabla de complejidad y contribución para EO</i>	D
<b>Determinar contribución en FPs de los recursos</b>		– El <i>autor</i> multiplica el número de <i>recursos</i> de cada <i>nivel de complejidad</i> por su contribución en <i>FPs</i> obtenida desde la <i>Tabla de complejidad y contribución para recursos</i> .	D
		– Su resultado se almacena en el <i>Formulario 4</i> .	D
		– <b>restricción:</b> No se cuenta con la <i>Tabla de complejidad y contribución para recursos</i>	D
<b>Identificar el límite del sistema</b>	18	– El <i>autor</i> ingresa su nombre y apellido, fecha y el título de la aplicación en el <i>Formulario 1</i> .	EI
		– El <i>autor</i> ingresa la <i>fecha de inicio</i> , paso del <i>proceso de medición</i> que realiza y la <i>hora de inicio</i> en el <i>Formulario 5</i> .	D
	19	– El <i>autor</i> confecciona la <i>lista de episodios disponibles</i> con todos los <i>episodios</i> de todos los <i>escenarios</i> obtenidos desde el <i>LEL &amp; escenarios</i> .	EI
	20	– La <i>episodios</i> de la <i>lista de episodios disponibles</i> se almacenan en el <i>Formulario 1</i> .	EI
		– El <i>autor</i> ingresa la <i>fecha de finalización</i> y la <i>hora de finalización</i> de ese paso.	D
		– Se calcula el intervalo de tiempo transcurrido.	D
		– Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
<b>Identificar episodios</b>		– El <i>autor</i> ingresa la <i>fecha de inicio</i> , <i>hora de inicio</i> y paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	D
	21	– El <i>autor</i> analiza cada uno de los <i>episodios</i> de la <i>lista de episodios disponibles</i> y determina su permanencia aplicando las <i>Reglas para identificar episodios</i> .	EO
		– El <i>autor</i> identifica con "D" aquellos <i>episodios descartados</i> .	D
		– Se asigna el número <i>ID</i> a cada <i>episodio</i> que no fue descartado.	D
	22	– Se almacena el <i>ID</i> de los <i>episodios</i> en el <i>Formulario 1</i> .	EI
		– El <i>autor</i> ingresa la <i>fecha de finalización</i> y <i>hora de finalización</i> de ese paso.	D
		– Se calcula el intervalo de tiempo transcurrido.	D
		– Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
	– <b>restricción:</b> No se cuenta con las <i>Reglas para identificar episodios</i>	D	
<b>Identificar recursos</b>	23	– El <i>autor</i> ingresa su nombre y apellido y el título de la aplicación en el <i>Formulario 3</i> .	EI
		– El <i>autor</i> ingresa la <i>fecha de inicio</i> , <i>hora de inicio</i> y el paso del <i>proceso de medición</i> en el <i>Formulario 5</i> .	D
	24	– El <i>autor</i> evalúa el/los <i>recursos</i> de cada <i>episodio</i> no descartado en la <i>Planilla de episodios</i> aplicando las <i>Reglas para identificar recursos</i> junto con el <i>LEL &amp; escenarios</i> .	EO

Formulario 1		SFP PLANILLA DE EPISODIOS	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool <b>AUTOR:</b> Mabel <span style="float: right;"><b>FECHA:</b> 14/06/08</span>			
Escenario	ID	Episodio	Tipo
	25	– El <i>autor</i> verifica si el <i>recurso</i> se encuentra incluido en el <i>Formulario 3</i> .	EO
	26	– <b>if</b> el <i>recurso</i> no está en incluido en el <i>Formulario 3</i> <b>then</b> se almacena el <i>recurso</i> en el <i>Formulario 3</i> .	EI
		– El <i>autor</i> ingresa la <i>fecha de finalización</i> y la <i>hora de finalización</i> de ese paso.	D
		– Se calcula el intervalo de tiempo transcurrido.	D
		– Se almacenan la <i>fecha de finalización</i> , la <i>hora de finalización</i> y el intervalo en el <i>Formulario 5</i> .	D
		– <b>restriction:</b> No se cuenta con las <i>Reglas para identificar recursos</i>	D
<b>Revisar una medición</b>	27	– El <i>revisor</i> ingresa su nombre y apellido, la <i>fecha de revisión</i> y el <i>Nº de revisión</i> en el <i>Formulario 2</i> y el <i>Formulario 3</i> .	EI
	28	– El <i>revisor</i> analiza los resultados del <i>Formulario 1</i> con el <i>LEL</i> y <i>Escenarios</i> y aplicando las <i>Reglas para identificar episodios</i> y las <i>Reglas para clasificar episodios</i>	EO
	29	– El <i>revisor</i> analiza los resultados del <i>Formulario 2</i> aplicando las <i>Reglas para contar DET en Els y EOs</i> y las <i>Reglas para contar DET en los recursos</i> y verifica cada <i>nivel de complejidad</i> con la <i>Tabla de complejidad y Contribución para EI</i> y la <i>Tabla de complejidad y contribución para EO</i> .	EO
	30	– El <i>revisor</i> verifica los totales de <i>EI</i> y <i>EO</i> por cada <i>nivel de complejidad</i> del <i>Formulario 4</i> .	EO
	31	– El <i>revisor</i> analiza el <i>Formulario 3</i> verificando con el <i>LEL &amp; Escenarios</i> de la aplicación los <i>recursos</i> detallados aplicando las <i>Reglas para identificar recursos</i> y las <i>Reglas para contar DET en los recursos</i> y <i>Regla para contar RET en los recursos</i> y verifica la <i>complejidad funcional</i> con la <i>Tabla de complejidad y contribución para recursos</i> .	EO
	32	– El <i>revisor</i> verifica los totales de <i>recursos</i> por cada <i>nivel de complejidad</i> del <i>Formulario 4</i> .	EO
	33	– El <i>revisor</i> verifica los cálculos del <i>Formulario 4</i> y la cuenta total de <i>FPS</i> .	EO
	34	– <b>if</b> existen diferencias con la medición existente <b>then</b> el <i>revisor</i> almacena los resultados en una nueva <i>versión</i> del <i>Formulario 1</i> y/o <i>Formulario 2</i> y/o <i>Formulario 3</i> y/o <i>Formulario 4</i> .	EI
		– <b>restriction:</b> No se cuenta con los formularios de la medición a revisar	D
		– <b>Crear una medición</b>	D
<b>Usar SFP</b>		– <b>Revisar una medición</b>	D
		– <b>restriction:</b> el <i>LEL &amp; Escenarios</i> no fue generado con la herramienta Baseline Mentor Workbench (BMW).	D

Formulario 2	SFP Complejidad de los episodios	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>N° REVISIÓN:</b>
<b>ESCENARIO:</b> Calcular el esfuerzo del proceso de medición		
<b>ID:</b> 1		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 5</i>		<i>esfuerzo</i>
<b>TOTAL</b> 1		<b>TOTAL</b> 1
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Calcular tamaño funcional		
<b>ID:</b> 2		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 5</i>		nombre
		título de la aplicación
		<i>fecha de inicio</i>
		<i>hora de inicio</i>
		paso del <i>proceso de medición</i>
<b>TOTAL</b> 1		<b>TOTAL</b> 5
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Calcular tamaño funcional		
<b>ID:</b> 3		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		cuenta de <i>FPs</i>
<b>TOTAL</b> 1		<b>TOTAL</b> 1
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Calcular tamaño funcional		
<b>ID:</b> 4		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 5</i>		<i>fecha de finalización</i>
		<i>hora de finalización</i>
		intervalo
<b>TOTAL</b> 1		<b>TOTAL</b> 3
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Clasificar episodios		
<b>ID:</b> 5		<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<i>lista de episodios</i>		<i>episodio</i>
<i>Reglas para clasificar episodios</i>		<i>EI / EO</i>
		<i>ID</i>
<b>TOTAL</b> 2		<b>TOTAL</b> 3
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Clasificar episodios		
<b>ID:</b> 6		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 1</i>		<i>episodio</i>
		<i>tipo EI / EO</i>
<b>TOTAL</b> 1		<b>TOTAL</b> 2
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Contar EIs y EOs por nivel de complejidad		
<b>ID:</b> 7		<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		nombre y apellido
		título de la aplicación
<b>TOTAL</b> 1		<b>TOTAL</b> 2
<b>Complejidad</b>		<b>baja</b>

Formulario 2	SFP Complejidad de los episodios	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
<b>N° REVISIÓN:</b>		
<b>ESCENARIO:</b> Contar EIs y EOs por nivel de complejidad		
<b>ID:</b>	8	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		total de <i>EIs complejidad</i> baja
		total de <i>EIs complejidad</i> media
		total de <i>EIs complejidad</i> alta
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>3</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Contar EIs y EOs por nivel de complejidad		
<b>ID:</b>	9	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		total de <i>EOs complejidad</i> baja
		total de <i>EOs complejidad</i> media
		total de <i>EOs complejidad</i> alta
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>3</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Contar recursos por nivel de complejidad		
<b>ID:</b>	10	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		total de <i>recursos de complejidad</i> baja
		total de <i>recursos de complejidad</i> media
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>2</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar complejidad de los episodios		
<b>ID:</b>	11	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 2</i>		nombre y apellido
		fecha
		<i>escenario</i>
		<i>ID</i>
		<i>tipo</i>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>5</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar complejidad de los episodios		
<b>ID:</b>	12	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 2</i>		numero de <i>DET</i>
		numero de <i>RTR</i>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>2</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar complejidad de los episodios		
<b>ID:</b>	13	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 2</i>		<i>complejidad funcional</i>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>1</b>
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar complejidad de los recursos		
<b>ID:</b>	14	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 3</i>		numero de <i>DET</i>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>1</b>
<b>Complejidad</b>		<b>baja</b>

Formulario 2	SFP Complejidad de los episodios	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>N° REVISIÓN:</b>
<b>ESCENARIO:</b> Determinar complejidad de los recursos		
<b>ID:</b>	15	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 3</i>		cuenta <i>RET</i>
<b>TOTAL</b>	1	<b>TOTAL</b> 1
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar complejidad de los recursos		
<b>ID:</b>	16	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 3</i>		<i>complejidad funcional</i>
<b>TOTAL</b>	1	<b>TOTAL</b> 1
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Determinar contribución en FPs de los episodios		
<b>ID:</b>	17	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 4</i>		resultado
<b>TOTAL</b>	1	<b>TOTAL</b> 1
<b>Complejidad</b>		<b>baja</b>
<b>ESCENARIO:</b> Identificar el límite del sistema		
<b>ID:</b>	18	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 1</i>		nombre y apellido
		fecha
		título de la aplicación
<b>TOTAL</b>	1	<b>TOTAL</b> 2
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Identificar el límite del sistema		
<b>ID:</b>	19	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>LEL &amp; escenarios</i>		<i>episodio</i>
<i>lista de episodios disponibles</i>		<i>escenario</i>
<b>TOTAL</b>	2	<b>TOTAL</b> 2
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Identificar el límite del sistema		
<b>ID:</b>	20	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 1</i>		<i>episodio</i>
<i>lista de episodios disponibles</i>		<i>escenario</i>
<b>TOTAL</b>	2	<b>TOTAL</b> 2
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Identificar episodios		
<b>ID:</b>	21	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<i>Reglas para identificar episodios.</i>		<i>episodio</i>
<i>lista de episodios disponibles</i>		
<b>TOTAL</b>	2	<b>TOTAL</b> 1
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Identificar episodios		
<b>ID:</b>	22	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<i>Formulario 1</i>		<i>autor</i>
		título de la aplicación



Formulario 2	SFP Complejidad de los episodios	
TÍTULO DE LA APLICACIÓN: SFP Tool		
AUTOR: Mabel		REVISOR:
ETAPA DEL PROYECTO: Elicitación		FECHA DE REVISIÓN:
		Nº REVISIÓN:
		Fecha
		<i>escenario</i>
		<i>ID</i>
		<i>episodio</i>
TOTAL	1	TOTAL 6
Complejidad		Baja
ESCENARIO: Identificar recursos		
ID:	23	TIPO: EI
RTR		DET
<i>Formulario 3</i>		nombre y apellido
		título de la aplicación
TOTAL	1	TOTAL 2
Complejidad		Baja
ESCENARIO: Identificar recursos		
ID:	24	TIPO: EO
RTR		DET
<i>Formulario 1</i>		<i>recurso</i>
<i>Reglas para identificar recursos</i>		<i>episodio</i>
<i>LEL &amp; escenarios</i>		
TOTAL	3	TOTAL 2
Complejidad		Baja
ESCENARIO: Identificar recursos		
ID:	25	TIPO: EO
RTR		DET
<i>Formulario 3</i>		<i>recurso</i>
TOTAL	1	TOTAL 1
Complejidad		Baja
ESCENARIO: Identificar recursos		
ID:	26	TIPO: EI
RTR		DET
<i>Formulario 3</i>		<i>recurso</i>
TOTAL	1	TOTAL 1
Complejidad		Baja
ESCENARIO: Revisar una medición		
ID:	27	TIPO: EI
RTR		DET
<i>Formulario 2</i>		nombre y apellido <i>revisor</i>
<i>Formulario 4</i>		<i>fecha de revisión</i>
		<i>Nº de revisión</i>
TOTAL	2	TOTAL 3
Complejidad		Baja
ESCENARIO: Revisar una medición		
ID:	28	TIPO: EO
RTR		DET
<i>Formulario 1</i>		<i>episodio</i>
<i>LEL y Escenarios</i>		<i>tipo</i>
<i>Reglas para identificar episodios</i>		<i>ID</i>
<i>Reglas para clasificar episodios</i>		<i>escenario</i>
TOTAL	4	TOTAL 4
Complejidad		Media
ESCENARIO: Revisar una medición		

Formulario 2	SFP Complejidad de los episodios	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>N° REVISIÓN:</b>
<b>ID:</b>	29	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<u>Formulario 2</u>		<u>ID</u>
<u>Reglas para contar DET en EIs y EOs</u>		<u>tipo del episodio (EI / EO)</u>
<u>Reglas para contar DET en los recursos</u>		<u>DETs</u>
<u>Tabla de complejidad y Contribución para EI</u>		<u>RTRs</u>
<u>Tabla de complejidad y contribución para EO</u>		número de <u>DETs</u>
		número de <u>RTRs</u>
		<u>complejidad</u>
<b>TOTAL</b>	<b>5</b>	<b>TOTAL</b> 7
<b>Complejidad</b>		<b>Alta</b>
<b>ESCENARIO:</b> Revisar una medición		
<b>ID:</b>	30	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<u>Formulario 4</u>		total <u>EI</u>
		total <u>EO</u>
		<u>complejidad</u>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> 3
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Revisar una medición		
<b>ID:</b>	31	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<u>Formulario 3</u>		<u>recursos</u>
<u>LEL y Escenarios</u>		<u>DETs</u>
<u>Reglas para identificar recursos</u>		número de <u>DETs</u>
<u>Reglas para contar DET en los recursos</u>		número de <u>RETs</u>
<u>Regla para contar RET en los recursos</u>		<u>Complejidad</u>
<u>Tabla de complejidad y contribución para recursos</u>		
<b>TOTAL</b>	<b>6</b>	<b>TOTAL</b> 5
<b>Complejidad</b>		<b>Media</b>
<b>ESCENARIO:</b> Revisar una medición		
<b>ID:</b>	32	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<u>Formulario 4</u>		total <u>recursos</u>
		<u>complejidad</u>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> 2
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Revisar una medición		
<b>ID:</b>	33	<b>TIPO:</b> EO
<b>RTR</b>		<b>DET</b>
<u>Formulario 4</u>		número de <u>EIs</u> por <u>nivel de complejidad</u>
		número de <u>EOs</u> por <u>nivel de complejidad</u>
		número de <u>recursos</u> por <u>nivel de complejidad</u>
		cuenta de <u>FPs</u>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> 4
<b>Complejidad</b>		<b>Baja</b>
<b>ESCENARIO:</b> Revisar una medición		
<b>ID:</b>	34	<b>TIPO:</b> EI
<b>RTR</b>		<b>DET</b>
<u>Formulario 1 y/o Formulario 2 y/o Formulario 3 y/o Formulario 4</u>		Resultados

<b>Formulario 2</b>	<b>SFP</b> <b>Complejidad de los episodios</b>	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>N° REVISIÓN:</b>
<b>TOTAL</b>	<b>1</b>	<b>TOTAL</b> <b>1</b>
<b>Complejidad</b>		<b>Baja</b>

<b>Formulario 3</b>	<b>SFP</b> <b>Complejidad de los recursos</b>	
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>
		<b>N° REVISIÓN:</b>
<b>Nombre:</b>	<i>Formulario 5</i>	<b>RET</b> <b>1</b>
<b>DET</b>		
título de la aplicación		
<i>autor</i>		
paso del <i>proceso de medición</i>		
<i>hora de inicio</i>		
<i>hora de finalización</i>		
intervalo de tiempo		
<i>fecha de inicio</i>		
<i>fecha de finalización</i>		
total en horas		
<b>TOTAL</b>	<b>9</b>	<b>Complejidad</b> <b>Baja</b>
<b>Nombre:</b>	<i>Formulario 4</i>	<b>RET</b> <b>1</b>
<b>DET</b>		
<i>autor</i>		
título de la aplicación		
etapa del proyecto		
Fecha		
número de <i>EIs</i> de <i>complejidad</i> baja		
número de <i>EIs</i> de <i>complejidad</i> media		
número de <i>EIs</i> de <i>complejidad</i> alta		
número de <i>EOs</i> de <i>complejidad</i> baja		
número de <i>EOs</i> de <i>complejidad</i> media		
número de <i>EOs</i> de <i>complejidad</i> alta		
número de <i>recursos</i> de <i>complejidad</i> baja		
número de <i>recursos</i> de <i>complejidad</i> media		
total por <i>complejidad</i> baja de <i>EIs</i>		
total por <i>complejidad</i> media de <i>EIs</i>		
total por <i>complejidad</i> alta de <i>EIs</i>		
total por <i>complejidad</i> baja de <i>EOs</i>		
total por <i>complejidad</i> media de <i>EOs</i>		
total por <i>complejidad</i> alta de <i>EOs</i>		
total por <i>complejidad</i> baja de <i>recursos</i>		
total por <i>complejidad</i> media de <i>recursos</i>		
cuenta de <i>FPs</i>		
<b>TOTAL</b>	<b>20</b>	<b>Complejidad</b> <b>Baja</b>
<b>Nombre:</b>	<i>Reglas para clasificar episodios</i>	<b>RET</b> <b>1</b>
<b>DET</b>		

Formulario 3	SFP	
Complejidad de los recursos		
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel	<b>REVISOR:</b>	
<b>ETAPA DEL PROYECTO:</b> Elicitación	<b>FECHA DE REVISIÓN:</b>	
	<b>N° REVISIÓN:</b>	
regla para clasificar <i>episodio</i> como <i>EI</i>		
regla para clasificar <i>episodio</i> como <i>EO</i>		
propósito principal		
nro. regla		
<b>TOTAL</b>	<b>4</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <i>Formulario 1</i>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<i>autor</i>		
título de la aplicación		
Fecha		
<i>escenario</i>		
<i>episodio</i>		
<i>ID episodio</i>		
<i>tipo episodio</i>		
<b>TOTAL</b>	<b>7</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <i>Formulario 2</i>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<i>autor</i>		
título de la aplicación		
<i>revisor</i>		
<i>fecha de revisión</i>		
<i>nro. de revisión</i>		
<i>escenario</i>		
<i>ID</i>		
<i>tipo (EI / EO)</i>		
<i>DET</i>		
<i>RTR</i>		
número de <i>DETs</i>		
número de <i>RTRS</i>		
<i>Complejidad</i>		
<b>TOTAL</b>	<b>13</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <i>Formulario 3</i>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<i>autor</i>		
título de la aplicación		
<i>revisor</i>		
<i>fecha de revisión</i>		
<i>nro. de revisión</i>		
nombre del <i>recurso</i>		
<i>DET</i>		
número de <i>DETs</i>		
número de <i>RETs</i>		
<i>complejidad del recurso</i>		
<b>TOTAL</b>	<b>10</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <i>Reglas para contar DET en Els y EOs</i>	<b>RET</b>	<b>1</b>
<b>DET</b>		
Regla		
nro regla		

Formulario 3	SFP	
<b>Complejidad de los recursos</b>		
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool		
<b>AUTOR:</b> Mabel	<b>REVISOR:</b>	
<b>ETAPA DEL PROYECTO:</b> Elicitación	<b>FECHA DE REVISIÓN:</b>	
	<b>N° REVISIÓN:</b>	
<b>TOTAL</b>	<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>Reglas para contar RTR en EIs y EOs</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<u>Regla</u>		
nro regla		
<b>TOTAL</b>	<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>Tabla de Complejidad y Contribución para EI</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<u>Complejidad</u>		
número de <u>DETs</u>		
número de <u>RTRs</u>		
valor numérico		
<b>TOTAL</b>	<b>4</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>Tabla de Complejidad y Contribución para EO</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<u>Complejidad</u>		
número de <u>DETs</u>		
número de <u>RTRs</u>		
valor numérico		
<b>TOTAL</b>	<b>4</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>Reglas para contar DET en los recursos</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<u>Regla</u>		
nro regla		
<b>TOTAL</b>	<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>LEL &amp; escenarios</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
nombre del <u>símbolo</u>		
Noción		
Impacto		
nombre del <u>escenario</u>		
Objetivo		
Contexto		
Actores		
<u>Recursos</u>		
<u>Episodios</u>		
<b>TOTAL</b>	<b>9</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>lista de episodios disponibles</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		
<u>Episodio</u>		
<u>Escenario</u>		
<b>TOTAL</b>	<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b> <u>regla para contar RET en los recursos</u>	<b>RET</b>	<b>1</b>
<b>DET</b>		

Formulario 3	SFP		
Complejidad de los recursos			
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool			
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>	
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>	
		<b>N° REVISIÓN:</b>	
Regla			
nro regla			
<b>TOTAL</b>		<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b>	<i>Reglas para identificar episodios.</i>	<b>RET</b>	<b>1</b>
<b>DET</b>			
Regla			
nro regla			
<b>TOTAL</b>		<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b>	<i>Reglas para identificar recursos</i>	<b>RET</b>	<b>1</b>
<b>DET</b>			
Regla			
nro regla			
<b>TOTAL</b>		<b>2</b>	<b>Complejidad Baja</b>
<b>Nombre:</b>	<i>Tabla de complejidad y contribución para recursos</i>	<b>RET</b>	<b>1</b>
<b>DET</b>			
<u>Complejidad</u>			
número de <i>DETs</i>			
número de <i>RETs</i>			
valor numérico			
<b>TOTAL</b>		<b>4</b>	<b>Complejidad Baja</b>

Formulario 4	SFP			
Cálculo de FP no Ajustados				
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool				
<b>AUTOR:</b> Mabel		<b>REVISOR:</b>		
<b>ETAPA DEL PROYECTO:</b> Elicitación		<b>FECHA DE REVISIÓN:</b>		
Tipo de Función	Complejidad Funcional		Totales por complejidad	Totales por Tipo de Función
Recursos	17	Baja x	7	119
		Media x	10	0
				119
EI	24	Baja x	3	72
		Media x	4	0
		Alta x	6	0
				72
EO	7	Baja x	3	21
	2	Media x	4	8
	1	Alta x	6	6
				35
<b>Total FP no Ajustados (SFP)</b>				<b>226</b>

Formulario 5		SFP			
		Horas de trabajo			
<b>TÍTULO DE LA APLICACIÓN:</b> SFP Tool					
<b>AUTOR:</b> Mabel					
Paso	Fecha	Inicio (hora)	Fin (hora)	Intervalo (hora)	Comentarios
	14/06/2008	21:23	21:37	00:14:00	armar planillas
	14/06/2008	21:45	21:50	00:05:00	armar planillas
1	14/06/2008	22:02	22:12	00:10:00	
2, 3	15/06/2008	11:10	11:27	00:17:00	
2, 3	15/06/2008	21:00	21:23	00:23:00	revisar clasificación
4	15/06/2008	22:31	22:55	00:24:00	
4	16/06/2008	15:29	15:49	00:20:00	
4	12/10/2008	19:04	19:19	00:15:00	
4	12/10/2008	19:26	19:30	00:04:00	
4	12/10/2008	21:33	22:20	00:47:00	
4	13/10/2008	17:23	18:18	00:55:00	
5	13/10/2008	19:44	20:02	00:18:00	
6	13/10/2008	20:25	20:47	00:22:00	
6	14/10/2008	15:10	15:31	00:21:00	
6	15/10/2008	10:38	10:49	00:11:00	
6	15/10/2008	11:09	11:17	00:08:00	
7	15/10/2008	11:34	11:53	00:19:00	
<b>Total</b>				<b>05:33:00</b>	

### D.3. Especificación de Requerimientos del Software

#### Sistema SFP Tool Versión 1.0

##### 1. Introducción

###### 1.1. Propósito

Este documento se restringe a la descripción de los requerimientos funcionales del software SFP Tool. Fue organizado de acuerdo a las prácticas recomendadas por el estándar IEEE 830-1998<sup>54</sup> [IEEE 1998]. Este documento debe ser utilizado por los integrantes del equipo de proyecto que implementarán y verificarán el correcto funcionamiento del sistema.

###### 1.2. Alcance

El sistema de SFP Tool es una herramienta que soporta el proceso SFP. SFP permite medir el tamaño funcional de los escenarios basados en el enfoque de Leite. El sistema permitirá realizar una nueva medición, continuar una medición existente o revisar una medición. El sistema no manejará el versionado de mediciones.

<sup>54</sup> Las secciones Acrónimos y Referencias, que deben estar incluidas de acuerdo a lo especificado en este estándar, fueron omitidas en esta parte del documento pues están incluidas en las respectivas secciones de esta tesis.

## **2. Descripción general**

### **2.1. Perspectiva del producto**

El sistema SFP Tool es un sistema informático independiente que permitirá semiautomatizar el proceso de medición, actualmente realizado en forma manual. El software trabaja con el LEL y Escenarios (L&E) de un sistema de software y maneja un conjunto de formularios que documentan los resultados intermedios y final de la medición. Además registra y reporta el esfuerzo (en horas) del proceso. El sistema es semiautomatizado porque existen pasos del proceso que necesariamente requieren la intervención del usuario. En futuras versiones el sistema debería evolucionar para permitir el manejo del versionado de mediciones.

### **2.2. Funciones del producto**

- Crear una nueva medición.
- Modificar una medición existente.
- Visualizar una medición.
- Cerrar medición.

### **2.3. Características del usuario**

El sistema SFP Tool tiene un único tipo de usuario. El Usuario debe tener experiencia en el uso de la técnica SFP, preferentemente haber aplicado previamente el proceso manual.

### **2.4. Restricciones**

- El software sólo debe procesar documentos de L&E generados por la herramienta BMW [Antonelli *et al.* 1999].

## **3. Requerimientos específicos**

### **3.1. Requerimientos de las interfaces de usuario**

- El sistema deberá utilizar interfaces gráficas que reproduzcan fielmente los formularios usados para el proceso manual.
- El sistema deberá presentar un mensaje de confirmación cuando el usuario intenta cerrar la medición.
- El sistema deberá presentar un mensaje de error cuando un procesamiento requerido no pudo ser realizado.

### **3.2. Operaciones**

- El sistema deberá trabajar en dos modos dependiendo del paso de proceso: automático o realizado por el usuario. La siguiente tabla presenta cada paso del proceso y el modo de operación.



<b>Paso</b>	<b>Modo Operación</b>
1. Construir la lista con los episodios de todos los escenarios	Automático
2. Identificar episodios *	Usuario
3. Clasificar episodios como EI y EO*	Usuario
4. Construir la lista de EI y EO Insertar, modificar o borrar DET y RTR * Contar DET y RTR Determinar complejidad	Automático usuario automático automático
5. Identificar recursos	Automático
6. Construir la lista de recursos Insertar, modificar o borrar DET * Contar DET Determinar complejidad	Automático usuario automático automático
7. Calcular el tamaño funcional	Automático

\* el registro y cálculo del esfuerzo es automático.

### 3.3. Requerimientos funcionales

#### 3.3.1. Generales

**RF1.** El sistema deberá incluir las operaciones necesarias para gestionar las mediciones. Las opciones son: nueva medición, abrir medición existente, guardar medición.

**RF2.** El sistema permitirá visualizar y almacenar los diferentes formularios usados durante el proceso.

**RF3.** El sistema permitirá cargar el archivo con los datos del L&E.

**RF3.1.** El sistema deberá presentar un mensaje de error si los datos son incorrectos (no corresponden a un L&E).

**RF4.** El sistema permitirá visualizar el L&E cuando el usuario lo requiera.

**RF5.** El sistema permitirá visualizar las Reglas cuando el usuario lo requiera.

**RF6.** El sistema permitirá almacenar la medición en cualquier paso del proceso y estado de la medición. Los estados de la medición son *completa*, *incompleta*. Una medición está completa si fueron calculados los FPs.

**RF6.1.** El sistema deberá presentar un mensaje de error si no se pudo realizar la operación.

**RF7.** El sistema permitirá cerrar la medición en cualquier paso del proceso y estado de la medición (completa o incompleta).

**RF7.1.** El sistema deberá presentar un mensaje de confirmación.

**RF8.** El sistema deberá recuperar la medición en el estado en que fue almacenada.

**RF9.** El sistema permitirá la revisión (modificación) si la medición está completa.

**RF10.** El sistema deberá proporcionar una pantalla de ayuda que describe las funciones del sistema.

**RF11.** El sistema permitirá exportar los formularios a una planilla de cálculo.

### 3.3.2. Nueva medición

**RF12.** El sistema deberá solicitar al usuario el ingreso del nombre del autor de la medición.

**RF13.** El sistema deberá solicitar al usuario el ingreso de la etapa del proyecto en que se realiza la medición.

**RF14.** El sistema deberá crear y presentar la pantalla del Formulario 1 con los siguientes datos: título de la aplicación (extraído del L&E), autor de la medición (ingresado por el usuario), fecha actual (suministrada por el sistema), nombre de cada escenario y sus respectivos episodios (extraídos del L&E).

**RF15.** El sistema deberá crear el Formulario 5 con el título de la aplicación y el autor de la medición.

### 3.3.3. Nueva medición / Medición existente

**RF16.** Si el usuario inicia (o reinicia en el caso de un paso incompleto) un paso del proceso que requiere su intervención (pasos 2, 3, 4 y 6), el sistema deberá agregar el número de paso del proceso, la fecha y hora de inicio (suministradas por el sistema) en el Formulario 5.

**RF17.** El sistema permitirá que el usuario identifique y clasifique los episodios en la pantalla del Formulario 1 mediante la selección de una opción desde una lista de opciones. Las opciones son: EI, EO y D.

**RF18.** El sistema deberá asignar un número de ID (en orden secuencial comenzando por 1) a cada episodio clasificado como EI y EO.

**RF19.** El sistema deberá controlar si todos los episodios del Formulario 1 están clasificados y habilitar el siguiente paso del proceso.

**RF20.** El sistema deberá crear y presentar la pantalla del Formulario 2 con el nombre del autor, el título de la aplicación y por cada EI y EO (del Formulario 1) el ID, el tipo y el nombre del escenario al que pertenece.

**RF21.** El sistema deberá presentar una lista de opciones para la edición de DET y RTR en la pantalla del Formulario 2. Las opciones son: Insertar fila, Eliminar fila, Agregar, Reemplazar y Borrar un elemento (DET o RTR).

**RF21.1.** El sistema deberá proveer facilidades para obtener los DET y RTR desde el L&E.

**RF21.2.** El sistema deberá proveer facilidades para agregar los DET y RTR en la pantalla del Formulario 2.

**RF22.** El sistema deberá calcular y registrar el total de DET y RTR en el Formulario 2.

**RF23.** El sistema deberá determinar automáticamente y registrar la complejidad de EIs y EOs en el Formulario 2.

**RF23.1.** La complejidad de EIs y EOs será determinada usando el total de DET y RTR como entrada a la Tabla de complejidad y contribución para EI y a la Tabla de complejidad y contribución para EO.

**RF24.** El sistema deberá controlar si se asignó complejidad a todas las EIs y EOs en la pantalla del Formulario 2 y habilitar el siguiente paso del proceso.

**RF25.** El sistema deberá almacenar los DETs y RTRs de cada EI y EO en el Formulario 2.

**RF26.** El sistema deberá crear y presentar la pantalla del Formulario 3 con el nombre del autor, el título de la aplicación y el nombre de todos los recursos (no repetidos) referenciados por las EIs y EOs del Formulario 1.

**RF27.** El sistema deberá presentar una lista de opciones para la edición de DET en la pantalla del Formulario 3. Las opciones son: Insertar fila, Eliminar fila, Agregar, Reemplazar y Borrar un elemento (DET).

**RF27.1.** El sistema deberá proveer facilidades para obtener los DET desde el L&E.

**RF27.2.** El sistema deberá proveer facilidades para agregar los datos del L&E en la pantalla del Formulario 3.

**RF28.** El sistema deberá calcular y registrar el total de DET en el Formulario 3.

**RF29.** El sistema deberá determinar automáticamente y registrar la complejidad de los recursos en el Formulario 3.

**RF29.1.** La complejidad será determinada usando el total de DET y 1 RET como entrada a la Tabla de complejidad y contribución para recursos.

**RF30.** El sistema deberá controlar si se asignó complejidad a todos los recursos en la pantalla del Formulario 3 y habilitar el siguiente paso del proceso.

**RF31.** El sistema deberá almacenar los DETs de cada recurso en el Formulario 3.

**RF32.** El sistema deberá crear y presentar la pantalla del Formulario 4 con el nombre del autor, el título de la aplicación, la etapa del proyecto, la fecha actual (suministrada por el sistema), el número de EIs y EOs por cada nivel de complejidad (obtenido del Formulario 2) y el número de recursos por cada nivel de complejidad (Formulario 3), la contribución en FP por cada nivel de complejidad de las EIs, EOs y recursos, el total de FP por tipo de función (EI, EO, recurso) y el total de FP. Las contribuciones en FP y los totales por tipo de función y total de FP deberán ser calculados.

**RF33.** El sistema deberá inhabilitar la edición en la pantalla del Formulario 4.

**RF34.** Si el usuario finaliza un paso del proceso (o el proceso completo), el sistema deberá agregar la hora de finalización (suministrada por el sistema) en el Formulario 5.

**RF35.** El sistema deberá presentar la pantalla del Formulario 5 con el número de paso del proceso, la fecha, la hora de inicio y finalización, el intervalo de tiempo (hora finalización - hora inicio) y el tiempo total (suma de los intervalos).

**RF36.** El sistema deberá inhabilitar la edición en la ventana del Formulario 5.

**RF37.** Al finalizar la medición el sistema deberá inhabilitar la edición en las ventanas de los Formularios 1, 2 y 3.

#### **3.3.4. Revisión**

**RF38.** El sistema sólo permitirá la revisión de los datos que afectan la complejidad de EIs, EOs (RTRs y DETs) y recursos (DETs).

**RF39.** Si es seleccionada la opción revisión, el sistema deberá solicitar al usuario el ingreso del nombre del autor.

**RF40.** El sistema deberá habilitar la edición en el respectivo formulario (Formulario 2 o 3).

**RF41.** Si se modifican los datos en la pantalla del Formulario 2 el sistema deberá almacenar el nombre del autor de la revisión, el número de revisión y la fecha de revisión en el Formulario 2.

**RF42.** Si se modifican los datos en la pantalla del Formulario 3 el sistema deberá almacenar el nombre del autor de la revisión, el número de revisión y la fecha de revisión en el Formulario 3.

**RF43.** El sistema deberá agregar el número de paso, la fecha de revisión, el nombre del revisor (obtenidos del formulario correspondiente) y la hora de inicio de la revisión (suministrada por el sistema) en el Formulario 5.

**RF44.** El sistema permitirá continuar el proceso a partir del paso en que se realicen las modificaciones (sección 3.5.3).

**RF45.** El sistema deberá presentar la pantalla del Formulario 5 con el número de paso del proceso, la fecha, la hora de inicio y finalización, el número de paso de la revisión, la fecha de revisión, la hora de inicio y finalización de la revisión, el nombre del revisor, el intervalo de tiempo (hora finalización - hora inicio) y el tiempo total (suma de los intervalos).

## D.4. Medición IFPUG

### D.4.1. Archivos lógicos

ILF	RET	DET	Complejidad
Formulario 1	1	7	baja
Formulario 2	1	6	baja
Formulario 3	1	5	baja
Formulario 4	1	24	baja
Formulario 5	1	6	baja

EIF	RET	DET	Complejidad
L&E	4	13	Baja
Ayuda	1	2	Baja
Reglas SFP	1	2	Baja

### D.4.2. Transacciones

Se indica el requerimiento funcional al que está asociada.

EI	FTR	DET	Complejidad
RF14	2	9	media
RF15	1	1	baja
RF16	1	1	baja
RF17	1	3	baja
RF20	2	6	media
RF21	2	8	media
RF21	2	8	media
RF21	1	4	baja
RF21	1	4	baja
RF21	1	6	baja
RF26	2	4	baja
RF27	2	8	media
RF27	1	4	baja
RF27	1	5	baja
RF33	0	1	baja
RF34	1	1	baja
RF41	1	3	baja
RF42	1	5	baja
RF43	1	1	baja

EO	FTR	DET	Complejidad
RF18	1	1	baja
RF22	1	2	baja
RF23	1	1	baja
RF28	1	1	baja
RF29	1	1	baja
RF32	3	27	alta
RF35	1	2	baja

EQ	FTR	DET	Complejidad
RF2	1	9	baja
RF2	1	15	baja
RF2	1	11	baja
RF2	1	25	media
RF2	1	14	baja
RF4	1	14	baja
RF5	1	3	baja
RF9	1	1	baja
RF10	1	4	baja
RF19	1	1	baja
RF24	1	1	baja
RF30	1	1	baja

**D.4.3. Resultado de la medición IFPUG**

Tipo de Función	Complejidad Funcional			Totales por complejidad	Totales por Tipo de Función
ILF	5	Baja	x	7	35
	0	Media	x	10	0
	0	Alta	x	15	0
					35
EIF	3	Baja	x	5	15
	0	Media	x	7	0
	0	Alta	x	10	0
					15
EI	14	Baja	x	3	42
	5	Media	x	4	20
	0	Alta	x	6	0
					62
EQ	11	Baja	x	3	33
	1	Media	x	4	4
	0	Alta	x	6	0
					37
EO	6	Baja	x	4	24
	0	Media	x	5	0
	1	Alta	x	7	7
					31
<b>Total FP no Ajustados</b>					<b>180</b>



## Apéndice E

### Software SFP Tool

La ejecución del procedimiento SFP es soportada por un proceso que consta de varios pasos. Con el objetivo de facilitar las mediciones fue desarrollado el software SFP Tool que permite automatizar dicho proceso. En este apéndice se presenta una descripción de las características de la herramienta y las actividades que deben ser realizadas para estimar los FP de un proyecto de software.

Esta herramienta permitió integrar en un software los diferentes utilitarios usados para la medición manual. Su aplicación en la práctica demostró que, además de reducirse el esfuerzo como consecuencia de la automatización de ciertos pasos del proceso, se evitan los errores de cálculo, se agiliza el acceso a los documentos requeridos durante el proceso y se favorece la detección de errores u omisiones en los escenarios.

#### E.1. El software SFP Tool

Esta herramienta, diseñada para soportar el proceso SFP, fue implementada en lenguaje visual Delphi, usando dos componentes principales: *Formula One* © Visual Components, Inc. para el manejo de formularios y *RichView*© 1997-2004 by Sergey Tkachenko para el hipertexto.

El software usa como entrada el archivo .rtf generado por la herramienta BMW (Baseline Menthor Workbench) [Antonelli *et al.* 1999], el que es usado para cargar el L&E del proyecto software. Hay ciertas partes del proceso que son realizadas automáticamente y otras que requieren la intervención del usuario. Como resultado del proceso se genera un archivo de extensión .vts - también puede ser exportado a una planilla *Microsoft*© *Excel* - en donde se presentan todos los formularios generados por la aplicación.

#### E.2. Uso de SFP Tool

##### E.2.1. Iniciar una sesión de trabajo

Al iniciar la aplicación, pueden observarse las siguientes opciones en el menú *Archivo*:

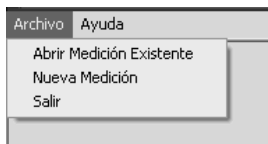


Figura E.1. Opciones del menú Archivo

- Nueva Medición: es utilizada para iniciar una medición nueva.
- Abrir Medición Existente: es utilizada para continuar, visualizar o revisar una medición realizada previamente.

##### E.2.2. Realizar una nueva medición

Esta opción permite comenzar una medición. La siguiente pantalla contiene una breve explicación sobre cómo cargar el archivo .rtf del L&E.



Una vez abierto el archivo del L&E, quedará habilitada la opción *Medir*. Al seleccionarla solicitará el nombre del autor de la medición para registrarlo en los formularios.

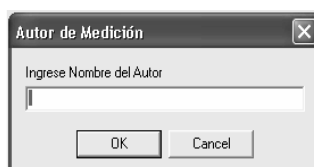


Figura E.3. Cuadro de diálogo para ingresar el nombre del autor

Se generará automáticamente la Planilla de Episodios donde estarán almacenados todos los escenarios con sus respectivos episodios, los que deberán ser clasificados utilizando las reglas definidas para este paso del proceso.

SFP Tool			
Medición Visor L&E Escenarios Ayuda			
Formulario 1		SFP	
Planilla de episodios			
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco			FECHA: 27/11/2008
AUTOR: Gabriela			
Escenario	ID	Episodio	Tipo
Confecionar aviso de deuda vencida			
		El Empleado completa el aviso de deuda vencida con la fecha de emisión, plazo en días y los datos del informe de deuda del cliente.	
		El empleado lo entrega a Gerencia.	
		if Gerencia firmó el aviso de deuda vencida then El Empleado lo envía al Cliente por correo electrónico.	EI EO D
		El Empleado archiva una copia del Aviso de Deuda Vencida en la Carpeta de Crédito.	
confeccionar avisos de préstamos			
		if fecha de vencimiento de la cuota del balance de préstamos es igual a fecha de vencimiento de Informe de Deuda then El empleado obtiene los datos del Cliente y tipo de préstamo que está en mora.	
		Confecionar aviso de deuda vencida.	

Figura E.4. Ventana del Formulario 1

Al seleccionar un episodio y pulsar el botón derecho del mouse se visualizará una lista desplegable ubicada sobre la columna Tipo, donde podrá ser seleccionada una de las opciones disponibles (EI, EO, D). La opción seleccionada se almacenará en dicha columna y automáticamente a los episodios clasificados como EI y EO se les asignará un número de ID correlativo (Figura E.5).



Formulario 1		SFP	
		Planilla de episodios	
TITULO DE LA APLICACION: Sistema de Notificaciones a Clientes de un Banco		FECHA: 20/11/2008	
AUTOR: Gabriela			
Escenario	ID	Episodio	Tipo
Confeccionar aviso de deuda vencida			
	1	El Empleado completa el aviso de deuda vencida con la fecha de emisión, plazo en días y los datos del informe de deuda del cliente.	EI
		El empleado lo entrega a Gerencia.	D
		if Gerencia firmó el aviso de deuda vencida then El Empleado lo envía al Cliente por correo electrónico.	D
	2	El Empleado archiva una copia del Aviso de Deuda Vencida en la Carpeta de Crédito.	EI
confeccionar avisos de préstamos			
	3	if fecha de vencimiento de la cuota del balance de préstamos es igual a fecha de vencimiento de Informe de Deuda then El empleado obtiene los datos del Cliente y tipo de préstamo que está en mora.	EO
		Confeccionar aviso de deuda vencida.	D

Figura E.5. Ventana del Formulario 1 con episodios clasificados

Las reglas que sirven de guía para el proceso de medición están disponibles para consulta mediante el botón Reglas, que permite visualizar una pantalla como la siguiente (Figura E.6):

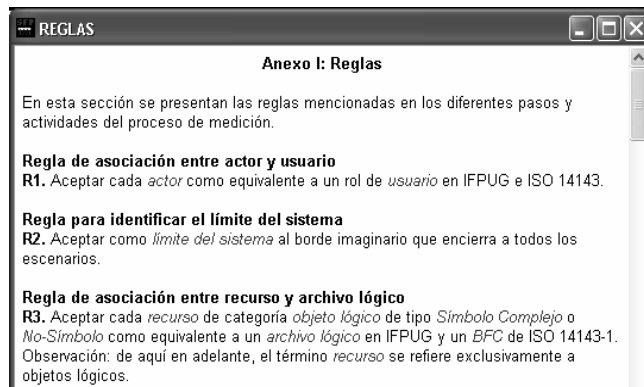



Figura E.6. Visor de Reglas

Al completar la clasificación de los episodios se habilita la opción *Siguiente* , que permite avanzar al próximo paso.

El Formulario 2 con todos los episodios que fueron clasificados como EI y EO es generado automáticamente (Figura E.7).

Formulario 2		SFP	
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco		Complejidad de los episodios	
AUTOR: Gabriela		REVISOR:	
		FECHA DE REVISIÓN:	
		N° REVISIÓN:	
ESCENARIO: Confeccionar aviso de deuda vencida			
ID: 1		TIPO: EI	
	RTR		DET
Aviso de Deuda Vencida		datos del Cliente	
		fecha de emisión	
		fecha de vencimiento	
		número de cuota	
		Plazo en días	
		tipo de préstamo	
		capital	
		interés	
		fecha de pago	
Informe de Deuda		total de la deuda	
TOTAL: 2		TOTAL: 10	
Complejidad:		Media	
ESCENARIO: Confeccionar aviso de deuda vencida			
ID: 2		TIPO: EI	
	RTR		DET
Carpeta de Crédito		Aviso de Deuda Vencida	
Aviso de Deuda Vencida			
TOTAL: 2		TOTAL: 1	
Complejidad:		Baja	

Figura E.7. Ventana del Formulario 2

En este formulario es posible acceder al menú contextual que presenta las opciones *Insertar Fila*, *Eliminar Fila* y *Pegar*.

Por cada EI y EO deben agregarse los RTR y DET en las respectivas filas y columnas. Éstos deben ser seleccionados desde el L&E. Para acceder al mismo se debe seleccionar la opción *Visor LEL y Escenarios* en la barra de menús.

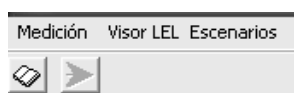


Figura E.8. Opción Visor LEL y Escenarios

Al seleccionar esta opción se visualizará una pantalla como la siguiente, donde se mostrará el contenido del archivo .rtf del L&E del proyecto que se está midiendo. Además, se puede navegar a través del hipertexto para consultar sobre las definiciones del LEL y los escenarios.

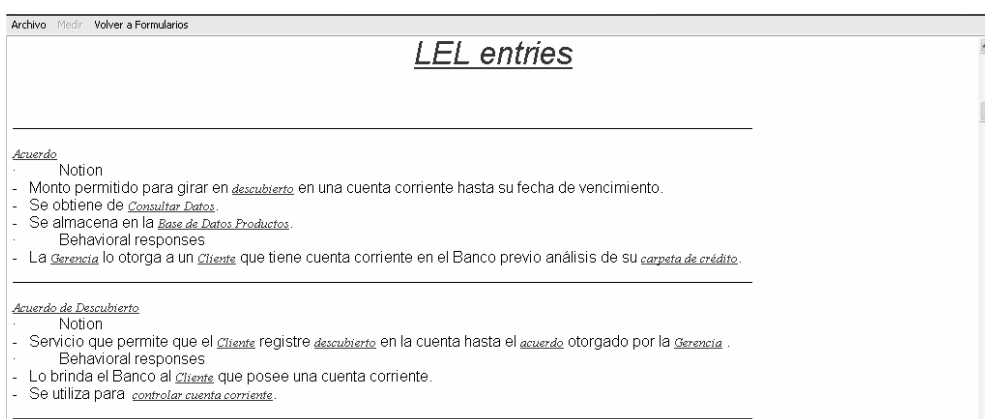


Figura E.9. Pantalla del Visor del LEL y Escenarios

La opción *Volver a Formularios* permite retornar al formulario en que se estaba trabajando.



Figura E.10. Opción Volver a Formularios

En el formulario activo mediante la opción Pegar del menú contextual se podrá agregar el contenido seleccionado en el *Visor del LEL y Escenarios*.

El Total de DET y RTR así como la respectiva complejidad de la EI/EO (baja, media o alta) - determinada desde las Tablas de Complejidad y Contribución para EIs y EOs - es generado automáticamente.

Al completar el Formulario 2 se habilita la opción *Siguiente*, para avanzar al próximo paso.

El Formulario 3 con todos los recursos que son referenciados en los episodios clasificados como EI y EO es generado automáticamente (Figura E.11).

Formulario 3		SFP	
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco		Complejidad de los recursos	
AUTOR: Gabriela		REVISOR:	
		FECHA DE REVISIÓN:	
		N° REVISIÓN:	
Nombre: informe de deuda		RET: 1	
DET			
tipo de préstamo			
número de cuota			
capital			
interés			
fecha de vencimiento			
fecha de pago			
total de la deuda			
TOTAL: 7		Complejidad: Baja	
Nombre: carpeta de crédito		RET: 1	
DET			
antecedentes patrimoniales presentados por el cliente			
fecha de vencimiento			
informes obtenidos por el banco sobre ese Cliente			
tipo de préstamo			
número de cuota			
capital			
interés			
fecha de vencimiento			
fecha de pago			
total de la deuda			
TOTAL: 10		Complejidad: Baja	

Figura E.11. Ventana del Formulario 3

Por cada recurso deben agregarse los DET en la respectiva columna. El Total de DET así como la respectiva complejidad del recurso (baja o media) – determinada desde la Tabla de Complejidad y Contribución para recursos – es generado automáticamente. En el caso de los recursos que no son símbolos del LEL, de acuerdo a la regla R34 se asigna automáticamente complejidad baja.

Al completar el Formulario 3 se habilita la opción *Siguiente*, para avanzar al próximo paso del proceso. Éste es realizado automáticamente, generando el Formulario 4 con los resultados de la estimación (Figura E.12).

Formulario 4		SFP			
		Cálculo de FP no Ajustados			
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco					
AUTOR: Gabriela					
ETAPA DEL PROYECTO: Elicitación de Requerimientos				FECHA: 30/11/2008	
Tipo de Función	Complejidad Funcional		Totales por Complejidad		Totales por Tipo de Función
Recurso	23	Baja	x	7	161
	0	Media	x	10	0
					161
EI	21	Baja	x	3	63
	5	Media	x	4	20
	0	Alta	x	6	0
					83
EO	7	Baja	x	3	21
	1	Media	x	4	4
	0	Alta	x	6	0
					25
Total FP no Ajustados					269

Figura E.12. Ventana del Formulario 4

A largo de todo el proceso en el Formulario 5 se van registrando la fecha y la hora de inicio y finalización, así como el tiempo total insumido por las diferentes actividades.

Formulario 5		SFP			
		Horas de trabajo			
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco					
AUTOR: Gabriela					
Paso	Fecha	Inicio (hora)	Fin (hora)	Intervalo (hora)	Comentarios
Paso 2 y 3	20/11/2008	10:27:36	10:48:11	0:20:35	Paso 1 realizado automáticamente.
Paso 4	20/11/2008	10:48:11	11:7:18	0:19:7	
Paso 4	24/11/2008	9:7:2	9:8:56	0:1:54	
Paso 4	24/11/2008	9:8:56	11:54:59	2:46:3	
Paso 4	30/11/2008	9:37:5	9:50:59	0:13:54	
	30/11/2008	12:37:26	12:50:29	0:13:3	
	30/11/2008	12:53:3	13:18:33	0:25:30	
	30/11/2008	15:55:52	17:48:51	1:52:59	Paso 7 - Medición completa
	05/06/2009	19:59:0			
<b>TOTAL</b>				<b>6:13:5</b>	

Figura E.13. Ventana del Formulario 5

La herramienta permite guardar la medición en cualquier punto del proceso.

**E.2.3. Abrir medición existente**

Una medición existente puede estar incompleta, en cuyo caso se podrá continuar el proceso o puede estar completa y en ese caso se permite la revisión de la medición.

**E.2.4. Continuar una medición**

En este caso, el proceso explicado en la sección E.2.2 comienza a partir del punto en que se encuentre la medición.

**E.2.5. Revisar una medición**

Sólo se permite la revisión de los Formularios 2 y 3 de una medición existente. En caso de efectuarse modificaciones se deben realizar los pasos del proceso a partir del punto en que se realizaron los cambios. Para realizar una revisión se requiere ingresar el nombre del revisor, el que quedará almacenado en el respectivo formulario junto con la fecha y el número de la revisión. En el Formulario 5 quedará registrado el nombre del revisor y el paso del proceso en que se efectuó la revisión.

Formulario 2		SFP	
		Complejidad de los episodios	
TÍTULO DE LA APLICACIÓN: Sistema de Notificaciones a Clientes de un Banco			
AUTOR: Gabriela		REVISOR: Gabriela Oriana	
		FECHA REVISIÓN: 01/12/2008	
		REVISIÓN N°: 1	
ESCENARIO: Confeccionar aviso de deuda vencida			
ID: 1		TIPO: EI	
	RTR		DET
Aviso de Deuda Vencida		datos del Cliente	
		fecha de emisión	

Figura E.14. Ventana del Formulario 2 con datos de la revisión

**E.2.6. Ayuda**

La herramienta cuenta con un sistema de ayuda, que explica como crear, visualizar y revisar una medición existente, en que consisten y para que se usan los distintos formularios que se manejan y los pasos del proceso SFP (Figura E.15).

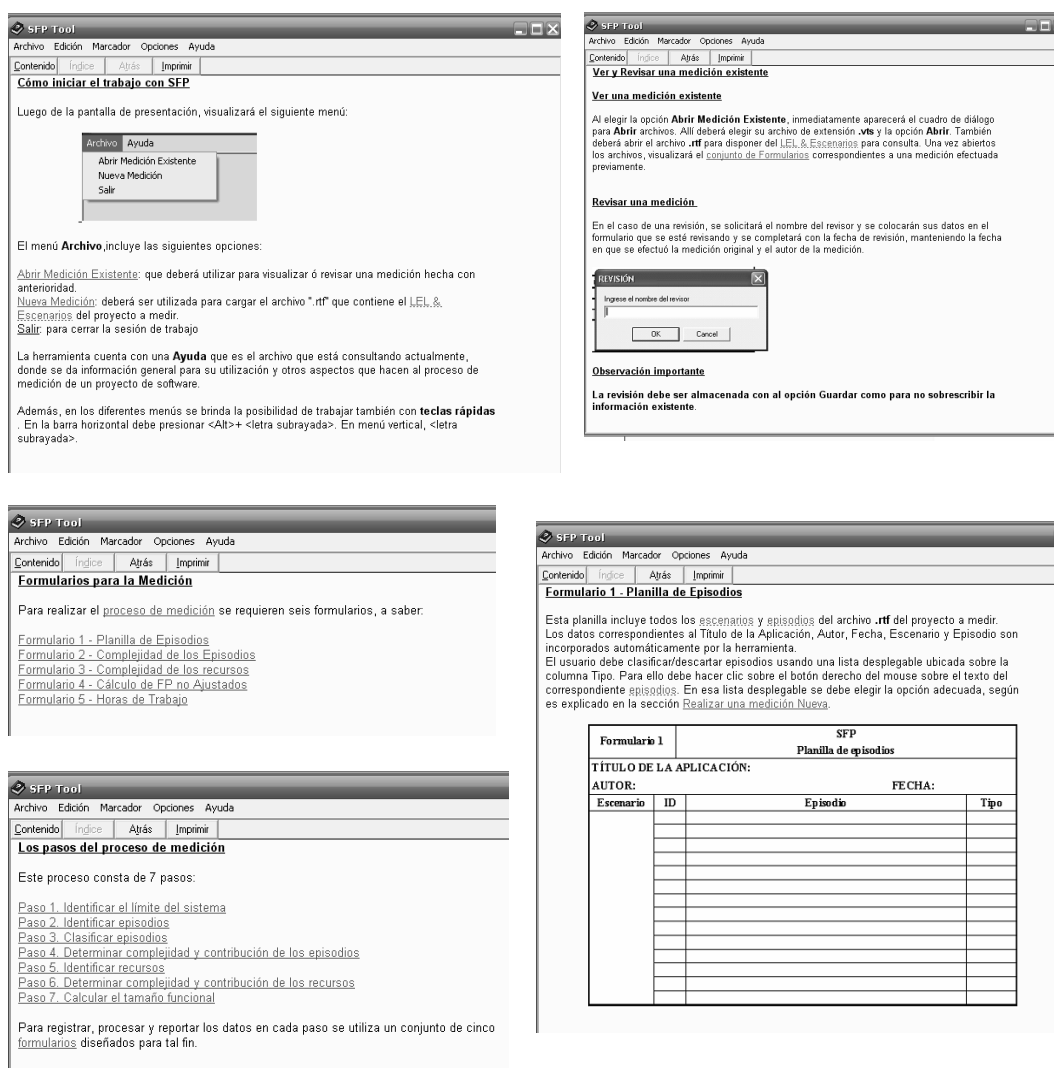


Figura E.15. Ventanas de ayuda

### E.2.7. Mediciones

El software fue usado para medir los escenarios de los sistemas *Notificaciones a Clientes de un Banco* y *Recepción del Hotel*. En ambos casos los resultados se observaron diferencias con la medición manual (Tabla E.1).

Tabla E.1. Comparación de la medición manual y automatizada

Sistema	Medición manual (FP)	Medición con SFP (FP)
Recepción del Hotel	103 (9 recursos)	82 (6 recursos)
Notificaciones a Clientes de un Banco	279 (25 recursos)	269 (23 recursos)

En particular, las mediciones del *Sistema Notificaciones a Clientes de un Banco* fueron realizadas por diferentes integrantes del equipo de investigación. Esto en parte podría explicar las diferencias en los resultados, como puede ser la distinta interpretación de las reglas y el grado de entrenamiento en el uso del procedimiento SFP. Independientemente de los aspectos mencionados, en ambos casos el software identificó un menor número de recursos. Esto se debió a que los recursos usados en los episodios no estaban especificados en la sección recursos del escenario o se habían cometido errores al

escribir el nombre de los recursos) particularmente en aquéllos que no son símbolos del L&E y por lo tanto el error no es detectado por el software BMW.

### **E.3. Conclusiones**

SFP Tool permite implementar de manera semiautomática el procedimiento SPF para estimar los Puntos Función de un sistema de software a partir de los escenarios generados durante la Elicitación de Requerimientos. Entre las ventajas de la herramienta pueden señalarse: genera los formularios y realiza los cálculos automáticamente, facilita el acceso al L&E y reglas, permite exportar los formularios a la planilla de cálculo *Microsoft® Excel*. De este modo se reduce el esfuerzo del proceso, se evitan los errores de cálculo y se asegura la consistencia de los datos debido a que los mismos son extraídos exclusivamente desde el L&E. Debido a que se conservan todos los formularios, la herramienta permite el seguimiento entre los FP y las entidades que los originaron.

El software fue usado para medir el tamaño funcional de los escenarios de sistemas que habían sido medidos manualmente. En algunos casos, las diferencias observadas frente a los resultados de la estimación manual son debidas a que los escenarios usados no fueron inspeccionados previamente para verificar que los recursos usados en los episodios estuvieran especificados en la sección recursos del escenario o que se habían cometido errores al escribir el nombre de los recursos. De este modo la herramienta no identificó esos recursos y no los incluyó en el respectivo formulario. Esto significa que para obtener resultados correctos es conveniente inspeccionar los escenarios, tal como lo proponen Doorn *et al.* (1998).