

Estimación del Esfuerzo a Partir del Modelo de Datos

A.C. Gastón Jiménez¹

Facultad de Informática – UNLP
(1900) La Plata, Buenos Aires, Argentina
gasto.gimenez@hospitalitaliano.com.ar

A.C. Germán Gioia¹

Facultad de Informática – UNLP
(1900) La Plata, Buenos Aires, Argentina
g_gioia@yahoo.com

A.C. Jorge Perez Herrera¹

Facultad de Informática – UNLP
(1900) La Plata, Buenos Aires, Argentina
turcoph@yahoo.com

Lic. Rodolfo Bertone²

LIDI - Facultad de Informática – UNLP
(1900) La Plata, Buenos Aires, Argentina
pbertone@lidi.info.unlp.edu.ar

Lic. Hugo D. Ramón²

LIDI - Facultad de Informática – UNLP
(1900) La Plata, Buenos Aires, Argentina
hramon@lidi.info.unlp.edu.ar

Palabras Claves

Métricas, Punto Función, Modelo de Datos, Estimación, Esfuerzo, Planificación, Recursos Humanos

Resumen

La necesidad de medir el software que se va a construir, a fin de realizar planificación, estimación y análisis de riesgo del problema es una constante encontrada por toda organización dedicada a desarrollar software .

En este trabajo se presenta el desarrollo de una herramienta que realiza la estimación del esfuerzo para un desarrollo basado en métricas de punto función obtenidas a partir del modelo de datos.

La métrica de punto función (PF) se elige, básicamente, debido a que por su característica es independiente del lenguaje.

¹ Alumnos Licenciatura en Informática.

² Profesor Adjunto Dedicación Exclusiva

Introducción

Las métricas son necesarias para medir la productividad de organizaciones dedicadas al desarrollo de software. De la misma manera también son necesarias para estimar tamaño de nuevos productos a desarrollar. Estos dos aspectos son fundamentales en la planificación de nuevos proyectos [1].

Obtener una métrica temprana para un proyecto de software facilita la estimación de recursos (humanos, de hardware y software), la planificación y el análisis de riesgo del proyecto. Si la estimación del esfuerzo puede realizarse sobre el modelo de datos conceptual del sistema se logra obtener resultados probables en etapas tempranas del desarrollo. El objetivo de esta publicación está basado en el estudio de métricas y del modelado de datos para poder derivar mediciones desde este último [2].

Métricas de software

Para realizar estimación sobre un proyecto se necesita una métrica que mida la productividad del software. Es clara la desventaja que poseen las métricas orientadas al código en donde no tenemos independencia del lenguaje. La métrica ideal de productividad y esfuerzo, mide cantidad de funcionalidad por unidad de tiempo. El problema se reduce a cuantificar la funcionalidad [3].

Los punto función (PF) [4] cuantifican la funcionalidad de un software. La idea subyacente es encontrar un estimativo que caracterice completamente el sistema y se correlacione con la productividad observada. Las métricas de PF caracterizan la complejidad del software y de esta manera pueden utilizarse para saber el tiempo total del proyecto y cuantas personas debemos involucrar en el mismo.

El método de PF es empírico y de acuerdo a la experimentación esta técnica se adapta muy bien a las aplicaciones comerciales. De acuerdo al método PF, cinco ítems determinan la complejidad de una aplicación y el PF de la aplicación es una suma ponderada de esos cinco ítems. Estos ítems son:

Número de entradas	Número de archivos
Número de salidas	Número de interfaces
Número de consultas	

Los pesos relativos de cada ítems se derivaron en forma empírica y validados en observaciones en cientos de proyectos [5]. Además cada peso es multiplicado por un factor de ponderación de acuerdo a una escala, para cada caso, que puede ser simple, medio o completo. Con estos valores se calcula la cantidad de PF de un problema utilizando la siguiente relación:

$$PF = \text{cuenta_total_anterior} * (0,65 + 0,01 * \sum Fi)$$

donde F_i ($i = 1..14$) son valores de ajuste de complejidad según las respuesta a un cuestionario preestablecido y ponderado, donde las respuestas están en una escala entera que va de 0 (no influencia) a 5 (influencia esencial) [6] [7].

Modelado de datos

El modelado de datos responde a una serie de preguntas específicas importantes para cualquier aplicación de software. Algunas de ellas son: cuales son los objetos de datos primarios que va a procesar el sistema?Cuál es la composición de cada objeto de datos y que atributos describe el objeto?Cuál es la relación entre los objetos y los procesos que los transforman?. Por lo tanto, a partir de la definición del modelo es posibles realizar un conjunto de estimaciones respecto de la envergadura del proyecto en tratamiento.

Para responder descriptivamente a las preguntas anteriores, es posible construir un modelo, denominado Entidad Relación, el cual permite identificar los objetos de datos y sus relaciones mediante

una notación gráfica [8]. Luego, mediante una serie de pasos específicos es posible derivar el modelo final de datos, generalmente relacional, sobre el cual se pretende realizar las métricas de punto función tempranas del sistema de software [9].

Motivación para Implementar la Herramienta

La necesidad imperiosa de tener estimaciones tempranas para cuantificar un proyecto es una realidad para la organización de software. A partir de una definición temprana, e inicial, del modelo de datos, creemos que es posible derivar una estimación del esfuerzo necesario para la construcción del sistema informático [10].

Una herramienta como la que se presenta posteriormente en el artículo avanza hacia este objetivo. Si bien el modelo de datos es el resultado parcial de la etapa de análisis de un proyecto, creemos que en el momento de su desarrollo es aceptable tener nuevas estimaciones que permitan adaptar la planificación ya generada del sistema, permitiendo corregir parámetros que, en su defecto, producirán un aumento en los riesgos de producción del software.

Además, como existe un gran número de proyectos terminados y en producción de los cuales se puede tomar métricas postmortem y evaluar su modelo de datos con la herramienta desarrollada; estamos en condiciones de evaluar el desarrollo a fin de depurar y analizar las distorsiones encontradas. En la sección de conclusiones se presentan algunos resultados obtenidos y se los cotejan contra los resultados reales.

En la siguiente sección se describen las principales características tenidas en cuenta para el desarrollo del sistema.

Definición del Modelo de Implementación

Las métricas de punto función presentan un conjunto de características que permiten realizar estimaciones realistas de un proyecto de software, permitiendo tratar con características externas del software y que fueran importantes para los usuarios. Además pueden relacionarse con la productividad económica de la organización y ser independientes del código o lenguajes de programación seleccionado. De aquí que el modelo de métricas seleccionado es PF.

Si bien la medición es una tarea que comienza con el proyecto de software mismo y se repite en cada etapa del proceso de desarrollo, es importante poder realizarlas lo más temprano posible en el ciclo de vida del proyecto y tratar de derivarla de las características fundamentales del mismo. El modelo de datos es una muy buena fuente de información para obtener este tipo de resultado. Sin embargo, el ajuste de la complejidad debe requerir intervención por parte del usuario, basándose en la medida cuantitativa del dominio de la información que se representa.

Se ha propuesto la construcción de un ambiente de diseño de modelos de datos relacional y la recolección automática de las métricas de PF a partir del mismo.

Las reglas usadas en la herramienta se basan en técnicas de análisis de PF ayudando a medir en un modo estándar el tamaño y la complejidad de las aplicaciones de software, esta técnica está principalmente aplicada al desarrollo de proyectos de software de negocios. Los principios de las técnicas pueden ser divididos en tres partes diferentes basados en el procesamiento de la información: datos, transacciones y complejidad. Los datos son divididos en entidades internas y externas; las transacciones

consisten en entradas, salidas y consultas; y la complejidad corresponde a los factores (14) que relacionan el alcance, también denominados características generales del sistema. Figura 1.

Presentación de la Herramienta

La herramienta, como se adelantó en el apartado anterior, consta de dos funciones o partes. Estas son: (1) un ambiente de diseño de modelos de datos relacionales, (2) un ambiente que permite la evaluación de la métrica para obtener el PF del proyecto. Además permite la actualización de los datos de referencia usados en la evaluación del PF.

Ambiente de diseño de modelos de datos relacionales

El propósito de este ambiente es permitir la creación de modelos de datos relacionales, almacenando nombres de tablas, estructuras de tablas (nombres de campos, nombres de índices, estructuras de índices) y relaciones entre tablas. El asistente de evaluación de la métrica utiliza los modelos generados para la obtención del punto función.

La figura 2 presenta el esquema de creación del modelo de datos. Se debe tener en cuenta que en la creación de relaciones entre tablas hemos considerado los tipos de dependencia en función de la necesidad de demostrar una relación entre una entidad fuerte y una débil, o relaciones de referencia entre entidades.

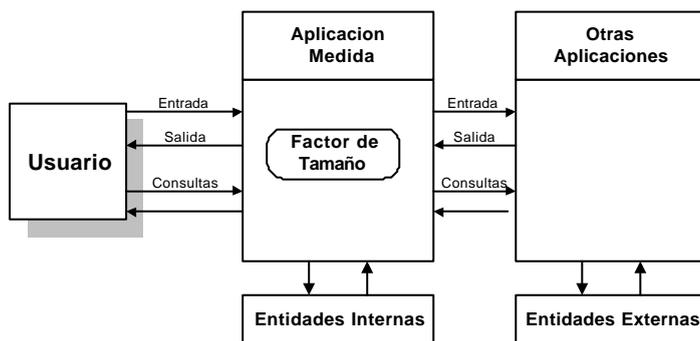


Figura 1

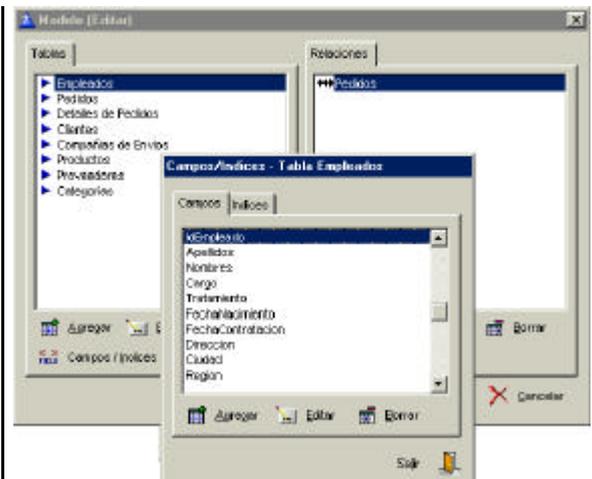


Figura 2

Ambiente para la evaluación de la métrica y obtención del punto función del proyecto

Este ambiente, es un asistente que genera la evaluación total de la métrica para un modelo de datos determinado, devolviendo como resultado final el punto función del proyecto. Figura 3.

Mediante una serie de pasos, el usuario podrá ir contestando una sucesión de preguntas que determinan los valores subjetivos y el ajuste de la complejidad que en conjunto con el cálculo automático de las reglas de la métrica determinarán el PF del proyecto.

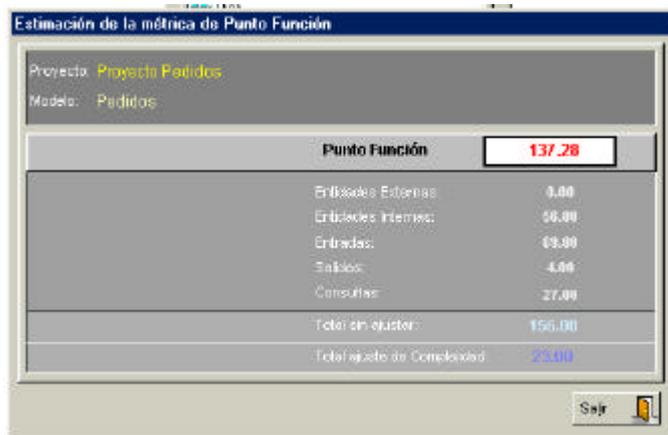


Figura 3

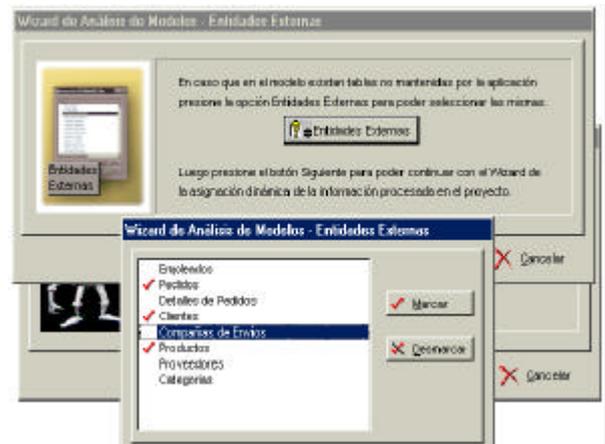


Figura 4

Reglas para la medición de Datos: Entidades internas

Una entidad es interna si la aplicación, que es el resultado del proyecto evaluado, les permite a los usuarios crear, borrar, modificar y (o) leer los ítem de la entidad. Los usuarios deben de haber pedido, y tener el conocimiento completo de esta función para ser considerada.

La complejidad de una entidad es basada en el número de ítem y número de relaciones entre las entidades. Todos los ítem de una entidad son computados. El modelo relacional determina el número de relaciones dentro de la entidad.

La regla de determinar la complejidad es como sigue:

Relaciones	Items		
	1-19	20-50	51+
1	Baja	Baja	Promedio
2-5	Baja	Promedio	Alta
6+	Promedio	Alta	Alta

Reglas para la medición de Datos: Entidades externas

Las entidades usadas pero no mantenidas por la aplicación son las entidades externas. Si la aplicación no mantiene los datos encontrados en la entidad, se habla de una entidad externa. Pueden tener su origen en una bases de datos compartida, un archivo común o una tabla de la referencia copiada en los programas. Calcular ítem y relaciones es idéntico que para las entidades internas, sólo que el número de puntos asignados para la complejidad es diferente. Figura 4

Reglas para la medición de Transacciones

Las transacciones representan el aspecto dinámico de la información procesada en una aplicación, a diferencia de los datos, los cuales son conservados. Las transacciones son clasificadas como entradas, salidas o consultas. Hay dos aproximaciones disponibles para evaluar el número de transacciones:

- Comenzando de los modelos del proceso

- Comenzando de las pantallas e informes como aparecen al usuario.

Ambas aproximaciones llevan a resultados similares. La primer aproximación (modelos) se utiliza cuando el administrador del proyecto desea estimar su proyecto y el equipo de desarrollo no ha diseñado las pantallas e informes. La segunda aproximación es preferible a la primera si las pantallas e informes están disponibles porque es más concreta y lleva la relación más íntima a lo que es realmente cumplido. Nosotros utilizaremos la primera aproximación ya que nos concentramos en una etapa temprana del desarrollo que no cuenta con diseño de la interfaz del sistema.

Entradas

Una entrada introduce ciertos cambios en los datos encontrados en la aplicación. El usuario genera entrada. Los cambios pueden producirse a través de una entrada directa que usa una pantalla, o indirectamente, usando una transacción de actualización de archivo de otra aplicación. La entrada de los datos también puede lograrse a través de un proceso de lote (batch). En todos los casos hay cambio en los datos, si es una alta, una modificación o un borrado.

La regla por determinar la complejidad es como sigue:

Entidades	Items		
	1-4	5-15	16+
1	Baja	Baja	Promedio
2	Baja	Promedio	Alta
3+	Promedio	Alta	Alta

Debemos computar una pantalla (lógica) de entrada para cada una de las operaciones (agregar, modificar, borrar) que se ejecuten en cada Grupo de Actualización.

La cantidad de ítem considerados para las operaciones agregar y modificar es la suma de los campos de las tablas pertenecientes al Grupo de Actualización, mientras que para la operación de borrado son considerados los campos de la clave primaria de la tabla principal del Grupo de Actualización. A estos campos habría que sumarle los campos de otras tablas no pertenecientes al Grupo que puedan sufrir cambios.

La cantidad de entidades considerados para las operaciones agregar, modificar y borrar es la cantidad de tablas pertenecientes al Grupo de Actualización. A estos habría que sumarle otras tablas no pertenecientes al Grupo que puedan sufrir cambios. Figura 5

Salidas

La salida es un resultado producido por la aplicación. Se produce como una parte regular de procesamiento independientemente de condiciones establecidas en los datos. Un informe mensual se produce sin tener en cuenta lo que ha pasado. El usuario comienza las preguntas directa o indirectamente.

Diferentes conjuntos de atributos presentados con una tecnología diferente no tomados en cuenta como distintas salidas (por ejemplo: si un informe se produce en el papel y en microfilm, no se cuenta como dos operaciones de salida). La regla para determinar la complejidad es como sigue:

Entidades	Items		
	1-5	6-19	20+
1	Baja	Baja	Promedio
2-3	Baja	Promedio	Alta
4+	Promedio	Alta	Alta

Consideramos, para un Grupo de Actualización, generar salidas para cada uno de los índices seleccionados por el usuario de la tabla principal del Grupo.

Para los ítem se tiene en cuenta todos los campos de las tablas que conforman el Grupo de Actualización menos los campos de las claves foráneas de cada una de las tablas secundarias usadas en la relación con la tabla principal. Para las entidades son tenidas en cuenta todas las tablas del Grupo de Actualización. Figura 6



Figura 5

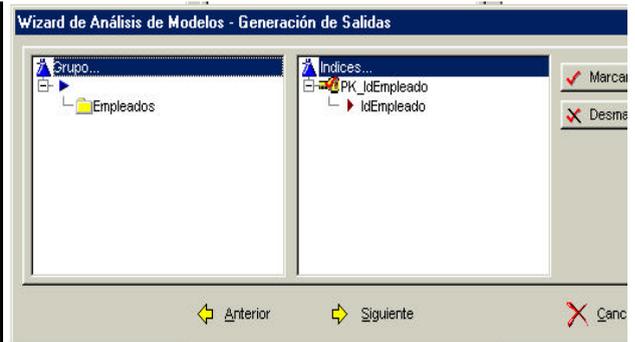


Figura 6

Consultas

Una consulta es una salida producida en la respuesta a un requerimiento (entrada). Esta transacción se caracteriza por el hecho que no modifica las entidades y no es el resultado de una modificación. Puede presentarse usando una pantalla o un informe, sin importar la técnica. La consulta involucra una respuesta directa a un requerimiento, pero no necesariamente una respuesta inmediata.

La regla por determinar la complejidad es dual. Para la parte de entrada de datos, la regla es como sigue:

Entidades	Items		
	1-4	5-15	16+
1	Baja	Baja	Promedio
2	Baja	Promedio	Alta
3+	Promedio	Alta	Alta

Para la parte de salida de datos, la regla es como sigue:

Entidades	Items		
	1-5	6-19	20+
1	Baja	Baja	Promedio
2-3	Baja	Promedio	Alta
4+	Promedio	Alta	Alta

El cálculo final es basado en la selección de la parte más compleja de la consulta (entrada o salida). En la práctica, el lado de la salida es siempre más complejo que el lado de la entrada.

Reglas para la medición de la complejidad

Estos factores son los propuestos por la métrica de PF y su forma de evaluación respeta los lineamientos de esta métrica.

Resultados Obtenidos

Se presentan a continuación las mediciones obtenidas sobre dos proyectos de software ya terminados y se compara los resultados obtenidos con las métricas postmortem realizadas.

El primer proyecto consistió en un sistema de Ventas. Dicho sistema se puede catalogar de complejidad baja. En líneas generales el proyecto consistió en informatizar las ventas de una organización, para lo cual se disponía información respecto de clientes, proveedores y empleados de la organización, los productos que se comercializan en la misma están categorizados en diversos grupos y pueden ser enviados al cliente si la compra se realiza vía telefónica o electrónica.

Los resultados obtenidos con la herramienta arrojaron 206 PF. Estos resultados fueron obtenidos por un grupo de trabajo ajeno al desarrollo del proyecto, de aquí que las ponderaciones subjetivas de la métrica de PF fueron evaluadas sin la influencia de los desarrolladores finales. Las métricas postmortem obtenidas para este proyecto fueron evaluadas en 225 PF. La ponderación del ajuste de la complejidad marco, en cierta medida la diferencia, y algunas consideraciones de manejo de información que divergieron respecto de las especificaciones finales obtenidas.

El segundo proyecto evaluado es un sistema de seguimiento de expedientes. Este caso tiene como principal característica que los estados de los expedientes son consultados desde diversas oficinas y por un número variado de usuarios. Cada expediente resume su información en: tema, iniciador y destino actual del expediente, detalles del contenido, fechas de generación, entrada y cambio de destinos. Además, los expedientes “viejos” se archivan con un formato especial. Los diferentes expedientes están relacionados entre sí, lo que motiva un mayor movimiento de información cuando se opera con ellos.

La somera descripción anterior tiene por objetivo indicar que este proyecto difieren en su procesamiento con el presentado anteriormente. El análisis llevado a cabo con la herramienta dio como resultado 299 PF y las evaluaciones obtenidas postmortem arrojaron 350 PF. En este caso la diferencia fue más apreciable que con el primer ejemplo. En este caso, se observó una mayor incidencia de los factores de corrección de complejidad. Las especificaciones del problema no fueron lo suficientemente rigurosas para describir con el suficiente nivel de detalle las características del procesamiento de los expedientes.

Conclusiones y trabajos futuros

Se evaluaron una serie de proyectos de software que se pueden catalogar de complejidad medio-baja, se presentaron algunos de los resultados obtenidos en el apartado anterior. A partir de estos resultados obtenidos mediante la herramienta y métricas postmortem realizadas es posible observar que la herramienta obtiene estimaciones aceptables, teniendo en cuenta que por aceptable entendemos una distorsión baja respecto de la realidad.

Si bien debemos considerar los resultados parcialmente, debido a que el número de proyectos evaluados hasta el momento es pequeño; creemos que nuestra herramienta se adecua perfectamente a las necesidades básica de una organización que desarrolle software hasta mediana complejidad. Las distorsiones observadas se debieron, en algunos casos, a la incorrecta ponderación de algunos aspectos subjetivos de la métrica de Punto Función.

Las evaluaciones realizadas fueron hechas en su totalidad sobre proyectos ya terminados, esto significa que en todos los casos se evaluaron modelos de datos en producción. Por lo tanto, es esperable una distorsión residual cuando se utilice la herramienta para la evaluación temprana de un sistema debido a que el modelo de datos definido puede no ser el modelo final de producción.

Las siguientes etapas del proyecto se dividen, básicamente en tres áreas:

Utilizar la herramienta para evaluar tempranamente desarrollos de software y comparar los resultados con métricas posmortem de esos desarrollos.

Tomar proyectos de mediana – alta complejidad y ponderar nuevamente los resultados.

Permitir derivar automáticamente el modelo de datos de un problema real hacia la definición del mismo para la herramienta.

Bibliografía

- [1] Ghezzi, C., Fundamentals of Software Engineering, Prentice Hall, 1991.
- [2] Abiad, S; Haraty, R; Mansour, N; Software metrics for small database applications Proceedings of the 2000 ACM symposium on Applied computing 2000 (volume 2) March 19 - 21, 2000, Como Italy Pages 866-870
- [3] Kemerer, C; Reliability of function points measurement: a field experiment, Communications of the ACM Volume 36 , Issue 2 (1993) Pages 85-97
- [4] Garmus, D., Function Point Analysis: Measurement Practices for Successful Software Projects, Addison-Wesley Information Technology Series, 2000.
- [5] Putnam, Measures for Excellence, Prentice Hall, 1992 .
- [6] Pressman, R; Ingeniería de Software, Un enfoque práctico. Cuarta edición. Mc Graw Hill. 1998
- [7] Florac, W'; Measuring the Software Process, Addison-Wesley Pub Co, 1999.
- [8] Silberschatz, A; Korth, H; Sudarshan, S; Fundamentos de bases de Datos, Tercera Edición, McGraw Hill, 1998
- [9] Hansen, G; Hansen, J; Diseño y Administración de Bases de Datos, Segunda Edición, Prentice Hall, 1997
- [10] Widmaier, J; Producing more reliable software: mature software engineering process vs. state-of-the-art technology?; International Conference on Software Engineering Proceedings of the 22nd international conference on Software engineering June 4 - 11, 2000, Limerick Ireland Pages 88-93