

Medición de Puntos Función sobre especificaciones en UML, una evaluación experimental

Mabel Bertolami

Departamento de Informática, Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco, Comodoro Rivadavia, R. Argentina
mbertolami@gmx.net

Alejandro Oliveros

Facultad de Informática, Universidad Nacional de La Plata, La Plata, R. Argentina
aoliveros@gmail.com

V Workshop de Ingeniería de Software y Bases de Datos

Resumen

Las técnicas de Puntos Función permiten estimar el tamaño del software en las etapas iniciales del desarrollo (desde los requerimientos funcionales del usuario). UML (Unified Modeling Language) es una notación ampliamente aceptada para la especificación y diseño de sistemas software. Diferentes enfoques han sido propuestos para medir el tamaño funcional desde dichos documentos. Tavares propone la medición de Puntos Función a partir de especificaciones de requerimientos expresadas en notación UML. El enfoque incluye un conjunto de reglas que establece la asociación entre los conceptos del estándar IFPUG y UML.

En este artículo presentamos la aplicación de la propuesta de Tavares sobre la especificación de un sistema real y la comparación con la medición IFPUG de la aplicación software implementada. Los resultados obtenidos confirman que la estimación temprana es muy aproximada a la medición de la aplicación terminada.

Palabras claves: tamaño funcional, especificación de requerimientos, UML.

1 INTRODUCCIÓN

El tamaño funcional es definido como el tamaño de un sistema software derivado de la cuantificación de los Requerimientos Funcionales del Usuario [7]. Está basado en el punto de vista del usuario y es independiente de la tecnología usada para el desarrollo e implementación. UML [8] es un lenguaje usado para la especificación y diseño de sistemas software. El interés creciente en aplicar métodos estándares para la medición de tamaño funcional a los requerimientos funcionales del usuario basados en modelos en UML ha resultado en variadas propuestas [10].

Tavares *et al.* [9] propone un enfoque basado en IFPUG [6] que a partir de un conjunto de reglas de asociación de conceptos permite el conteo de Puntos Función (FP) desde los Casos de Uso (UC) y diagramas de clases.

En este trabajo presentamos una evaluación empírica de la medición de un sistema software basada en la aplicación de dos diferentes técnicas. El tamaño funcional de la especificación en UML fue estimado usando el enfoque de Tavares y este resultado fue comparado con la medición de la aplicación mediante el método estándar IFPUG [6]. A diferencia de la experiencia de Tavares la medición IFPUG fue realizada sobre la aplicación implementada en lugar de usar la especificación

tradicional. Este proceso permitió comprobar que la medición de FP desde las especificaciones en UML es compatible con la medición estándar lo que corrobora la experiencia reportada por el autor [9].

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 el proceso de evaluación. En la sección 4 se describe el caso de estudio. En la sección 5 se presenta el enfoque de Tavares y su aplicación a los UC. En la sección 6 se describe la medición de la aplicación con el método IFPUG. En la sección 7 se presenta el análisis de los resultados y en la sección 8 las conclusiones.

2 TRABAJOS RELACIONADOS

Entre los trabajos relacionados se han seleccionado aquéllos que presentan estudios comparativos entre las estimaciones producidas por extensiones a IFPUG FPA para la estimación temprana basadas en documentos UML y las mediciones del método estándar IFPUG.

La propuesta de Fetcke [4] establece un mapeo del método OOSE (Object-Oriented Software Engineering) basado en UC de Jacobson con el modelo abstracto de FP. El objetivo es medir en la fase de análisis de requerimientos, cuando se han definido el modelo de UC y el modelo de objetos. Este mapeo fue formulado como un conjunto de reglas que soportan las reglas de IFPUG FPA. Los actores de los UC son los usuarios del sistema, los UC se asocian con las funciones transaccionales y las clases con los archivos lógicos (propone usar las clases del modelo de objetos del dominio o del modelo del análisis).

Cantone *et al.* [3] definen un modelo de convertibilidad de los elementos de UML en entidades del Análisis de Puntos Función (FPA) y una herramienta semiautomatizada para contar FPs. Este enfoque usa tres diagramas UML: diagrama de UC, diagrama de secuencia y diagrama de clases e incluye un conjunto de reglas que extiende el modelo de conversión propuesto por Fetcke *et al.* Un estudio comparativo demostró que mediante la herramienta se obtuvo un resultado similar al obtenido por un experto en IFPUG.

Harput *et al.* [5] proponen un modelo semiautomatizado que transforma un modelo de requerimientos orientado a objetos en un modelo tradicional del FPA. El enfoque se basa exclusivamente en los requerimientos, debido a que su objetivo es obtener una estimación temprana del tamaño del software. La transformación es aplicada sobre los UC, los diagramas de secuencia y de clase en base a un conjunto de reglas que asocian las clases a funciones de datos y los UC y requerimientos funcionales a funciones transaccionales. Una ventaja de usar el FPA es que los resultados obtenidos son comparables a los existentes del método FPA. Los autores aplican el enfoque al Caso de estudio 3 de IFPUG y a un caso del mundo real, en ambos se demuestra la compatibilidad con las mediciones del método estándar.

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.

El procedimiento de evaluación consistió en medir el tamaño funcional de una aplicación existente a partir de la especificación en UML y compararlo con los FP resultantes de aplicar el método IFPUG a la aplicación terminada. Se realizaron las siguientes actividades:

1. Medir el tamaño funcional a partir de la especificación del sistema en UML: fue aplicado el enfoque de Tavares sobre los UC y el diagrama de clases.
2. Medir la aplicación software: fue medido el tamaño funcional de la aplicación implementada con el método estándar IFPUG.

3. Analizar los resultados y obtener conclusiones: la medición basada en el enfoque de Tavares fue comparada con la medición IFPUG y con otras propuestas de estimación temprana. Este análisis permitió evaluar la exactitud de la estimación.

4 CASO DE ESTUDIO

Se trata de un sistema de gestión para un comercio¹ dedicado a la venta de indumentaria. La documentación disponible consiste en los Diagramas de Casos de Uso, descripción de los UC, diagramas de colaboración, diagrama de clases y el manual de usuario. Se dispone además de la aplicación software y la base de datos del sistema. La Figura 1 presenta el diagrama de contexto del sistema.

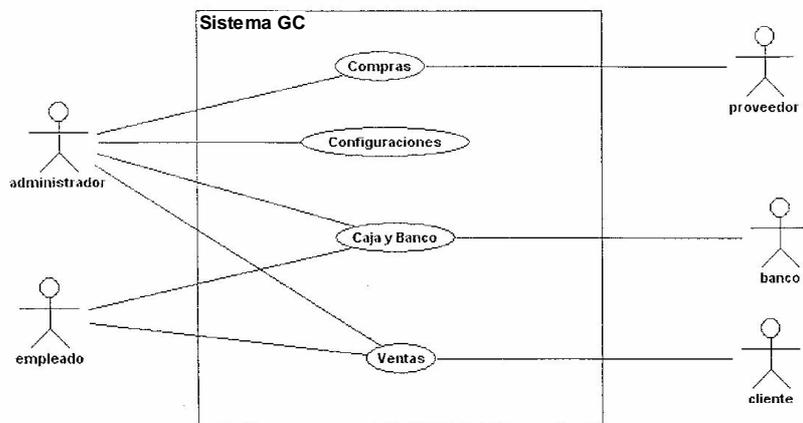


Figura 1. Diagrama de contexto del Sistema GC

Los principales requerimientos funcionales del sistema están representados en los UC:

UC Configuraciones
– Gestionar Usuarios
– Gestionar Configuraciones generales
UC Compras
– Gestionar Proveedores
– Gestionar Artículos
– Gestionar Compras
– Generar Listados
UC Ventas
– Gestionar Ventas
– Gestionar Clientes
– Gestionar Listados
UC Caja y bancos
– Vincular Medios Pago
– Gestionar Cheques
– Gestionar Conceptos
– Gestionar Tarjetas
– Gestionar Depósito

¹ Por razones de confidencialidad el nombre original ha sido sustituido por GC (Gestión Comercio).

- Gestionar Carteras
- Gestionar Bancos y Crédito

5 MEDICIÓN DE LA ESPECIFICACIÓN CON EL ENFOQUE DE TAVARES

La propuesta de Tavares *et al* [9], similar a la de Fetcke [4], establece una correspondencia entre los conceptos de IFPUG [6] y UML. El enfoque define un conjunto de reglas que permite identificar las funciones transaccionales a partir de los diagramas de UC y las funciones de datos desde el diagrama de clases. Esta propuesta fue aplicada a más de 120 proyectos que permitieron demostrar la exactitud de las estimaciones producidas en la fase de requerimientos con respecto a la medición realizada con el paradigma tradicional.

A continuación se presenta la aplicación del proceso y las reglas propuestas en [9] al caso de estudio.

En los UC el límite está representado por el borde conceptual entre los actores y el sistema. Considerando el diagrama de la Figura 1 y aplicando las reglas 1-2 [9] fueron identificados los siguientes usuarios: *Administrador, Empleado, Proveedor, Cliente y Banco.*

Para identificar las transacciones candidatas fueron evaluados los diagramas de UC de menor nivel correspondientes a cada UC de alto nivel (Configuraciones, Compras, Caja y Banco, Ventas), lo que dio como resultado 135 transacciones candidatas. Por ejemplo, aplicando las reglas 12-13 al UC Configuraciones (Figura 2) se obtuvo el listado de candidatos a transacciones de la Tabla 2.

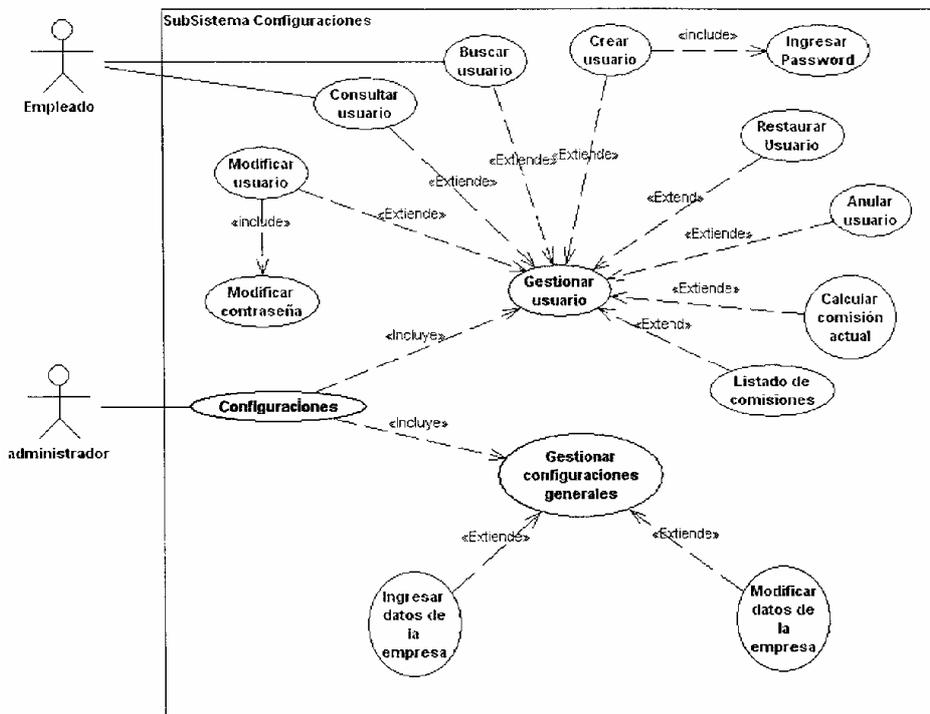


Figura 2. UC Configuraciones

Tabla 2. Transacciones candidatas del UC Configuraciones

UC Configuraciones
Actor: <i>Administrador</i>

1. UC Gestionar usuario
1.1. Crear usuario
1.1.1. Ingresar password
1.2. Modificar usuario
1.2.1. Modificar contraseña
1.3. Consultar usuario
1.4. Buscar usuario
1.5. Restaurar usuario
1.6. Anular usuario
1.7. Calcular comisión actual
1.8. Listado de comisiones
2. UC Gestionar configuraciones generales
2.1. Ingresar datos de la empresa
2.2. Modificar datos de la empresa
Actor: Empleado
1. Consultar usuario
2. Buscar usuario

Se aplicó el criterio de Cantone *et al.* [3] con respecto a la regla 13: los UC relacionados mediante las relaciones *extiende* e *incluye* también son candidatos para funciones transaccionales. La identificación de las clases candidatas a archivo lógico fue realizada aplicando las respectivas reglas [9] a las 35 clases que incluye el diagrama de clases (Tabla 3).

Tabla 3. Clases del sistema, relaciones entre clases y reglas aplicadas

Clase	Regla	Clase	Regla
<i>Entidades</i>	9 – RET ³	ArtNP	
Empresas (H) ¹		TStock (A)	
TDeCredito (H)		TArticulos (A)	
TBancos (H)		ArtRem	
<i>Personas</i>	9 – RET	TStock (A)	
Usuarios (H)		TArticulos (A)	
TPassword		<i>MediosPago</i>	9 - RET
TMails		TTarjetas (H)	
Teléfonos		TCarteras	
THistorialBajas		TDepositos (H)	
<i>Comprobantes</i>	9 – RET	TCheques (H)	
TFacturas (H)		TCheques3 (H)	10 - RET
TNotas (H)		HistorialMovCaja	
TNotaPedidos (H)		MovCaja	
TRemitos (H)		TConfiguraciones	
ArtFact		<i>Cuentas</i>	9 - RET
TStock (A) ²		CajaChica (H)	
TArticulos (A)		TCtaCtes (H)	
NCD		TCtaBancarias (H)	
TStock (A)		TArticulos	
TArticulos (A)		TRubros	
		TStock	

¹H: herencia; ²A: agregación, ³RET: Tipo de Registro Elemental

Según la regla 9 las 5 clases abstractas (indicadas en cursiva) son candidatas a RET. La clase TCheques3 se considera un RET de TCheques por aplicación de la regla 10. En el caso de las clases TStock y TArticulos se aplica se aplica la interpretación de Cantone *et al.* [3] para la regla 8, donde una alternativa es considerar un archivo lógico por cada clase participante en la agregación (cada clase contribuye por separado a la cuenta de archivos lógicos y luego no se toma en consideración la agregación). Fue agregado un archivo lógico TMensajes para representar los mensajes de confirmación, verificación y error, aspecto no contemplado en el diagrama de clases (regla 11).

El conjunto definitivo de transacciones fue obtenido aplicando las reglas 12 - 13 al listado inicial y excluyendo las transacciones repetidas y las que no cumplen con la definición de proceso elemental de IFPUG. Este proceso permitió identificar 84 transacciones. Para clasificarlas como Entrada Externa (EI), Consulta Externa (EQ) y Salida Externa (EO) se usaron las reglas de IFPUG. Para seguir con el ejemplo, a partir de las transacciones candidatas de la Tabla 2 se obtuvo el listado de la Tabla 4.

Tabla 4. Transacciones identificadas y clasificadas para el UC Configuraciones

UC Configuraciones	Tipo
1. UC Gestionar usuario	
1.1. Crear usuario	EI
1.1.1. Ingresar password	EI
1.2. Modificar usuario	EI
1.2.1. Modificar contraseña	EI
1.3. Consultar usuario	EQ
1.4. Buscar usuario	EQ
1.5. Restaurar usuario	EI
1.6. Anular usuario	EI
1.7. Calcular comisión actual	EO
1.8. Listado de comisiones	EO
2. UC Gestionar configuraciones generales	
2.1. Ingresar datos de la empresa	EI
2.2. Modificar datos de la empresa	EI

Con respecto a los archivos lógicos, de la lista de 30 archivos lógicos candidatos se descartaron los siguientes: TDeCredito se considera un RET de TBancos, TConfiguraciones se considera RET de Empresa, TPassword se considera RET de Usuario y TCtaBancarias se considera RET de TBancos. Para clasificarlos se usan las reglas de IFPUG. Los 25 archivos lógicos (Tabla 3) restantes son Archivos lógicos Internos (ILFs). El archivo TMensajes no es mantenido por un proceso elemental, por lo tanto es un Archivo de Interfaz Externo (EIF).

Para determinar la contribución de transacciones y archivos se aplican las reglas 4, 6 y 14 que permiten identificar los Tipos de Datos Elementales (DETs), RETs y Tipos de Archivos Referenciados (FTRs). La complejidad y contribución de transacciones y archivos se determina aplicando las tablas de IFPUG.

Con respecto a las transacciones, el criterio adoptado para los UCs que figuran en el diagrama pero no se dispone de la descripción, fue considerarlos transacciones de complejidad baja. La Tabla 5 presenta un ejemplo de transacción de cada tipo y su contribución en FP.

Tabla 5. Un ejemplo de cada tipo de transacción del UC Configuraciones

1. Gestionar usuario	#DET	#FTR	Tipo	Compl.	FP
1.1. Crear usuario	17	4	EI	alta	6
1.3. Consultar usuario	16	3	EQ	media	4
1.7. Calcular comisión actual	4	3	EO	baja	4

En lo referido a la evaluación de la complejidad y contribución de los archivos lógicos, todos son de complejidad *baja* y su contribución es de 7 FPs. Por ejemplo, el ILF Empresas tiene 13 DETs y 3 RETs.

La Tabla 7 presenta el resumen del conteo de FP del sistema.

Tabla 7. Resumen de la cuenta de FP

Función	Compl. baja		Compl. media		Compl. alta		Total FP
	Nro.	FP	Nro.	FP	Nro.	FP	
EI	21	63	14	56	23	138	257
EO	5	20	0	0	0	0	20
EQ	10	30	10	40	0	0	70
Transacciones							347
ILF	25	175	0	0	0	0	175
EIF	1	5	0	0	0	0	5
Archivos							180
Total							527

6 MEDICIÓN DE LA APLICACIÓN CON EL MÉTODO IFPUG FPA

La medición se realizó de acuerdo al procedimiento propuesto en el manual para la Medición de Puntos Función [6]. Como se trata de una aplicación existente, 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 medición basada en UC. Es una aplicación software independiente cuyo límite está representado en la Figura 3.

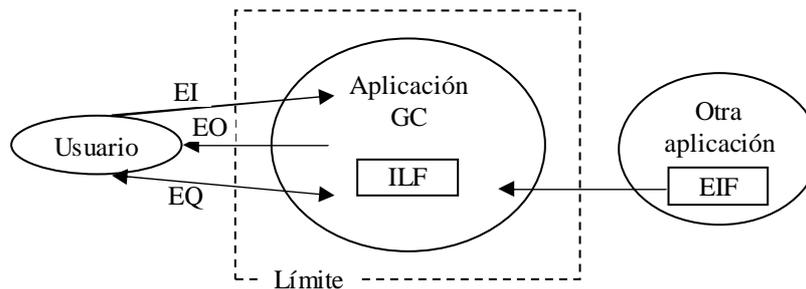


Figura 3. La aplicación desde el punto de vista del usuario [6]

A partir de la documentación disponible y la aplicación software instalada se identificaron, clasificaron y midieron las funciones transaccionales y de datos.

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 se identificaron 18 ILFs y 1 EIF.

Un ejemplo de ILF es *Empresas* que contiene 15 DETs y 2 RET (contiene 1 RET opcional - Configuraciones - que es requerido cuando se trata de la empresa local). Este ILF tiene complejidad *baja* y contribuye con 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 software.

Un ejemplo de EI es la transacción *Agregar nuevo artículo* (Figura 4). 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 *TArticulos* y los ILFs referenciados son *TRubros* y *Empresas* (este último se refiere a la empresa del *Proveedor*). Por lo tanto hay 3 FTR. Esta EI es de complejidad *alta* y su contribución es de 6 FPs.

The image shows a software window titled "Crear Artículo" with a "General" tab. It contains several input fields: "Código", "Código proveedor", "Descripción", "Material", "Punto de reposición", "Existencia máxima", "Precio costo", "Precio unitario", "Rubro", and "Proveedor". At the bottom, there are two buttons labeled "Aceptar" and "Cancelar".

Figura 4. Ventana "Crear Artículo"

La Tabla 8 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 8. Resumen de la cuenta de FP

Función	Compl. baja		Compl. media		Compl. alta		Total
	Nro.	FP	Nro.	FP	Nro.	FP	
EI	18	54	11	44	24	144	242
EO	4	16	2	10	2	14	40
EQ	10	30	22	88	1	6	124
Transacciones	32	100	35	142	27	164	406
ILF	18	126	0	0	0	0	126
EIF	1	5	0	0	0	0	5
Archivos	19	131	0	0	0	0	131
							537

Como puede observarse en la Tabla 8 existe una diferencia significativa con el número de transacciones de salida (EQ + EO) identificadas para la medición de UC (Tabla 7). Esto se explica porque, como fue señalado en la sección 6, no fueron considerados los UC que forman parte de una relación de generalización, que en todos los casos representaban operaciones de salida.

7 ANÁLISIS DE LOS RESULTADOS

El análisis de los resultados obtenidos (Tabla 9) tiene por objetivo hacer una evaluación de la capacidad de SFP como herramienta para estimar los FP de un sistema.

Tabla 9. Resumen de la medición de tamaño funcional

Técnica	FP
Tavares	527
IFPUG	537

En los siguientes párrafos se describen las principales observaciones respecto a la medición con la propuesta de Tavares:

1. El error relativo de la medición es del 2% con respecto a la medición de la aplicación por el método IFPUG.
2. En comparación con otras propuestas de estimación temprana:
 - Early Function Points Method [1] propone diferentes funciones de regresión para estimar los FP, una de las cuales está basada en el número de transacciones: $FP = 50,784 + 6,289 * (\#EI + \#EO + \#EQ)$. En nuestro caso, usando las transacciones identificadas desde los UC (83) se obtienen 573 FP. Este valor supera en un 9% al medido a partir de los UC.
 - Bundschuh [2] provee un modelo empírico que usa como entrada la suma del EIs y EOs (IO): $FP = 7.3 * \#IO + 56$. Usando el mismo criterio que en el caso anterior, $\#EI + \#EO$ es 63, lo que da como resultado 516 FP, que es un 2% menor que la medición de los UC.

8 CONCLUSIONES

En este artículo presentamos la evaluación de la exactitud de la medición de tamaño funcional basada en la especificación en UML de un sistema de gestión. Esta medición fue comparada con la medición de la aplicación de acuerdo con las reglas del método IFPUG.

A partir de medir la aplicación con el método IFPUG y usar ese resultado como valor de referencia se pudo determinar que el error relativo de la medición basada en UC es del 2%, el cual está dentro del rango de $\pm 10\%$ reportado en la literatura.

Este resultado confirma que la medición temprana de tamaño funcional producida a partir de los UC es suficientemente aproximada a la medición con el estándar IFPUG e incluso es mejor que la producida a partir de otras propuestas de estimación. Esta experiencia permitió ratificar las conclusiones de Tavares *et al.*

AGRADECIMIENTOS

Los autores agradecen al Departamento de Informática de la Facultad de Ingeniería de la UNPSJB por facilitar el material para esta investigación.

REFERENCIAS

- [1] 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.
- [2] Bundschuh M., "Function Point Prognosis", *Proceedings of FESMA 98*, May 1998.

- [3] Cantone, G., Pace, D., Calavaro, G., "Applying function point to unified modeling language: Conversion model and pilot study", *Proceedings 10th International Symposium on Software Metrics (METRICS'04)*, IEEE Computer Society, pp. 280–291, 2004.
- [4] Fetcke, T., Abran, A., Nguyen, T., *Mapping the OO-Jacobson Approach into Function Point Analysis*, Université du Québec à Montreal, Software Engineering Management Research Laboratory, 1998.
- [5] Harput, V., Kaindl, H., Kramer, S., "Extending Function Point Analysis of Object-Oriented Requirements Specifications", *11th IEEE International Software Metrics Symposium (METRICS'05)*, 2005.
- [6] IFPUG, Manual para la Medición de Puntos Función, Versión 4.1.1., AEMES, 2000.
- [7] International ISO/IEC Standard 14143-1, "Information technology - Software measurement - Functional size, Part 1: Definition of concepts", 1998.
- [8] Rumbaugh, J., Jacobson, I., Booch, G., "The Unified Modeling Language Reference Manual", *Addison-Wesley*, 1999.
- [9] 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.
- [10] van den Berg, K., Dekkers, T., Oudshoorn, R., Functional size measurement applied to UML-based user requirements, *Proceedings of SMEF 2005*, Roma, Italia, 2005.