# Reparto de la carga de trabajo en la realización de prácticas en grupo mediante una herramienta de estimación

Macario Polo, Mario Piattini y Francisco Ruiz
Escuela Superior de Informática – Universidad de Castilla-La Mancha
Paseo de la Universidad, 4
13071-Ciudad Real
macario.polo@uclm.es

**Resumen.**
*En la calificación final de muchas asignaturas se pondera, junto a la nota del examen final o de los posibles parciales, la realización de un trabajo práctico, que a veces puede realizarse en grupo. Es deseable que, en estos casos, los diferentes alumnos que realizan el trabajo dispongan de algún método que les permita realizar, a priori, un reparto lo más equitativo posible, de manera que cada uno realizara un esfuerzo aproximadamente similar al del resto de compañeros.*
*En este artículo se presenta una herramienta que, con el fin mencionado, hemos empezado a utilizar este año en Ingeniería del Software II, de quinto curso. La herramienta realiza una estimación del tamaño en puntos-función de un sistema orientado a objetos utilizando el método de Antoniol et al. [2]. Un beneficio añadido del uso de la herramienta es la percepción, por parte de los alumnos, de la utilidad de las métricas de software, que habitualmente son simplemente mencionadas en las clases de teoría.*

## 1. Introducción.

En la evaluación de muchas asignaturas interviene, además de la calificación obtenida en los exámenes, la nota obtenida en algún trabajo práctico, que puede ser a veces realizado en grupo. El reparto de las cargas de trabajo entre los diferentes alumnos debe ser lo más equitativo posible, de manera que todos dediquen aproximadamente el mismo esfuerzo al desarrollo del trabajo.

Dependiendo de la asignatura, es posible que los alumnos conozcan ya el concepto de métrica y que, incluso, conozcan detalles sobre algunas de las métricas más conocidas. En el caso de Ingeniería del Software II, del 5º curso de Ingeniería en Informática de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha, el estudio en detalle de algunas métricas forma parte de los contenidos del primer parcial. Además, algunos alumnos conocen métricas adicionales utilizadas en gestión de proyectos, en el caso de que hayan cursado en 4º Planificación y Gestión de Sistemas de Información. El segundo parcial de Ingeniería del Software II se dedica a desarrollo orientado a objetos, haciendo un énfasis especial en la aplicación de patrones de diseño. Como práctica del segundo parcial, los alumnos deben desarrollar una aplicación completa por parejas, debiendo entregar al profesor toda la documentación generada, lo que incluye diagramas de casos de uso, algunos diagramas de interacción, algunos diagramas de estado y uno o más diagramas de clases.

Con el objetivo de que los dos compañeros dediquen el mismo esfuerzo al desarrollo de la práctica, y teniendo en cuenta que en las clases de laboratorio se utiliza la herramienta Rational Rose (aunque puede utilizarse cualquier otra, como Poseidon, que es además gratuita), hemos desarrollado una herramienta que es capaz de realizar una estimación del tamaño en puntos-función del sistema que se debe desarrollar, a partir del diagrama de clases dibujado con Rational Rose. El método utilizado para calcular el número de puntos-función es el descrito por Antoniol et al. [2].

Éste es el primer año en que utilizamos esta herramienta, por lo que no disponemos de datos propios sobre la bondad de las estimaciones realizadas. Los autores del método, sin embargo, dedican gran parte de su artículo a la validación empírica de las estimaciones, ofreciendo excelentes resultados. No obstante, entendemos que hay suficientes diferencias entre los proyectos evaluados (proyectos industriales frente a prácticas de una asignatura) como para revisar,

ajustar y adaptar el método a nuestro entorno docente.

A pesar de esta carencia de resultados propios sobre las estimaciones realizadas, los alumnos perciben claramente los beneficios del uso de métricas en un proyecto real.

El resto de este artículo se organiza de la siguiente manera: en la Sección 2 presentamos resumidamente el método utilizado por la herramienta para realizar la estimación del número de puntos-función. Las características de la herramienta y su utilización en prácticas se presentan en la Sección 3; por último, presentamos en la Sección 4 nuestras conclusiones y trabajos futuros.

## 2. Breve descripción del método implementado para calcular puntos-función.

Antoniol et al. [2] proponen la utilización del diagrama de clases de un sistema orientado a objetos para realizar una estimación de su futura funcionalidad mediante el uso de puntos-función. Proponen cuatro variantes para realizar la estimación, si bien obtienen resultados muy parecidos con todas ellas. Es por esto por lo que, para desarrollar la herramienta, hemos implementado la variante más sencilla.

En esta variante, cada clase se cuenta como un Internal Logical File (ILF), y cada actor como un External Input File (EIF). Cuando estos conjuntos han sido cargados con los elementos adecuados, se recorren de la siguiente manera:

- Inicialmente, a cada ILF y EIF se le asigna el valor RET=1. Por cada atributo simple del elemento se cuenta un DET, y se le suma un RET por cada asociación o agregación de cardinalidad mayor que uno.
- Si el número de DET es menor que 51, se le asigna complejidad Low (L), y Average (A) en caso contrario.

A continuación se consideran los métodos de cada ILF y EIF. Los autores agrupan las External Inputs, External Outputs y External Inquiries bajo la única denominación de "Service Requests" (SR). Cada método concreto (no abstracto) del ILF o EIF considerado se considera una SR, asignándole un DET por cada argumento simple, y un RET por cada argumento complejo.

A cada SR se le asigna complejidad Low, Average o High de acuerdo con la Tabla 1:

| | FTR<=1 | FTR<=2 | FTR>2 |
|---|---|---|---|
| DET<=4 | Low | Low | Average |
| DET<=15 | Low | Average | High |
| DET>15 | Average | High | High |

Tabla 1. Asignación de complejidad a las Service Requests.

Una vez calculada la complejidad de todos los elementos (ILFs, EIFs y SRs), se calcula el número de Puntos-función sin ajustar (Unadjusted function-points) del sistema según el número de elementos de cada tipo (ILF, EIF o SR) y su complejidad. Para hacer este cálculo se rellenan las casillas sombreadas de la Tabla 2:

| | Low | | | | Average | | | | High | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tipo de elemento | Cantidad | | Peso | | Cantidad | | Peso | | Cantidad | | Peso | | |
| SR | | X | 3 | + | | X | 5 | + | | X | 7 | = | |
| ILF | | X | 7 | + | | X | 10 | + | | X | 15 | = | |
| EIF | | X | 5 | + | | X | 7 | + | | X | 10 | = | |
| | | | | | | | Nº de puntos -función sin ajustar: | | | | | | |

Tabla 2. Pesos para calcular el número de puntos-función sin ajustar.

Una vez calculado el número de Puntos-función sin ajustar, el usuario puntúa, de acuerdo ya a las normas de Albretch [1] y del IFPUG (International Function-Points User Group) un conjunto de 14 parámetros correspondientes a 14 características generales del sistema (véase Tabla 3) con un valor entre 0 y 5, según el grado de influencia de la característica en el sistema (Tabla 4).

1. Comunicación de datos
2. Procesamiento distribuido
3. Perfomance (desempeño)
4. Configuración del equipamiento
5. Volumen de transacciones
6. Entrada de datos *on-line*
7. Interfase con el usuario
8. Actualización *on-line*
9. Procesamiento complejo
10. Reusabilidad
11. Facilidad de implementación
12. Facilidad de operación
13. Múltiples locales
14. Facilidad de cambios

Tabla 3. Características generales del sistema.

| Descripción | Nivel de influencia |
|---|---|
| No está presente o no influye | 0 |
| Influencia mínima | 1 |
| Influencia moderada | 2 |
| Influencia media | 3 |
| Influencia significativa | 4 |
| Influencia fuerte | 5 |

Tabla 4. Niveles de influencia de las "características generales del sistema".

La suma de los catorce niveles de influencia se utiliza en la siguiente ecuación para calcular el parámetro "Factor de ajuste":

Factor de ajuste = (Nivel de influencia *0,01) + 0,65

Ecuación 1. Ecuación para el cálculo del factor de ajuste.

Calculado el factor de ajuste, se obtiene el resultado esperado, que es el Número de puntos-función ajustados:

Puntos-función ajustados = Puntos-función sin ajustar X Factor de ajuste

Ecuación 2. Calculo del número de puntos-función ajustados.

## 3. Descripción de la herramienta y su uso para la realización de prácticas.

Como se observa, el proceso del cálculo de puntos-función es complicado. Afortunadamente, nuestra herramienta automatiza la mayoría de los pasos.

La Figura 1 muestra la ventana principal de la aplicación. Permite la apertura de cualquier modelo Rational Rose (ficheros con extensión .mdl). Mediante el botón "Load", la herramienta carga la lista de paquetes existentes en el modelo. Es posible que algunos de los paquetes no deban ser evaluados, por estar siendo completamente reutilizados. En el ejemplo mostrado en la figura se han descartado los paquetes estándar de java (aparecen en la lista "Discarded packages"), de manera que la herramienta evaluará únicamente las clases que deban ser desarrolladas por los alumnos.

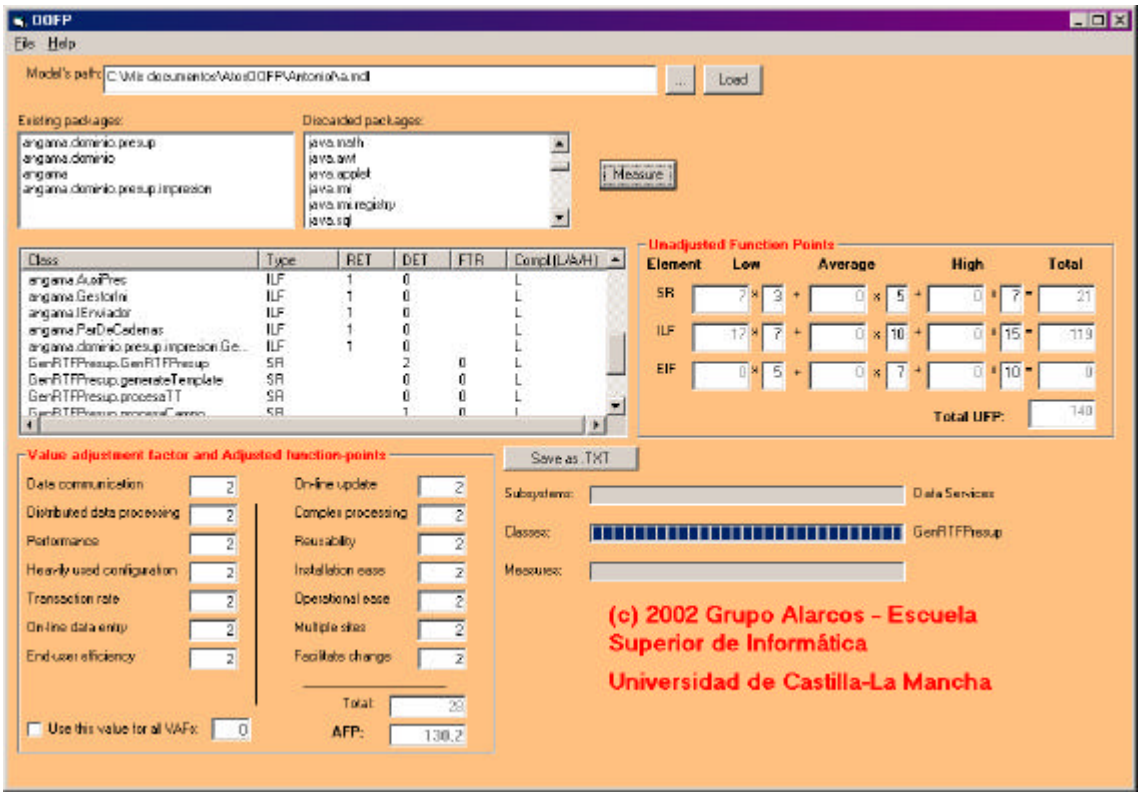


Figura 1. Ventana principal de la aplicación.

Una vez seleccionados los paquetes que se desean evaluar, se arranca el proceso de medida mediante el botón "Measure", que lanza las operaciones necesarias para realizar la medición según el método descrito en la sección anterior.

Los elementos evaluados, su clasificación como ILF, EIF o SR y su complejidad (Low, Average o High) aparecen en la lista que ocupa el lateral izquierdo de la pantalla. Los puntos-función sin ajustar son calculados automáticamente y mostrados en la tabla etiquetada "Unadjusted function-points". El usuario debe dar un valor entre 0 y 5 a las catorce características generales del sistema, con lo que el número de puntos-función ajustados aparece automáticamente en la caja de texto etiquetada "AFP" (130,2 en el ejemplo).

### 3.1 Uso de la herramienta por los grupos de alumnos.

Los alumnos deben disponer, evidentemente, del diagrama de clases que representa la estructura de su práctica con un cierto nivel de detalle (clases, métodos y parámetros de los métodos). Deben conseguir estructurar el trabajo en paquetes, de tal manera que el número de paquetes a desarrollar por cada alumno tenga un tamaño en puntos-función aproximadamente igual.

La experiencia de utilización en clase comienza con una breve descripción del método de los puntos-función, explicando un poco más en detalle el método de cálculo implementado en la herramienta. Realmente, se percibe en los alumnos mucha curiosidad por utilizar una herramienta de estimación "de verdad", y se observa que salen del aula de prácticas muy

satisfechos, probablemente al comprobar que las métricas tienen aplicaciones bien concretas y útiles, y que no son sólo meras fórmulas de dudosa aplicación.

## 4. Conclusiones y trabajos futuros.

La herramienta presentada permite, por un lado, repartir equitativamente el esfuerzo que deben dedicar los alumnos a desarrollar las prácticas de Ingeniería del software; y, por otro, les hace ver que las métricas de software son herramientas que tienen verdaderamente mucha utilidad.

La métrica elegida para realizar la estimación (funcionalidad de la aplicación en puntos-función) ha sido criticada por varios autores (por ejemplo, [3]); sin embargo, su utilización en las empresas está muy extendida y en la actualidad se han realizado múltiples adaptaciones para su utilización en el paradigma orientado a objetos ([2][4][5]).

Para finales de este curso dispondremos de datos que nos permitan evaluar la validez de las predicciones de este método en el entorno tan particular en que nos encontramos. Muy probablemente, deberemos realizar ajustes al método para aproximar las predicciones a los valores reales de funcionalidad. Podremos, posiblemente, encontrar una equivalencia válida entre puntos-función y líneas de código, que permita a los alumnos realizar estimaciones de tamaño seguras.

Esta herramienta se encuentra a libre disposición de toda la comunidad universitaria. Esperamos que otros compañeros la descarguen y utilicen, que nos envíen resultados cuantitativos de su aplicación, de manera que podamos mejorar los ajustes y estimaciones para ofrecerlos en una nueva edición de Jenui.

## Referencias.

[1] Albretch, AJ. (1979). *Measuring application development productivity.* Proc. of the IBM Application Development Symposium (pp. 83-92). Monterrey, Canadá.

[2] Antoniol, G., Lokan, C., Caldiera, G., y Fiutem, R. (1999). A Function Point-Like Measure for Object-Oriented Software. Empirical Software Engineering, 4(3), 263-287.

[3] Dolado, J. y Fernández, L. (1999). ¿Merece la pena usar los puntos de función? Novática, nº 140, 57-62.

[4] Sneed, H. (1999). Project control for software quality. En Proc. of the EXCMO./SCOPE Conference. Shaker Publishing.

[5] Uemura, T., Kusumoto, S. y Katsuro, I. (2001). Function-point analysus using design specifications based on the Unified Modelling Language. Journal of Software Maitnenance and Evolution: Research and Practice, 13(4), 223-243.