

Evaluación Empírica de la Estimación de Tamaño Funcional de Escenarios

Mabel Bertolami

Departamento de Informática, Facultad de Ingeniería, UNPSJB, Argentina
mbertolami@gmx.net

Alejandro Oliveros

Magíster de Ingeniería de Software, Facultad de Informática, UNLP, Argentina
aoliveros@gmail.com

Resumen

El uso creciente de las métricas de Puntos Función para la gestión de proyectos de software ha promovido el desarrollo de variantes del Análisis de Puntos Función para aproximar el tamaño funcional cuando la documentación, el tiempo o los recursos disponibles no permiten realizar el cálculo estándar.

Scenario Function Points (SFP) es un procedimiento basado en el método IFPUG que permite estimar el tamaño funcional de los escenarios. En este artículo se presenta una validación empírica, sobre un sistema software ya implementado, y basada en la comparación entre el tamaño de los escenarios estimado por SFP y la medición del sistema con el método IFPUG. Esta última medición fue evaluada mediante el proceso de validación propuesto por Morris y Desharnais.

Los resultados obtenidos son alentadores en cuanto a la capacidad de estimación de SFP y confirman la posibilidad y necesidad del plan de ampliar la experimentación sobre un mayor número de casos.

1. Introducción

Las técnicas de Puntos Función (FP) permiten la medición temprana y eficiente del tamaño de un sistema software, pero su aplicación requiere disponer de la descripción detallada de los requerimientos funcionales del usuario. Hay dos situaciones en que resulta decisivo contar con un método de estimación compatible pero alternativo al estándar de FP. Uno de estos casos ocurre en las fases más tempranas de un proyecto de desarrollo o mejora del software cuando es imposible realizar el conteo de FP de acuerdo a las reglas del estándar IFPUG [16]. Los elementos requeridos por este cálculo no siempre son identificables al comienzo de un proyecto, cuando la necesidad de predicciones de esfuerzo, costo y tiempo basadas en la evaluación del tamaño es mucho más fuerte que al final de la fase de especificación funcional. El otro caso se presenta cuando se necesita una evaluación de un software existente pero la documentación o el tiempo y los recursos requeridos para el cálculo detallado de FP no

están disponibles. Para esas y otras situaciones análogas ha surgido la demanda de métodos para estimar – no contar – FP desde las organizaciones involucradas en el negocio del software [23]. En ese caso es imprescindible reestimar el proyecto tan pronto se cuente con más información y en diferentes fases del ciclo de desarrollo.

En trabajos previos hemos presentado dos propuestas – SMKII¹ [3] y SFP [5] – para estimar el tamaño funcional de los escenarios basados en el enfoque de Leite [21]. Estos escenarios son derivados desde el Léxico Extendido del Lenguaje (LEL) mediante heurísticas específicas [20].

El procedimiento SFP está basado en la asociación conceptual entre las funciones transaccionales y de datos de IFPUG y los episodios y los recursos de los escenarios, respectivamente. Incluye un conjunto de reglas para identificar, clasificar y cuantificar las entidades del modelo y una función para determinar los FP del sistema. La ejecución del procedimiento SFP es soportada por un proceso que, partiendo de los escenarios y a través de la aplicación de las respectivas reglas en sucesivos pasos, concluye con una estimación del tamaño funcional de los escenarios.

Como parte de un plan de validación del procedimiento SFP en este trabajo se propone aplicarlo a los escenarios generados a partir de una aplicación implementada y comparar la estimación resultante con la medición de FPs de la aplicación producida por el método estándar IFPUG [16]. Para validar el resultado de la medición de FPs de la aplicación fue aplicada la propuesta de validación de Morris y Desharnais [27].

Los escenarios, por su propia génesis, son independientes de la metodología de desarrollo o lenguaje propuestos para la implementación de una aplicación software. La estimación de tamaño funcional basada en escenarios podría representar una alternativa para el conteo de los FP de una aplicación instalada (o *baseline* de FP [16]) que podrían usarse cuando se estudia la factibilidad de un proyecto de mejoras² o se evalúa el

¹ Adaptación del método MKII [25] para escenarios, esta denominación fue usada para facilitar la escritura.

² El término “mejora” se refiere a cambios en la funcionalidad de un sistema de información [28].

reemplazo de un sistema y no se dispone de la documentación.

El resto de este artículo está organizado de la siguiente manera: en la sección 2 se resumen los trabajos relacionados. En la sección 3 se describe la estrategia de evaluación aplicada. En la sección 4 se presenta el caso de estudio. En la sección 5 se describe el enfoque de estimación y su aplicación a los escenarios. La sección 6 incluye un resumen del método IFPUG, la medición de la aplicación y su validación. En la sección 7 se analizan los resultados y se describen las limitaciones del estudio. En la sección 8 se presentan las conclusiones y trabajos futuros.

2. Trabajos relacionados

Entre los trabajos relacionados se han seleccionado aquéllos que presentan estudios comparativos entre variantes del Análisis de Puntos Función (FPA) para la estimación temprana y el método IFPUG, independientemente del dominio al cual están dirigidos.

Meli [24] presenta una comparación de las mediciones de 15 aplicaciones usando el estándar IFPUG 4.0 con las estimaciones basadas en Early Function Points (EFP). Esta técnica permite una estimación más rápida y temprana de IFPUG FP en base a la identificación de objetos de software (datos y funcionalidad) con diferentes niveles de detalle. En su artículo demuestra que existe una fuerte correlación entre ambas medidas.

Desharnais y Abran [10] evaluaron la exactitud de la técnica Function Points Simplified (FPS) comparando su resultado con el valor real obtenido con IFPUG. El estudio sobre 90 proyectos determinó que existe una alta correlación entre las mediciones y la diferencia promedio entre las medidas es menor que el 5%. FPS usa un único factor para cada tipo de función en lugar de los 3 propuestos por el FPA. El método permite estimaciones rápidas y es adecuado cuando la documentación está incompleta.

En el campo de la estimación de aplicaciones Web, Cândido [8] propone una simplificación de NESMA Early FPA [15] que asigna complejidad baja a todas las funciones. Un estudio empírico basado en 20 aplicaciones de una compañía demostró que los resultados obtenidos son muy aproximados a los producidos por el método IFPUG.

En el marco de la estimación del tamaño funcional de proyectos de software orientado a objetos, Abrahão et al. [1] presentan un experimento basado en la medición con IFPUG y OOmFP sobre los documentos de diseño producidos desde una única especificación de requerimientos. Las evidencias empíricas demostraron que en ese contexto OOmFP produce resultados más consistentes y precisos.

3. Descripción del Proceso de evaluación

El FPA estándar mide el tamaño funcional a partir de la especificación de requerimientos. La exactitud de la medición puede ser testada frente a la medición del producto software terminado.

SFP permite estimar el tamaño funcional de un sistema a partir de los escenarios, sin embargo, ¿es consistente la estimación de los FPs de los escenarios de la aplicación con la medición de la aplicación basada en el método IFPUG FPA?, ¿es más simple estimar los FP de los escenarios que medir la aplicación con IFPUG?

El procedimiento de validación elegido, como parte de un programa mayor de validación del procedimiento SFP, consistió en construir los escenarios de una aplicación existente y con éstos estimar los FP y compararlos con los FP resultantes de aplicar el método IFPUG a la aplicación terminada.

Se realizaron las siguientes actividades (resumidas en el esquema de la Figura 1):

1. Construir el LEL y los Escenarios (L&E): se elaboró una estrategia para describir el L&E de un sistema implementado y se aplicó a un sistema real.
2. Estimar el tamaño funcional con SFP: se aplicó el procedimiento SFP a los escenarios generados en el paso previo.
3. Medir la aplicación software con el método IFPUG: se midió el tamaño funcional de la aplicación implementada de acuerdo a las reglas del método estándar IFPUG.
4. Validar la medición: se evaluó el resultado obtenido en el paso previo considerando las pautas sugeridas por Morris y Desharnais [27] para validar una medición de FPs.
5. Analizar los resultados y obtener conclusiones sobre la capacidad de estimación de SFP: se comparó la estimación producida por SFP con la medición basada en IFPUG y con otras propuestas de estimación. Este análisis permitió evaluar la exactitud de la estimación y su factibilidad de aplicación como herramienta de estimación.

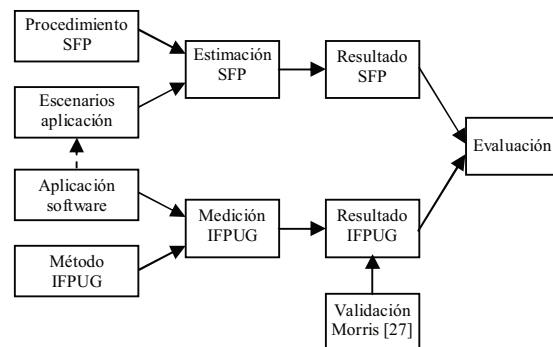


Figura 1. Estrategia de evaluación de SFP

4. Caso de estudio

Debido al fin exploratorio del trabajo se consideró conveniente elegir un sistema software relativamente pequeño. A partir de la información disponible sobre el sistema se desarrolló el L&E, condición para aplicar el procedimiento SFP.

4.1. Descripción del Sistema

El Sistema de Presentación de Programas (SPP) es utilizado³ por los profesores de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco (UNPSJB) para generar el programa de cada asignatura. El SPP forma parte de un sistema mayor que incluye otras dos aplicaciones software – SPPAdmin y SPPWeb – que son usadas exclusivamente por el administrador. El sistema general consta de dos bases de datos – SPP y Programas de las asignaturas – 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 trabajo se circunscribe a la aplicación SPP usada por los profesores. El software presenta varias secciones: General, Contenidos Mínimos, Programa Analítico, Cronograma Actividades, Bibliografía, Metodologías Enseñanza y Formas de Evaluación.

4.2. Construcción del L&E

Este proceso fue realizado por el primer autor y Elena Centeno, ambos se desempeñan como profesores y son usuarios del SPP. Luego de recolectar la información

sobre el sistema se elaboró la versión inicial del LEL y a través de sucesivos intercambios y revisiones se produjo la versión final. De manera similar se generaron los escenarios.

Se carece de un proceso definido para construir los escenarios de un sistema implementado y carente de documentación, por lo tanto se definió la siguiente estrategia para construirlo.

1. **Establecer el Universo de Discurso (UD)**⁴. A partir del análisis del contexto en que se desarrollaría el proceso fueron identificados los siguientes recursos y fuentes de información disponibles: el programador, el administrador del sistema, los usuarios, el software bajo estudio, el manual del usuario y el código fuente.
2. **Conocer las características del software.** Para ello se realizaron las siguientes actividades:
 - *entrevistar al desarrollador y al administrador del software.* Estas entrevistas permitieron entender el funcionamiento del sistema en general y de la aplicación SPP en particular. A medida que surgieron dudas se hicieron nuevas consultas.
 - *revisar el código fuente del software.* En nuestro caso, esta actividad aportó información sobre la organización de los datos que usa la aplicación. 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 las diferentes secciones del SPP, incluyendo las ventanas de ayuda y el manual del usuario. Estos últimos permitieron aclarar aspectos específicos acerca del uso y configuración del software, el llenado de los formularios, el modo de almacenamiento de la información ingresada y la generación de los productos requeridos por la Facultad. En esta etapa 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 que incluye 41 entradas del LEL. Luego los símbolos fueron clasificados. La descripción de la noción e impacto requirió iteraciones de los pasos 2 y 3 para completar o verificar dichas descripciones.
5. **Generar los escenarios.** Se aplicó la heurística propuesta por Leite [20] para describir los escenarios a partir de los símbolos del LEL. Durante este paso fue necesario repetir las actividades del paso 2 para completar o verificar la descripción de los escenarios. Como resultado se generaron 18 escenarios⁵.

³ En el momento en que se comenzó esta investigación; recientemente fue reemplazado por una aplicación de características y funcionalidad similares, pero en otro ambiente.

⁴ 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 [20].

⁵ El L&E puede ser solicitado al primer autor.

5. Estimar el tamaño funcional

El tamaño funcional de los escenarios fue estimado con el procedimiento SFP [5]. En esta sección se presentan los conceptos que soportan el enfoque y los resultados de su aplicación al sistema SPP.

El procedimiento SFP permite estimar el tamaño en FP de un sistema software a partir de los escenarios. Su diseño está basado en el Método IFPUG FPA [16] y en la estructura propuesta en el estándar ISO/IEC 14143-1 [17] para los métodos de medición del tamaño funcional.

La definición de SFP establece asociaciones conceptuales entre los componentes lógicos de los escenarios y los propuestos por IFPUG. El modelo de SFP consta de tres tipos de BFC (Componente Funcional Básico): *recurso*, *Entrada Externa* (EI) y *Salida Externa* (EO). Un recurso representa una entidad de soporte de información (sin distinguir entre datos “internos” y “externos” a la aplicación). Una EI tiene semántica similar pero no es equivalente a la EI de IFPUG, pues estas últimas mantienen un ILF mientras que las EIs en SFP mantienen un recurso. La EO tiene el propósito de presentar datos independientemente de la naturaleza de los mismos (éstos pueden ser recuperados desde un recurso o generados por el proceso) y es la combinación de EO y EQ de IFPUG. SFP incluye *reglas de identificación y clasificación* de los tipos de entidades, *reglas de asignación numérica* para valorar las entidades del modelo y una función para derivar el tamaño funcional desde los componentes individuales. La complejidad y contribución en FP de EIs, EOs y recursos es determinada de modo similar a IFPUG usando las tablas para EI, EQ e ILF respectivamente.

El proceso definido para el SFP consta de los siguientes pasos:

1. *Identificar el límite del sistema*: se construye una lista con todos los episodios de todos los escenarios.
2. *Identificar episodios*: se aplican las reglas de identificación de episodios para determinar el conjunto definitivo de episodios. En este paso se puede producir el descarte de algunos episodios.
3. *Clasificar episodios*: se aplican las reglas para clasificar los episodios como EI y EO.
4. *Determinar complejidad y contribución de las EIs y EOs*: comprende la cuenta de Tipos de Datos Elementales (DETs) y Tipos de Recursos Referenciados (RTRs) en base a las reglas definidas para EI y EO y su posterior uso como valores de entrada en las Tablas de IFPUG para EI y EQ.
5. *Identificar recursos*: cada episodio es examinado y sus recursos son identificados usando las reglas para identificar recursos.
6. *Determinar complejidad y contribución de los recursos*: comprende la cuenta de DETs y Tipos de Registros Elementales (RETs) en base a las reglas definidas para los recursos. El nivel de detalle del LEL

no permite identificar diferentes RETs en un recurso, por lo tanto se considera que todos tienen 1 (un) RET. Ambos valores (#DETs y 1) son usados como entrada a la Tabla para ILF de IFPUG.

Los pasos 3-4 y 5-6 pueden realizarse en paralelo.

7. *Calcular el tamaño funcional*: aplicando la siguiente fórmula se determinan los FPs de los escenarios.

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

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

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 el Apéndice A se presentan los motivos para descartar un episodio candidato y la cantidad de episodios eliminados en cada caso.

En la Tabla 1 se resumen los resultados del proceso de estimación aplicado a los escenarios del sistema SPP. La misma incluye el esfuerzo requerido por el proceso de estimación. En el Apéndice B se incluye un ejemplo sobre la valoración de la complejidad y contribución en FPs de una EI y un recurso.

Tabla 1. Resumen de los resultados del proceso SFP

EI			FP	EO			FP	Recurso		FP	Total FP	Esfuerzo (hs)
Compl. ¹				Compl.				Compl.				
B	M	A		B	M	A	B	M				
28	0	0	84	15	0	0	45	14	0	98	227	04:40

¹ Compl.: complejidad; B: baja, M: media, A: alta

Cabe mencionar que también fue aplicado el procedimiento SMKII [3] a estos escenarios obteniéndose 257 FP, que es un 13% mayor que el valor de SFP. Este resultado es compatible con los obtenidos sobre otros casos [3] y es debido a las diferencias conceptuales entre los enfoques.

6. Medir el tamaño funcional

La aplicación SPP fue medida con el método IFPUG [16]. La relevancia del resultado para el propósito de este estudio y la falta de mediciones propias sobre proyectos similares motivó la evaluación de la consistencia interna de la cuenta de FP mediante la propuesta de validación de Morris y Desharnais [27].

6.1. Medición de la aplicación SPP con el Método IFPUG FPA

La medición fue realizada de acuerdo al procedimiento propuesto en el manual para la Medición de Puntos Función [16]. Como la aplicación existe y está en

funcionamiento, el *tipo de la cuenta* es “Medición de FP de una aplicación”. El *alcance* de esta cuenta comprende toda la funcionalidad suministrada al usuario por el software. El *propósito de la cuenta* es comparar el resultado con la estimación basada en los escenarios. Para el propósito de este estudio SPP es considerada una aplicación software independiente, por lo tanto el límite está representado por la línea que la separa de otras aplicaciones, como se indica en la Figura 2.

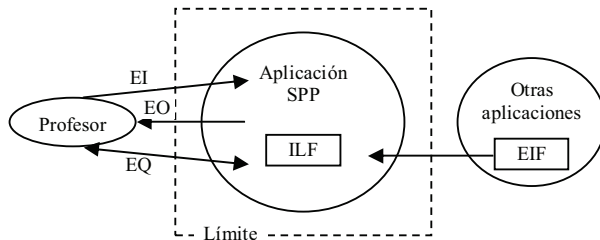


Figura 2. La aplicación desde el punto de vista del usuario [16]

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 se ha realizado en base al manual del usuario y la aplicación operacional existente [9].

El código fuente aportó valiosa información para conocer 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 (como 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.

Un ejemplo de ILF es *Programa analítico* que contiene 3 DETs (*Número, Título, Contenido*) y 1 RET. Este ILF tiene complejidad *baja* y contribuye con 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).

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 (DETs y FTRs) de las EIs, EQs y EOs fue determinada utilizando una estrategia similar.

El proceso para *Agregar nueva unidad al Programa analítico* (Figura 3) es un ejemplo de 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 [16]). El ILF mantenido es *Programa analítico*, por lo tanto hay 1 FTR. Esta EI es de complejidad *baja* y su contribución es de 3 FPs.

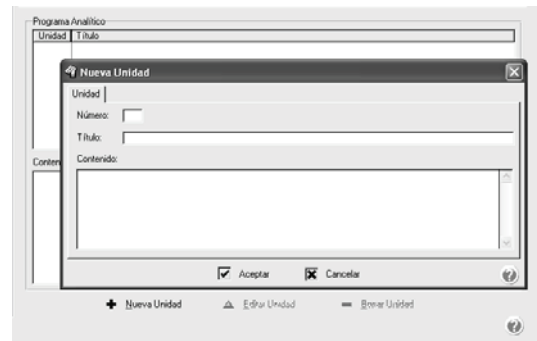


Figura 3. Pantalla “Nueva Unidad” del Programa Analítico

La Tabla 2 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 2. Resumen de la cuenta de FP

Función	Compl. baja		Compl. media		Compl. 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%
Transacciones	31	93	2	9	3	20	-	122	70%
ILF	6	42	0	0	0	0	7	42	24%
EIF	2	10	0	0	0	0	5	10	6%
Archivos	8	52	0	0	0	0	-	52	30%
								174	100%

Cabe mencionar que también fue medida la aplicación SPP con el método MKII [25] obteniéndose 182 FP, que sólo difiere en un 5% con la medición IFPUG. El resultado también fue corroborado aplicando la fórmula de conversión - válida hasta aproximadamente 1500 IFPUG FPs - propuesta por Symons [30]:

$I = 1000 * (\text{SQRT}(0.81 + 0.002 * M) - 0.9)$, siendo $M = \text{MKII FP}$ e $I = \text{IFPUG FP}$.

$I = 1000 * (\text{SQRT}(0.81 + 0.002 * 182) - 0.9) \therefore I = 184$ FP, lo que representa una diferencia del 6% respecto a la medición real.

6.2. Validación de la medición

En esta sección se presenta un resumen de las propiedades de la cuenta de FP y/o atributos del software evaluados en los tres niveles del proceso de validación. En el proceso de validación de la cuenta de FP se siguió el método de chequeo propuesto en [27].

Revisión de alto nivel. El tamaño funcional fue comparado con la estimación producida mediante “backfiring” [19]. Esta técnica provee una tabla de conversión entre el número de líneas de código fuente (SLOC) y los FP para diferentes lenguajes de programación [14]. Aunque es cuestionada por la potencial inconsistencia de sus resultados [10], [27], lo cierto es que es adoptada por muchas herramientas de estimación [6], [19]. Conociendo el número de SLOC (8022⁶) 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 [18]; 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).

La verificación de la correcta identificación de las EIs (todas actualizan ILFs) y de la contribución en FPs de los archivos externos con respecto al total de archivos (19% frente al tope sugerido del 50%) permitieron comprobar la correcta ubicación del límite del sistema.

Revisión de nivel intermedio. La relación entre el tamaño funcional y el número de archivos lógicos fue comparada con las establecidas a partir del análisis estadístico de proyectos [27]. 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.

$$FP = 30.813 * \#ILF \therefore FP = 185 (1)$$

$$FP = 22.094 * (\#ILF + \#EIF) \therefore FP = 177 (2)$$

La proporción entre el número de funciones de entrada y salida y el número de archivos (Tabla 2) fue evaluada con respecto a los valores estadísticos de otras organizaciones (Tabla 3).

Tabla 3. Relación entre transacciones y archivos

Relación	EI			EO			EQ		
	Act. ¹	Esp. ²	%	Act.	Esp.	%	Act.	Esp.	%
ILF	3.33	2.7	23%	0.50	1.2	-58%	2.17	1.2	81%
ILF+EIF	2.50	1.9	32%	0.38	0.9	-58%	1.63	0.8	103%

¹ Act: actual; ² Esp.: esperado

⁶ Líneas de código físicas, excluyendo líneas en blanco y comentarios [29], contadas manual y automáticamente [11].

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.

Otro aspecto examinado fue el porcentaje de contribución a la cuenta de FP por cada tipo de función. Los porcentajes típicos para aplicaciones MIS (Management Information Systems) 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 [27]. Para la medición actual (Tabla 2) 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 está dentro del 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%.

Por último, fue evaluada la complejidad de las funciones. La complejidad promedio [27] de las transacciones y archivos se corresponde con la complejidad media y baja de las tablas de IFPUG [16], respectivamente. Para la medición actual (Tabla 2) 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. Los resultados de la revisión de los niveles previos sugieren que la cuenta es potencialmente válida. El análisis detallado de las funciones transaccionales y de datos ratifica lo expresado en el nivel intermedio, lo que permite suponer que la cuenta es válida.

7. Evaluación

Esta sección incluye el análisis de los resultados obtenidos, la comparación con otras propuestas y las limitaciones de este estudio.

7.1. Análisis de los resultados

El análisis de los resultados obtenidos (Tabla 4) tiene por objetivo hacer una evaluación acerca de la capacidad

de SFP como herramienta para estimar los FPs a partir de los escenarios.

Tabla 4. Resumen de la estimación/medición de tamaño funcional del sistema SPP

Estimación SFP	Medición IFPUG
227	174

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 con SFP es del 30%. El error relativo es calculado como $(IFPUG\ FP - SFP\ FP) / IFPUG\ FP$ y se usa para evaluar la exactitud de la estimación. Esto plantea un nuevo interrogante, ¿es aceptable este error? Una posible respuesta se puede obtener observando la variación del rango de las estimaciones de tamaño (Figura 4) a medida que progresa el ciclo de vida del software [7]. En las etapas más tempranas (antes de completar la especificación de requerimientos) el rango es mayor que $1.5x - 0.5x$ y llega hasta $4x - 0.25x$ en la fase de factibilidad. En este contexto, el error de nuestra estimación puede considerarse admisible. Las amplitudes mencionadas en la Figura 4 no deben sólo considerarse como extremos, sino como la mejor precisión 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" [22]. Esto requiere completar las estimaciones con un "grado de incertidumbre", que en este caso no hemos estimado pero a priori puede considerarse elevado.

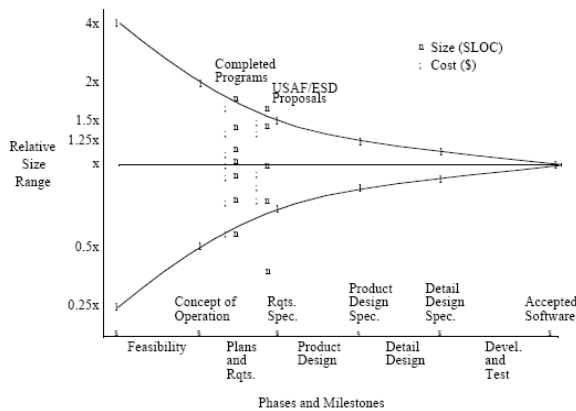


Figura 4. Exactitud de la estimación de costo y tamaño del software vs. fase [7]

- Morris [26] define un número de niveles para el conteo de FP, donde el nivel puede estar restringido, por ejemplo, por la calidad de la documentación o el tiempo disponible. Para los niveles 5 (Rough Count) y 6 (Size Approximation) el grado de exactitud es de $\pm 20 - 25\%$.
- Comparado con otras propuestas de estimación: 1) Rule of the "Thirties" [12] 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 nuestro caso, a partir de las 8 entidades identificadas y el valor 31, el cálculo produciría 248 FP. Este valor supera en un 43% al medido con IFPUG y en un 9% al estimado con SFP; 2) ISBSG Benchmark [13], Tichenor [23], Early Function Points Method [2], NESMA Early FPA [15] requieren un mayor grado de conocimiento del sistema que permita identificar algunas (ILFs y/o EIFs) o todas las funciones de IFPUG.

7.2. Limitaciones del estudio

Este trabajo, como fue explicitado, representa una primera experiencia hacia la validación de SFP y está basado en una sola aplicación de tamaño relativamente pequeño. Se prevé ampliar la experimentación sobre otros casos de tamaños variados.

La medición IFPUG no fue realizada por un especialista. Este aspecto no invalida nuestra propuesta pues se apunta a tener una aproximación del tamaño del sistema que sirva para evaluar la estimación de SFP, en este caso orientada a la etapa de factibilidad, para la cual 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.

8. Conclusiones y Trabajos futuros

En este trabajo presentamos una evaluación experimental de la estimación de tamaño funcional basada en escenarios. La estrategia elegida consistió en comparar esa estimación con la medición obtenida con el método IFPUG. El estudio se basó en desarrollar ambos procesos con un sistema software implementado del que se carecía de documentación.

La realización del trabajo exigió resolver dos cuestiones preliminares. En primer lugar, debido a que los escenarios son usados como entrada para el proceso SFP fue necesario elaborar un proceso para construirlos a partir de la aplicación. En segundo lugar, la relevancia de la medición de acuerdo al proceso de IFPUG para el objetivo de este estudio determinó la necesidad de investigar la validez de esta cuenta mediante un proceso que, según sus autores, ha demostrado ser exitoso para detectar cuentas incorrectas.

Luego de aplicar 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 el valor estimado por SFP es suficientemente aproximado al obtenido con IFPUG. En comparación con otras propuestas, el resultado es similar y requiere un menor esfuerzo, aunque esto deberá ser corroborado a través de más experimentación.

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 FP que proviene de la aplicación en funcionamiento.

El trabajo experimental en curso tiene como meta demostrar la validez de las estimaciones de SFP, éste podría representar una alternativa para facilitar la estimación de proyectos de mejora o reemplazo de sistemas cuya documentación no está disponible. En esos casos la medición con el estándar IFPUG requiere un esfuerzo significativo que podría ser aliviado a partir de la medición de los escenarios, los cuales podrían ser contruidos aplicando la estrategia descripta. También contribuirá a establecer estimaciones de tamaño en las etapas más iniciales del proceso de construcción.

Para ratificar estos resultados se planea extender la validación empírica de las estimaciones a un mayor número de casos, actividad que será prioritaria en los trabajos futuros. Otros temas de interés son comparar la estimación en diferentes momentos del ciclo de vida del desarrollo y agilizar la estimación de FP usando un modelo de estimación basado en los datos disponibles para reducir el esfuerzo requerido por el proceso.

Apéndice A: Razones para eliminar episodios

Un *episodio simple* es una sentencia simple que representa una acción atómica que se ejecuta en forma completa e independiente de cualquier otra acción dejando la aplicación en estado consistente. Las reglas de identificación de episodios establecen que se descarta un episodio candidato que no cumple con la definición de episodio simple, un episodio que es un escenario, una sentencia que especifica una excepción o una restricción y un episodio que no referencia un recurso (mantener o leer).

Además, una de las reglas para identificar EI y EO establece que el proceso lógico es único entre los procesos lógicos realizados por otras EIs y EOs.

Aplicando dichas reglas se descartaron 69 episodios según el detalle indicado en la Tabla 5.

Tabla 5. Detalle de episodios descartados

Motivo	Cantidad
no es proceso elemental	10
episodio escenario	15
especifica excepción	8
especifica restricción	4
no referencia un recurso	20
proceso lógico no es único	12

Apéndice B: Ejemplos de aplicación de SFP

1. Determinar la contribución de una EI.

La siguiente EI recibe datos desde fuera del límite del sistema y actualiza el recurso *programa analítico*: Si el *profesor* selecciona la opción Nuevo entonces ingresa el número, título y contenido de una nueva *unidad* en el formulario del *programa analítico*.

3 DETs: *número, título, contenido*. Son determinados aplicando las Reglas para Identificar y Contar DET en EIs y EOs.

1 RTR: *programa analítico*. Es determinado aplicando las Reglas para contar RTR en EI.

Complejidad y contribución en FPs: Baja (3 FPs). Son determinadas usando el #DET y #RTR como entrada a la tabla para EI de IFPUG.

2. Determinar la contribución de un recurso.

El recurso *programa analítico* es identificado aplicando las Reglas para identificar recursos.

3 DETs: *número, título y contenido*. Son determinados aplicando las Reglas para Identificar y Contar DETs en los recursos.

1 RET. Es establecido por la Regla para contar RET.

Complejidad y contribución en FPs: Baja (7 FPs). Es determinado usando el #DET y #RET como entrada a la tabla para ILF de IFPUG.

Agradecimientos

Los autores agradecen a la Facultad de Ingeniería de la UNPSJB por permitir el uso del software SPP para esta investigación.

Referencias

- [1] Abrahao, S., Poels, G., Pastor, O., "Assessing the Reproducibility and Accuracy of Functional Size Measurement Methods through Experimentation", *Proc. ISESE 04*, 2004.

- [2] Asensio, R., Sanchis, F., Torre, F., García, V., Uría, G., "A Preliminary Study for the development of an Early Method for the Measurement in Function Points of a Software Product", <http://arxiv.org/pdf/cs.SE/0402015>, consultado en junio de 2005.
- [3] Bertolami, M., Oliveros, A., "Análisis de Puntos Función en la elicitación de requerimientos", *Proc. 6th Workshop on Requirements Engineering*, Piracicaba, Brasil, 2003, pp. 32-47.
- [4] Bertolami, M., Oliveros, A., "Proceso de medición de funcionalidad en la Elicitación de Requerimientos", *Proc. 7º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software, IDEAS2004*, Arequipa, Perú, 2004, pp. 91-102.
- [5] Bertolami, M., Oliveros, A., "SFP: Un Procedimiento de Estimación de Puntos Función de Escenarios", *Proc. 9th Workshop on Requirements Engineering*, Rio de Janeiro, Brasil, 2006, pp. 101-108.
- [6] Boehm, B. et al., *COCOMO II Model Definition Manual*, University of Southern California, Los Angeles, Calif., 1998.
- [7] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R., "Cost models for future software life cycle processes: COCOMO 2.0". *Annals of Software Engineering*, Vol. 1, 1995, pp. 57-94.
- [8] Cândido, E., Sanches, R., "Estimating the size of web applications by using a simplified function point method", *Proc. WebMedia and LA-Web*, pp. 98-105, 2004.
- [9] Dekkers, T., "(Extended) Functional size measurement methods are also applicable in enhancement projects", *IWSM2003: 13th International Workshop on Software Measurement*, Montréal, Canada, 2003.
- [10] Desharnais, J., Abran, A., "Approximation techniques for measuring function points", *Proc. of the 13th International Workshop on Software Measurement (IWSM 2003)*, Montréal (Canada), pp. 270-286, Sep. 2003.
- [11] GeroneSoft Code Counter Pro Software, <http://www.geronesoft.com/>, v1.24, Apr 2005.
- [12] [http://www.isbsg.org/isbsg.nsf/weben/Estimating Functional Size.htm](http://www.isbsg.org/isbsg.nsf/weben/Estimating_Functional_Size.htm), consultado en mayo de 2005.
- [13] [http://www.totalmetrics.com/cms/servlet/ISBSG - metrics, products & services for software development.htm](http://www.totalmetrics.com/cms/servlet/ISBSG_products_services_for_software_development.htm), consultado en mayo de 2005.
- [14] [http://www.mayesoconsulting.com/IndustryData2.htm# Industry Data](http://www.mayesoconsulting.com/IndustryData2.htm#IndustryData), Apr. 2005.
- [15] <http://www.nesma.nl/english/earlyfpa.htm>, consultado en marzo de 2005.
- [16] IFPUG, *Manual para la Medición de Puntos Función*, Versión 4.1.1., AEMES, 2000.
- [17] International ISO/IEC Standard 14143-1, "Information technology - Software measurement - Functional size, Part 1: Definition of concepts", 1998.
- [18] Jones, C., Programming Languages Table, Release 8.2, Software Productivity Research, Inc., Mar. 1996.
- [19] Jones, C., Strengths and weaknesses of software metrics, Software Productivity Research, Version 5, Mar 2006.
- [20] Leite, J., Hadad, G., Doorn, J., Kaplan, G., "A Scenario Construction Process", *Requirements Engineering Journal*, 2000.
- [21] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., "Enhancing Requirements Baseline with Scenarios", *Requirements Engineering Journal*, Vol. 2, Nro. 4, 1997, pp. 184-198.
- [22] McConnell, S., *Software estimation. Demystifying the Black Art*, Microsoft Press, 2006.
- [23] Meli, R. and Santillo, L., "Function Point Estimation Methods: a Comparative Overview", *FESMA 99*, Oct. 6-8, Amsterdam, 1999.
- [24] Meli, R., "Early and Extended Function Point: A New Method for Function Points Estimation", *IFPUG-Fall Conference*, Scottsdale, Arizona, USA, Sep. 1997.
- [25] *MKII Function Points Analysis Counting Practices Manual*, Version 1.3.1, UKSMA (United Kingdom Software Metrics Association), 1998.
- [26] Morris, P., "Levels of Function Point Counting", Version 1.3, *Total Metrics*, 2004.
- [27] Morris, P., Desharnais, J., "Function Point Analysis - Validating The Result", Version 1.4, *Total Metrics Pty Ltd and SELAM*, Jul. 2001.
- [28] NESMA, "Function Point Analysis for Software Enhancement", Professional guide of the Netherlands Software Metrics Users Association, Version 1.0, 2001.
- [29] Rozum, J., "Defining and Understanding Software Measurement Data", *Proc. of the 5th Annual Conference on Applications of Software Measurement*, Nov. 1994.
- [30] Symons, C., "Conversion between IFPUG 4.0 and MKII Function points", Software Measurement Services Ltd, 1999.