

Análisis comparativo de métodos de estimación basados en puntos de función para proyectos web

Gladys Dapozo, Yanina Medina, Berenice Lencina, Gabriel Pedrozo Petrazzini
Facultad de Ciencias Exactas y Naturales y Agrimensura
Universidad Nacional del Nordeste, Av. Libertad 5450, 3400, Corrientes,
Corrientes, Argentina
{gndapozo, yanina}@exa.unne.edu.ar

Abstract. Una de las tareas más importantes en la gestión de proyectos de software consiste en la estimación del esfuerzo, costo y duración que demandará el mismo. La falta de precisión en la estimación puede ocasionar en las empresas de software incumplimiento de plazos, entrega de productos incompletos, aumento del precio final y pérdida de competitividad. En este trabajo se muestra el resultado de la aplicación de diferentes métodos de estimación, basados en puntos de función, aplicado a proyectos de desarrollo web realizados por los alumnos de la asignatura Taller de Programación I. Los resultados indican que a mayor especificidad en cuanto al tipo de proyecto y la adecuada configuración de parámetros que identifican el entorno de aplicación, el ajuste de la estimación es mayor.

Keywords: Gestión de proyectos de software. Estimación de software. Estimación web.

1 Introducción

Según McConnell [1] “una estimación es una predicción de cuánto tiempo durará o costará un proyecto”, constituye la base para la planificación de los proyectos. El desarrollo del software requiere de la estimación para controlar y administrar los recursos que se necesitan utilizar antes y durante el proyecto. Son numerosas las variables, relacionadas con los recursos humanos, el contexto y las políticas que intervienen en su proceso que pueden afectar los resultados finales.

Estimar adecuadamente el tiempo, costo y recursos requeridos para realizar un proyecto de desarrollo o mantenimiento de software, es un reto que enfrentan hoy en día las empresas de software. Las pequeñas y medianas empresas de software que inician proyectos de mejora de sus procesos, se enfrentan con el desafío de seleccionar, por una parte, el método de estimación más adecuado a su contexto y, por otra, adoptar mejores prácticas que permitan ir consolidando una base histórica de estimación, con el propósito de disminuir cada vez más la desviación entre los valores estimados y reales [2].

Existen diversos métodos para estimar el esfuerzo de desarrollo, que son clasificados en dos categorías: la primera comprende a los métodos paramétricos, son aquellos en los cuales el proceso de cuantificación del resultado está basado en un

proceso mecánico, por ejemplo la aplicación de una fórmula derivada de los datos históricos. La segunda categoría, comprende a los métodos heurísticos, son aquellos en los cuales la cuantificación del resultado se produce a partir del juicio y/o la experiencia de un experto [3]. En un trabajo previo de los autores se elaboró un cuadro para resumir los principales métodos reportados en la literatura [4].

Estimación basada en puntos de función:

Los Puntos de Función (PF) constituyen una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la cantidad de funcionalidad requerida y entregada a los usuarios. Miden el tamaño lógico o funcional de los proyectos o aplicaciones de software basado en los requerimientos funcionales del usuario.

El método de Puntos de Función fue publicado por primera vez en el año 1979 por Allan J. Albrecht [5] y se obtienen utilizando una relación empírica basada en medidas cuantitativas del dominio de información del software y valoraciones subjetivas de su complejidad. Pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la explotación y mantenimiento.

Los proyectos de ingeniería web adoptan normalmente el modelo de proceso ágil. Por este motivo, es frecuente utilizar una medición de puntos de función modificada en conjunto con los pasos de la estimación en proyectos ágiles.

A continuación, se describen algunos métodos basados en puntos de función.

FP-Lite

La propuesta de FP-Lite deriva del método de análisis de puntos de función definido por el Grupo Internacional de Usuarios de Puntos de Función (IFPUG), simplificando los cálculos para obtener los puntos función -que requerían de mucha documentación que no siempre está disponible-, y dejando como opcional la aplicación del factor de ajuste, dado que el aporte a la precisión que otorga no compensa el esfuerzo que se necesita para calcularlo. Este método puede ser aplicado tanto a proyectos en desarrollo o proyectos de mantenimiento.

El método parte de la descripción de los requerimientos del sistema, a partir de los cuales se identifican los límites de la aplicación desde el punto de vista del usuario. Luego se procede a determinar los cinco elementos funcionales: Ficheros Lógicos Internos (ILF), Ficheros Lógicos Externos (ELF), y aquellas transacciones que son entradas (EI – entradas externas), salidas (EO – salidas externas) o consultas (EQ – consultas externas).

Para el cálculo del valor final de los PF se asigna un peso a cada elemento considerando solo la complejidad media establecida por IFPUG:

$$PF = \sum_{i=1}^n E_i C_i$$

Ei es cada elemento funcional y Ci la complejidad de cada uno

Para la estimación del esfuerzo (horas-hombre) y duración (meses) se utiliza la fórmula: $C \times PF^E$, donde C y E son factores de corrección y dependen de diferentes características del proyecto. Los valores de estas constantes para cada característica se detallan en [6].

Puntos de Casos de Uso (UCP)

La metodología UCP se basa en la utilización de casos de uso como dato de entrada para calcular el esfuerzo -en horas-hombre (hh)- necesario para el desarrollo de un proyecto de software [7].

El método utiliza cuatro variables principales:

- Actores (Ac): se evalúa su característica intrínseca y se determina la forma en que interactúan con el sistema a desarrollar, asignándole un peso: los simples tienen 1, los medios 2, y los complejos 3.
- Casos de uso (CU): Se clasifican en: *Simple* (peso=5) si tiene 3 o menos transacciones; *Medio* (peso=10) si posee de 4 a 7 transacciones; y *Complejo* (peso=15) si posee más de 7 transacciones.
- Factores técnicos de complejidad: Influencias técnicas que puedan afectar el proceso de desarrollo. Existen trece, cada uno ponderado en función de su impacto relativo en una escala de 0 a 5, donde 0 es irrelevante, 3 es medio, y 5 es esencial.
- Factores del entorno del proyecto: indican la influencia del grupo humano involucrado en el proyecto. Se utiliza un total de 8 factores, con una escala de 0 a 5 para ponderarlos, siendo 0 irrelevante, 1 un fuerte impacto negativo, 3 es promedio y 5 con un fuerte impacto positivo.

Los pasos a seguir para conseguir la estimación son los siguientes [8]:

- a) Revisar los aspectos clave de los requerimientos para calcular un recuento de Puntos Caso de Uso sin ajustar (UUCP – Unadjusted Use Case Points).

$$UUCP = \sum_{i=1}^n A_c P_i + \sum_{i=1}^m C_U P_i, \text{ donde } n \text{ es el n}^\circ \text{ de actores, } m \text{ el n}^\circ \text{ de CU y } P_i \text{ el peso} \quad (1)$$

- b) Evaluar los factores técnicos (TCF) y de entorno (ECF) para crear los factores de ajuste.

$$TCF = 0,6 + 0,01 \sum_{i=1}^{13} T F_i P_{o_i}, \text{ donde } T F_i \text{ es el peso del factor } i, \text{ y } P_{o_i} \text{ la ponderación} \quad (2)$$

$$ECF = 1,4 - 0,03 \sum_{i=1}^8 E F_i P_{o_i}, \text{ donde } E F_i \text{ es el peso del factor } i, \text{ y } P_{o_i} \text{ la ponderación} \quad (3)$$

- c) Ajustar los UUCP para obtener los Puntos Caso de Uso ajustados (UCP), que posteriormente se transformarán en una estimación de esfuerzo (horas-hombre).

$$UCP = UUCP * TCF * ECF \quad (4)$$

Llegado a este punto se obtendrá una aproximación del tamaño en puntos de casos de uso. Luego se calcula el esfuerzo considerando el Factor de Productividad, el cual indica la cantidad de horas-hombre necesarias para completar un punto de caso de uso. El autor establece un valor de 20 FP si no se cuentan con datos históricos en la empresa. Conseguido el esfuerzo se calcula la duración del proyecto en días.

$$\begin{aligned} \text{Esfuerzo} &= UCP * FP, \text{ donde } FP \text{ es el factor de productividad} \\ \text{Duración} &= \text{Esfuerzo} / \text{Horas por día} \end{aligned}$$

Webmo

En un intento por dar respuesta a las diferencias entre las estimaciones de desarrollos clásicos y de proyectos web, aparece WebMO, un modelo de estimación orientado a aplicaciones web basado en COCOMO II, propuesto por Reifer [9].

La unidad de medida de tamaño que utiliza es el “Web Object”, similar a los puntos de función, pero abarca además funciones específicas de las aplicaciones web, tales como: Links, Multimedia, Scripts, etc. [10].

Para calcular el esfuerzo y la duración, se utilizan las siguientes formulas:

$$\text{Esfuerzo} = A (\sum cd_i) (\text{tamaño})^{P1} \quad (5)$$

$$\text{Duración} = B (\text{esfuerzo})^{P2} \quad (6)$$

Donde:

- A, B son constantes multiplicativas de esfuerzo y P1, P2 exponentes, que se determinan en función del tipo de proyecto y del tamaño. Ver tabla 1.
- Cd_i: Manejadores de costos.
- Tamaño: líneas de código fuente, expresado en miles. Se obtiene multiplicando el total de Web Objects por una constante asociada al lenguaje de programación.

Tabla 1: Constantes de acuerdo al tipo de proyecto

Parámetros / Exponentes	A	B	P1	P2
Basado en el comercio electrónico	2,3	2	1,05	*
Aplicación financiera	2,7	2,2	1,05	*
Aplicación de B2B (negocio a negocio)	2	1,5	1	*
Aplicación basada en utilitarios informativos	2,1	2	1	*

(*) 0,5 para menos de 300 puntos de objeto o 0,32 para más 300.

Para calcular el producto de los manejadores de costos (cd_i) se seleccionan los valores (que se detallan en [10]), asociados a distintos aspectos vinculados con el proyecto:

- CPLX: Complejidad y confiabilidad del producto
- PDIF: Dificultad de la plataforma
- PERS: Capacidades del personal
- PREX: Experiencia del personal
- FCIL: Comodidades
- SCED: Restricciones de horarios
- RUSE: Grado de reutilización planeado
- TEAM: Trabajo en equipo
- PEFF: Eficiencia del proceso

Los Web Objects se calculan a partir de los siguiente predictores:

- ILF: Archivos lógicos internos.
- EIF: Archivos externos
- EI: Entradas externas
- EO: Salidas externas
- EQ: Consultas externas
- MmF: Archivos multimedia
- WBB: Constructor de bloques web
- Scripts: número de macros, contenedores, etc
- Links: número de referencias

asignándole a cada uno un peso (bajo, medio, alto) según corresponda.

2 Metodología

Para la realización del análisis comparativo de los métodos de estimación, se utilizó la información obtenida de los proyectos de desarrollo realizados por los alumnos de la asignatura Taller de Programación I, de la carrera Licenciatura en Sistemas de Información de la UNNE.

Esta asignatura, ubicada en el tercer año del plan de estudio, tiene como objetivo profundizar el estudio de herramientas de desarrollo de software orientadas a la plataforma web. Contribuye específicamente a la formación del Analista Programador Universitario, título intermedio de la carrera, cuyo perfil comprende el desarrollo, modificación y mantenimiento de aplicaciones informáticas, mediante la utilización de herramientas de desarrollo de uso generalizado en el mercado laboral.

Para cumplir este objetivo se plantea el desarrollo y seguimiento de una aplicación web sencilla pero completa, que incluya todos los componentes necesarios: modelado de la aplicación, diseño gráfico y de contenidos, gestor de base de datos, tecnologías de programación en cliente y en servidor.

Para este experimento los alumnos debían cumplir las siguientes consignas:

- Desarrollar una aplicación orientada al comercio electrónico, respetando los criterios vinculados con la presentación y las funcionalidades requeridas, que fueron previamente definidos por el profesor.
- Realizar la especificación de requerimientos utilizando la metodología NDT, que incluye el modelado con diagramas de casos de uso.
- Utilizar las herramientas: Eclipse, HTML, CSS, PHP5 y PhpMyAdmin.
- Realizar la presentación de los resultados del proyecto en 2 fases: la primera incluye la programación de los documentos HTML y de las hojas de estilo, y la segunda, la programación en PHP, diseño y conexión de la base de datos.
- Por cada fase, presentar una minuta, indicando la tarea realizada y la duración correspondiente.

Al finalizar el plazo previsto, se presentaron 13 proyectos. A partir de los datos obtenidos de la minuta entregada, se obtuvo un promedio de la duración real del desarrollo de los alumnos. Para la primera fase se estableció una duración promedio de 2 hs por día, durante 14 días. Y para la segunda fase una duración promedio de 4 hs por día, durante 30 días. Considerando la duración medida en meses de 20 días de 5 hs, se obtuvo una duración real de 1,48 meses.

Para definir el contexto, se supone que ningún estudiante tiene experiencia en el desarrollo de aplicaciones web, ni en las herramientas que se enseñan en el cursado de la asignatura.

Teniendo en cuenta las consignas dadas para el desarrollo de la aplicación, las características de las herramientas y el contexto específico, se realizó la estimación de la duración con los siguientes métodos: FP-Lite, Casos de Uso y Webmo, a fin de determinar qué método predice mejor la duración para este tipo de proyecto.

3 Resultados

A continuación se presentan los resultados de la aplicación de cada uno de los métodos. Cabe destacar que en todos los casos se consideró el mes de 20 días y 5 hs por día, por la naturaleza de un trabajo académico.

3.1 FP-LITE

Para la estimación este método considera el tipo de proyecto a desarrollar. En este caso, dado el contexto específico, los proyectos se consideraron nuevos, multiplataforma, utilizando lenguajes de 4ta. generación.

Para el conteo de puntos de función, se analizaron los elementos funcionales requeridos, determinando los siguientes:

- ILF (ficheros lógicos internos): usuarios, productos, ventas, consultas. Total: 4
- ELF (ficheros lógicos externos): no tiene. Total: 0
- EI (entradas externas): agregar productos, agregar usuarios, eliminar productos, eliminar usuarios, modificar usuarios, modificar productos, registrar facturas. Total= 7
- EQ (consultas externas): consulta de usuarios, consulta de servicios. Total= 2
- EO (salidas externas): no tiene.

La estimación que provee es puntual, pero se recomienda utilizar un intervalo en torno al 20% del valor obtenido. El resultado optimista es el representado por PF-1, el esperado o promedio es PF-2 y el pesimista PF-3.

Dado que el método considera 6 fases para el ciclo de vida (Planificación, especificación, diseño, construcción, pruebas e implantación), en función de las características de los proyectos considerados, se tomó en cuenta sólo la fase de *Construcción*, por tanto se calculó el esfuerzo de desarrollo como el 40% del Esfuerzo Tota, tal como se muestra en la tabla 2. La duración se ajustó a las condiciones establecidas (mes de 20 días y día de 5 hs), y se obtuvo el valor **de 2,6 meses**.

Tabla 2: Estimación utilizando FP-Lite

Elemento	Peso	Cantidad	Total (Cantidad x Peso)
ILF	10	4	40
ELF	7	0	0
EI	4	7	28
EQ	5	2	10
EO	4	0	0
	PF-1 (-20%)	PF-2	PF-3 (+20%)
Total FP sin ajustar	62,4	78	93,6
Esfuerzo total (h-h)	873,9	1021,4	1160,3
Duración total (meses 20d x 8h)	3,8	4,0	4,2
Esfuerzo de desarrollo	349,6	408,6	464,1
Duración (meses 20d x 8h)	1,5	1,6	1,7
Duración (meses 20d x 5h)	2,4	2,6	2,7

3.2 Puntos de caso de uso

Para la aplicación de este método se determinaron:

Actores: administrador y cliente. Tipo: complejo porque interactúan a través de una interfaz gráfica.

Casos de uso: alta, baja, modificación de productos; alta, baja, modificación de usuarios, consulta de usuarios, consulta de servicios, registrar consultas de los usuarios, registrar ventas. Tipo: simple; total=10

Factores técnicos: utilizando la fórmula (2), el valor final de TCF fue de 0,8. En la tabla 3 se tiene un detalle de la ponderación de cada uno de los factores.

Tabla 3: Ponderación de cada factor técnico de complejidad

FT	Descripción	Peso	Ponderación	Total
T1	Sistema distribuido	2	0	0
T2	Objetivos de rendimiento	1	2	2
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento complejo	1	0	0
T5	Reusabilidad	1	3	3
T6	Fácil de instalar	0,5	0	0
T7	Fácil de usar	0,5	3	1,5
T8	Portabilidad	2	3	6
T9	Fácil de cambiar	1	3	3
T10	Uso concurrente	1	0	0
T11	Características de seguridad	1	2	2
T12	Provee acceso a terceros	1	3	3
T13	Formación especial requerida	1	0	0

Se considera que los factores T1, T4, T6, T10 y T13, no son relevantes (ponderación 0) dado que la aplicación no se ha diseñado para que sea distribuida, no hará cálculos matemáticos complejos, el sistema no se instala, no tendrá acceso concurrente y los usuarios no serán expertos (esta orientados para el público en general que desee comprar un artículo o contratar un servicio). Los objetivos de rendimiento y de seguridad no son prioritarios, por tanto, la ponderación es 2. Los factores valorados con 3 tienen un impacto medio en el proyecto.

Factores de entorno: con la fórmula (3) se obtuvo el valor final de ECF de 0,5. En la tabla 4 se muestra la valoración que recibió cada uno de los factores:

Tabla 4: Valoración de los factores del entorno

FE	Descripción	Peso	Ponderación	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	3	1,5
E3	Experiencia en la orientación a objetos	1	4	4
E4	Capacidad del analista líder	0,5	4	2
E5	Motivación	1	5	5
E6	Estabilidad en los requerimientos	2	5	10
E7	Personal de medio tiempo	-1	0	0
E8	Dificultad en el lenguaje de programación	1	3	3

Los factores E1, E2 y E8 se han considerado con un valor promedio debido a la experiencia previa que adquirieron los alumnos en el transcurso de la carrera. E3 tiene una ponderación relevante porque los alumnos previamente cursaron la asignatura Programación Orientada a Objetos y E4 porque el analista líder es el propio profesor. E5 tiene la más alta ponderación debido a que el proyecto propone libertad para aplicar las soluciones, representa un desafío e implica la aprobación de la materia. Igual que E6 que justifica su ponderación porque no habrá cambios en los requerimientos planteados al inicio. E7 es 0 porque el proyecto no cuenta con personal de medio tiempo.

El cálculo final se presenta en la tabla 5. Se ha tomado un factor de productividad (FP) de 17 al ser casos de usos simples.

Tabla 5: Estimación utilizando Casos de Uso

$UUCP = \text{Actores} * \text{Peso} + \text{Casos de uso} * \text{Peso} = 2 * 3 + 10 * 5 = 56$
$UCP = TFC * ECF * UUCP = 0,8 * 0,5 * 56 = 23,4$
Esfuerzo = $UCP * FP = 23,4 * 17 = 397,5$ hh
Duración = $\text{Esfuerzo} / \text{horas por día} / \text{días por meses} = 397,5 / 5 / 20 = \mathbf{3,97 \text{ meses}}$

3.3 WebMO

Para este método de estimación se ha determinado que los proyectos son de tipo comercio electrónico, por lo que los valores de las constantes de esfuerzo proporcionadas por WebMo son: A=2,3; B=2; P1=1,03 y P2=0,5.

Como los alumnos programaron en diversos lenguajes (PHP, HTML, CSS, SQL) se consideró el default para los lenguajes de 4GL = 20 para la constante de lenguaje.

Para establecer el valor de los multiplicadores de esfuerzo, se tomaron los valores establecidos en la tabla Cost Driver que define el método [9]. En la tabla 6 se puede observar el valor seleccionado en cada caso:

Tabla 6: Selección de multiplicadores de esfuerzo

Cost Driver	Ratings				
	Verylow	Low	Nominal	High	Veryhigh
CPLX	0,63	0,85	1	1,3	1,67
PDIF	0,75	0,87	1	1,21	1,41
PERS	1,55	1,35	1	0,75	0,58
PREX	1,35	1,19	1	0,87	0,71
FCIL	1,35	1,13	1	0,85	0,68
SCED	1,35	1,15	1	1,05	1,1
RUSE	1,35	1,15	1	1,25	1,48
TEAM	1,45	1,31	1	0,75	0,62
PEFF	1,35	1,2	1	0,85	0,65

El recuento de objetos web (WO) se calculó seleccionando los valores que corresponden a cada aspecto del proyecto. Los valores resultantes se muestran en la tabla 7.

Tabla 7: Recuento de objetos web

Web Objects Predictors	Cantidad x peso	Total
ILF	4x7	28
EI	7x3	21
EQ	2x3	6
MMF	4x4	16
WBB	3x3	9
QL	4x3	12
Total	92	92

$$\text{Tamaño} = \text{WO} * \text{constante del lenguaje} / 1000$$

$$\text{Tamaño} = 92 * 20 / 1000 = 1,84 \text{ (Miles de líneas de código)}$$

En la tabla 8 se muestran los valores resultantes de la estimación de esfuerzo y duración.

Tabla 8: Estimación con Webmo

Esfuerzo: 2,0 personas-mes
Duración total: 2,8 meses (20 días, 8hs)
Duración del desarrollo: 1,12 (40% correspondiente al desarrollo)
Duración del desarrollo ajustado: 1,79 meses (20 días, 5 hs)

La duración de los proyectos de los alumnos se compara con la duración estimada proporcionada por cada uno de los métodos utilizados en la tabla 9.

Tabla 9: Duraciones obtenidas vs duración real

Promedio duración real de los proyectos de alumnos	FP-Lite	Casos de uso	Webmo
1,48	2,6	3,97	1,79

La duración calculada por Webmo es la que más se aproxima a la duración real y la más alejada es la duración proporcionada por el método de Puntos de Casos de Usos. Las estimaciones obtenidas son coherentes con las características de los métodos. Los puntos de casos de uso son un método de estimación temprana con lo cual no incorpora datos que puedan aportar mayor precisión. El método FP-Lite utiliza principalmente las funcionalidades básicas del sistema aportando parámetros en función de las características técnicas del desarrollo y Webmo aporta información específica de proyectos web.

4 Conclusiones

Se realizó un análisis comparativo de métodos de estimación basados en puntos de función, aplicados a proyectos de desarrollo de una asignatura de grado cuyo objetivo es la introducción a la programación web.

Se seleccionaron los métodos FP-Lite, Método basado en Casos de Uso y el método Webmo, por ser específico para aplicaciones web.

La duración calculada por Webmo es la que más se aproxima a la duración real y la más alejada es la duración proporcionada por el método de Puntos de Casos de Usos.

Se concluye que cuando el método posee factores que proveen mayor información sobre el tipo de desarrollo, en este caso web, proveen una mejor estimación.

Como trabajo futuro se propone aplicar los mismos métodos en un proyecto de desarrollo real, a nivel de empresas de software, a fin de validar los resultados obtenidos y verificar la adecuación de los factores que proveen los métodos en este tipo de proyectos.

Referencias

1. McConnell, S.: Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft. (2006)
2. Moløkken, K., Jørgensen, M., Tanilkan, S.S., H, Gallis, Lien, A.C. and Hove, S.E.: A survey on software estimation in the Norwegian industry. In Proceedings of the 10th International Symposium on Software Metrics. pp. 208–219. (2004)
3. Nasir M., Ahmad F.: An Empirical Study to Investigate Software Estimation Trend in Organizations Targeting CMMI. Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. (2006)
4. Dapozo, Greiner, Medina, Ferraro, Pedrozo Petrazzini, Lencina.: Métodos de estimación de software. Un análisis desde un enfoque evolutivo. III Jornadas de Investigación en Ingeniería del NEA y países limítrofes. UTN - Facultad Regional Resistencia. ISBN: 978-950-42-0157-1. 9 y 10 de Junio de 2014. (2014)
5. Albrecht A.: Measuring Application Development Productivity. Proc of IBM applications. (1979)
6. Piattini, et al.: Medición y Estimación del Software. AlfaOmega Editores. (2008)
7. C. A. Remón, P. Thomas.: Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso. XVI Congreso Argentino de Ciencias de la Computación. (2010)
8. M. C. García Y J. Garzás.: Método de Estimación de Puntos de Caso de Uso. Disponible en: <http://www.kybeleconsulting.com/articulos/estimacion-puntos-caso-de-uso/> ed., KybeleConsulting.
9. Reifer, Donald J.: Web Development: Estimating Quick-to-Market Software. Software, IEEE Computer Society, November/December 2000, pp. 57-64. (2000)
10. Reifer, Donald J.: Estimating Web Development Costs: There Are Differences. CROSSTALK The Journal of Defense Software Engineering. (2002)