

# Métodos de estimación de esfuerzo y duración en proyectos web pequeños

Gladys N. Dapozo, Yanina Medina, Andrea B. Lencina, Gabriel O. Pedrozo  
Petrazzini

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura

Universidad Nacional del Nordeste, Av. Libertad 5450, 3400, Corrientes,  
Corrientes (CR), Argentina

{gndapozo, yanina}@exa.unne.edu.ar

**Abstract.** Para las empresas de software una estimación deficiente del esfuerzo y duración que conlleva un proyecto puede ocasionar incumplimiento de plazos, entrega de productos incompletos y pérdida de competitividad. Para evaluar métodos que mejor se adecuen a contextos de desarrollo de proyectos de pequeña o mediana envergadura, en este trabajo se muestra el resultado obtenido con diferentes métodos de estimación aplicados a proyectos de alumnos de una asignatura orientada a la programación Web. Resulta que a mayor especificidad en cuanto al tipo de proyecto y una mayor disponibilidad y pertinencia de datos históricos, los valores de esfuerzo y duración estimados se aproximan más a los valores reales.

**Resumo.** Para as empresas de software uma estimação deficiente do esforço e duração que leva um projeto pode ocasionar descumprimento de prazos, entrega de produtos incompletos e perda de competitividade. Para avaliar métodos de adequação melhor a contextos de desenvolvimento de projetos de pequena ou mediana magnitude, neste projeto se apresenta o resultado da aplicação de diferentes métodos de estimação aplicados a projetos de alunos de uma cadeira orientada à programação web. Resulta que a maior especificidade em quanto ao tipo de projeto e uma maior disponibilidade e relevância de dados históricos, os valores de esforço e duração estimados se aproximam mais aos valores reais.

## 1. Introducción

Según [McConnell 2006] “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 el proceso de desarrollo, 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 consolidar una base histórica de estimación, con el propósito de

disminuir cada vez más la desviación entre los valores estimados y reales [Moløkken et al 2004].

Existen diversos métodos para estimar el esfuerzo de desarrollo, 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 [Nasir and Ahmad 2006]. En un trabajo previo de los autores se elaboró un cuadro para resumir los principales métodos reportados en la literatura [Dapozo et al 2014].

## **1.2. 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 por Albrecht en el año 1979 [Albrecht 1979]. Para obtener los puntos de función utiliza 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.

A continuación se describen algunos métodos de estimación seleccionados en función de sus requerimientos de información, que los hacen adecuados para contextos de desarrollo de proyectos de pequeña o mediana envergadura.

## **1.3. 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 requerido, en horas-hombre (hh), para el desarrollo de un proyecto de software [Remón and Thomas2010].

El método utiliza cuatro variables principales:

- Actores (Ac): se evalúan sus características intrínsecas y se determina la forma en que interactúan con el sistema a desarrollar, asignándole un peso de acuerdo a la complejidad de la interacción: los simples 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 ponderación de 0 a 5, siendo 0-irrelevante, 1-fuerte impacto negativo, 3-promedio y 5-fuerte impacto positivo.

Los pasos a seguir para obtener la estimación son los siguientes [García and Garzás 2010]:

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

$$UUCP = \sum_{i=1}^n AC_i P_i + \sum_{i=1}^m CU_i P_i \quad (1)$$

Donde n es el n° de actores, m el n° de CU y P el peso

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

$$TCF = 0,6 + 0,01 \sum_{i=1}^{13} TF_i P_{o_i} \quad (2)$$

, TF<sub>i</sub> es el peso del factor i y P<sub>o</sub> la ponderación

$$ECF = 1,4 - 0,03 \sum_{i=1}^8 EF_i P_{o_i} \quad (3)$$

, EF<sub>i</sub> es el peso del factor i y P<sub>o</sub> la ponderación

- 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.

$$\text{Esfuerzo} = UCP * FP, \text{ donde FP es el factor de productividad} \quad (5)$$

$$\text{Duración} = \text{Esfuerzo} / \text{Horas por día}$$

#### 1.4. Webmo

Este método, propuesto por [Reifer 2000] y basado en COCOMO II, intenta dar respuesta a las diferencias entre las estimaciones de desarrollos clásicos y de proyectos Web. Utiliza como unidad de medida de tamaño el “Web Object”, similar a los puntos de función, pero abarca además funciones específicas de las aplicaciones Web, tales como: Links, Multimedia, Scripts, etc. Para calcular el esfuerzo y la duración, se utilizan las siguientes fórmulas:

$$\text{Esfuerzo} = A (\prod cd_i) (\text{tamaño})^{P1} \quad (6)$$

$$\text{Duración} = B (\text{esfuerzo})^{P2} \quad (7)$$

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<sub>i</sub>: 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<sub>i</sub>), se seleccionan los valores que se detallan en [Reifer 2002], 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

En cada caso, se asigna un peso (bajo, medio, alto) según corresponda.

## 1.5. CWADEE

El CWADEE (Chilean Web Application Development Effort Estimation) es un método para la estimación de esfuerzo en proyectos de software web, enfocado en apoyar al estimador experto para obtener una estimación más precisa en escenarios inmaduros [Ochoa2007]. El método está diseñado para ser utilizado con poca información histórica de la empresa, para proyectos Web pequeños y medianos, el tipo de proyecto usual en empresas tipo pyme.

Introduce una métrica denominada Data Web Points (DWP) que mide el tamaño de forma indirecta. Para ello, parte del modelo de datos del sistema a desarrollar que se obtiene de las especificaciones del cliente en las primeras entrevistas.

El método realiza una clasificación de las entidades y relaciones del modelo de datos asignándoles un tipo de patrón. Los patrones para las entidades son: Regular, Dependiente y Relación, considerando relaciones 1-N y 1-1. Luego, realiza un conteo de la cantidad de entidades y relaciones para cada tipo de patrón y le asigna un peso determinado por el estimador experto. Finalmente, se suman la cantidad de patrones por su peso para obtener los DWP.

Para calcular el esfuerzo, hace uso de la fórmula:

$$E = UC \prod_i CD_i (DWP \cdot (1 + X^*))^P \quad (8)$$

Donde:

- E es el esfuerzo en horas-hombre
- UC es el costo de usuario
- CD es el producto de los directores de costo
- DWP es la cantidad de Data Web Points que representan el tamaño de la aplicación
- X\* es el coeficiente de representatividad de los DWP
- P es una constante

El valor de la constante P se define como 1,05 para proyectos con DWP < 300 y de 1,02 para proyectos con DWP > 300.

El costo de usuario (UC) es una función de los tipos de usuario que interactúan con el sistema. El método define tres usuarios por defecto: el Administrador, el Actualizador y el Consultor, pero es posible definir otros en función del sistema. Para cada usuario se determina el porcentaje de uso de funcionalidades del sistema o fracción de ámbito (I) y el grado de reutilización de las funcionalidades del sistema (R), que puede venir de la implementación de funcionalidad para otros usuarios del sistema, o de otros sistemas. Teniendo esto en cuenta, se utiliza la siguiente fórmula para determinar el costo de usuario:

$$UC = \sum_i I_i (1 - R_i) \quad (9)$$

Los directores de costo son tomados del método WebMo, excepto uno, denominado CLIEN, que se refiere al tipo de Cliente (Conocimiento tecnológico que tiene el cliente). Los valores fueron ajustados luego del estudio de 22 proyectos al momento de definir CWADEE, por lo que presentan valores diferentes de los de WebMo original.

Por último, el coeficiente de representatividad de los DWP indica la cantidad de esfuerzo adicional que se requiere para implementar funcionalidades que no se pueden deducir a partir del modelo de datos, tales como las funcionalidades adicionales, imágenes, navegación, etc. Para determinar su valor se debe recurrir a la información histórica de la empresa sobre proyectos similares, específicamente el esfuerzo medido en horas-hombre. Los pasos son:

En primer lugar, se determina el valor real de DWP (DPW\*) de un proyecto despejando de la fórmula de esfuerzo:

$$DWP^* = \sqrt[P]{\frac{E}{UC * CD}}$$

(10)

Luego se obtiene X, el porcentaje del tamaño del sistema que no puede inferirse del modelo de datos:

$$X = \frac{DWP}{DWP^*}$$

(11)

Si se tienen más proyectos similares, se realiza un promedio de todos ellos para obtener X\*, de lo contrario, se considera X\*=X.

## 1. 6. RESC

El método RESC (Raw Estimation based on Standard Components) está pensado para estimar proyectos Web utilizando datos históricos. Explora las similitudes en cuanto a funcionalidad que tienen los componentes presentes en las páginas Web considerando el tiempo, esfuerzo y el costo invertido por la persona o equipo que lo elaboró [Ochoa et al 2007].

El método toma un proyecto similar del que se tiene registro de esfuerzo, tiempo y costo invertidos, además de información pertinente a los siguientes componentes: *Mantenimiento de datos, Menú/Navegación, Consultas, Informes por impresora/pantalla, Procesos en background, Autenticación de datos, Tablas, Notificaciones*. A partir de estos datos se determina la cantidad total de los puntos de función del proyecto anterior (PF'), y con ellos se calcula por PF la dedicación (DH) y el costo (CPH)

$$DH = \frac{\text{Esfuerzo}}{PF'}$$

(12)

$$CPH = \text{CostoTotal} / PF'$$

Tomando estos valores como datos históricos, al momento de estimar el proyecto nuevo, se estima a través del juicio de experto la cantidad de componentes con los que contará el producto, indicando para cada tipo de componente la cantidad máxima, la más probable y la mínima. Seguidamente se ajustan considerando su ponderación:

$$\text{Estimación Ajustada} = (4 * \text{Est.} + \text{Máx.} + \text{Min.}) / 6$$

Donde: Est.: cantidad más probable; Máx: cantidad máxima y Min: cantidad mínima.

Una vez obtenida la estimación ajustada se calcula la cantidad total de puntos de función (TPF) que tendrá el proyecto:

$$TPF = \sum_{i=1}^n \text{QUOTE } PF = i = 1nEiCi, \text{ donde } Ei \text{ es cada elemento funcional y } Ci \text{ la complejidad de cada}$$

(13)

Luego, se calcula el tiempo, esfuerzo y costo que demandará este proyecto:

$$\text{Esfuerzo} = TPF * DH$$

$$\text{Costo} = TPF * CPH$$

Tiempo (en meses) = Esfuerzo /HD /DM

Donde:

- HD: Horas al día dedicadas al proyecto.
- DM: Días al mes dedicado al proyecto.

## 2 Metodología

Para evaluar el comportamiento de los métodos descriptos, se consideró la información del proceso de desarrollo, estipulado como trabajo práctico obligatorio realizado 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 a la base de datos.
- Por cada fase, se presenta una minuta (reporte breve), 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: Puntos de Casos de Uso, CWADEE, Webmo y RESC, a fin de determinar qué método predice mejor la duración para este tipo de proyecto, en función de la información que cada uno requiere.

### 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 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	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	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**

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

### 3.2 CWADEE

En primer lugar se realizó el cálculo de los valores históricos. Para ello se consideró un proyecto similar desarrollado en la misma asignatura en el año anterior. Los datos obtenidos fueron:

- Esfuerzo total de 140 hh.
- Un costo de usuario (UC) de 0,96 obtenido con la fórmula 9 y los valores de la tabla 6

**Tabla 6: Valores de Fracción de Ámbito y Grado de Reuso del proyecto anterior**

Tipo	Fracción de Ámbito (I)	Grado de Reuso (R)
Administrador	0,7	0,2
Consultor	0,5	0,5
Visitante	0,3	0,5

El valor de los CD fue de 0,63. Los valores seleccionados se resaltan en la tabla 7.

**Tabla 7: Manejadores de costos para el proyecto nuevo**

Comp.	Verylow	Low	Nominal	High	Veryhigh
COPLX	0,65	0,85	1	1,35	1,6
DIFPLA	0,9	1	1,1	1,3	1,65
PERS	1,6	1,3	1,05	0,9	0,8
EXPER	1,3	1,15	1,05	0,9	0,8
INFRA	1,35	1,15	1	0,9	0,85
SCHED	1,4	1,2	1	0,95	0,9
CLIEN	1,55	1,35	1,1	0,9	0,75
WTEAM	1,5	1,3	1,05	0,9	0,8
EFPRO	1,3	1,1	1	0,85	0,65

El valor de los DWP\* se obtuvo con los valores de UC = 0.96 y CD=0.63, y aplicando la fórmula 10 se llegó a DWP\*=178.

Para el nuevo proyecto, se identificaron 3 entidades regulares (tipo de usuario, categorías de productos y consultas), 4 entidades dependientes (usuarios, productos, ventas, detalle de venta), y 3 relaciones 1-N (tipo de usuario – usuario, categorías de productos – productos, ventas – detalles de ventas).

De la clasificación en patrones de las entidades y relaciones del modelo de datos, se obtuvo el conteo de DWP, presentados en la tabla 8:

**Tabla 8: Definición de las cantidades de DWP**

Tipo	Cantid	Peso	Tot
Regulares	3	4	12
Dependien	4	12	48
1-N	3	5	15
Total			75

Dado que los DWP no superan las 300 unidades, el valor de la constante P que interviene en la fórmula del esfuerzo es de 1,05.

El coeficiente X\* se calculó a partir de los DWP del proyecto actual y de los DWP\* obtenido del dato histórico y aplicando la fórmula 11, siendo su valor de 0,42.

Los directores de costo seleccionados se resaltan en la tabla 9.

**Tabla 9: Manejadores de costos para el proyecto nuevo**

Comp.	Verylow	Lo	Nominal	High	Veryhigh
COPLX	0,65	0,85	1	1,35	1,6
DIFPLA	0,9	1	1,1	1,3	1,65
PERS	1,6	1,3	1,05	0,9	0,8
EXPER	1,3	1,15	1,05	0,9	0,8
INFRA	1,35	1,15	1	0,9	0,85
SCHED	1,4	1,2	1	0,95	0,9
CLIEN	1,55	1,35	1,1	0,9	0,75

WTEAM	1,5	1,3	1,05	0,9	0,8
EFPRO	1,3	1,1	1	0,85	0,65

Así, el valor calculado de CD fue de 0,87.

Por último, en la tabla 10 se muestra la clasificación de los usuarios con sus correspondientes porcentajes de porcentaje de uso (I) y grado de reuso (R):

**Tabla 10: Valores de Fracción de Ámbito y Grado de Reuso**

Tipo	Fracción de Ámbito (I)	Grado de Reuso (R)
Administrador	0,5	0,1
Consultor	0,5	0,1
Visitante	0,3	0,1

Entonces, el valor de UC utilizando la fórmula 9 fue de 1,17.

Con todos los datos se puede calcular el esfuerzo con el uso de la fórmula 8:

$$E = 1,17 * 0,87 + (1 + 0,42)^{1,05} = 136,90 \text{ hh}$$

Ajustando el resultado dado en hh a meses de 20 días, 5 horas cada uno, el total en meses es de:

$$136,90 / 5 / 20 = 1,37 \text{ meses}$$

### 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 [Reifer 2000]. En la tabla 11 se puede observar el valor seleccionado en cada caso:

**Tabla 11: 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, como se muestra en la tabla 12.

**Tabla 12: Recuento de objetos web**

Web ObjectsPredictors	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

- Tamaño = WO \* constante del lenguaje / 1000
- Tamaño = 92 \* 20 / 1000 = 1,84 (Miles de líneas de código)

A continuación se muestran los valores resultantes de la estimación de esfuerzo y duración.

- 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)

### 3.4 RESC

Para la estimación con este método, en primer lugar se calcularon los valores históricos considerando el proyecto anterior de la asignatura, tomado como referencia.

Se calculó el tamaño del proyecto histórico medido en puntos de función (ver tabla 13).

**Tabla 13: Peso estimado de componentes estándares (Histórico)**

COMPONENTES	Chico		Mediano		Grande		Total (PF)
	Peso	Cant	Peso	Cant	Peso	Cant	
Mantenedor de datos	2	0	4	1	6	0	4
Menú/Navegación	4	1	7	1	10	0	11
Consultas	2	0	4	3	6	0	12
Informes por	2	0	6	3	10	1	28
Autenticación de usuarios	2	0	2	2	2	0	4
Capa de acceso a datos	6	1	10	0	14	0	6
Tablas	1	4	1,5	2	2	0	7
Notificaciones	2	1	2,5	1	3	0	4,5
Total							76,5

El tiempo que demoró la implementación del proyecto histórico fue de 20 días de 7 horas. El esfuerzo fue de 140 hh y la dedicación (DH) fue de 1,83 horas por Punto de Función aplicando la fórmula 12.

La estimación del proyecto nuevo se realizó a partir de las consignas de desarrollo dadas a los alumnos en el año 2014. En la tabla 14 se muestran las determinaciones de las cantidades de componentes estándares (mínimas, más probable y

máximas), y la cantidad total de puntos de función (TPF) que se obtuvo aplicando la fórmula 13.

**Tabla 14: Estimación de componentes estándares**

COMPONENTE	Peso (PF)	Min	Max	Estimado	Ajustado	Total
Mantenedor de datos - chico	2	-	-	-	-	-
Mantenedor de datos - mediano	4	1	3	1	1,34	5,36
Mantenedor de datos - grande	6	-	-	-	-	-
Menú/Navegación - chico	4	1	2	1	1,17	4,68
Menú/Navegación - mediano	7	1	3	1	1,34	9,38
Menú/Navegación - grande	10	0	1	0	0,17	1,7
Consultas - chico	2	1	3	2	2	4
Consultas - mediano	4	1	2	2	1,84	7,36
Consultas - grande	6	0	1	0	0,17	1,02
Informes por impresora/pantalla -chico	2	-	-	-	-	-
Informes por impresora/pantalla -	6	1	4	2	2,17	13,02
Informes por impresora/pantalla -grande	10	-	-	-	-	-
Autenticación de usuarios - chico	2	0	1	1	0,84	1,68
Autenticación de usuarios - mediano	2	1	2	1	1,17	2,34
Autenticación de usuarios - grande	2	-	-	-	-	-
Capa de acceso a datos - chico	6	0	0	0	0	0
Capa de acceso a datos - mediano	10	1	2	1	1,17	11,7
Capa de acceso a datos - grande	14	-	-	-	-	-
Tablas - chico	1	3	5	4	4	4
Tablas - mediano	1,5	3	6	4	4,17	6,255
Tablas - grande	2	1	2	1	1,17	2,34
Notificaciones - chico	2	1	3	2	2	4
Notificaciones - mediano	2,5	1	2	1	1,17	2,925
Notificaciones - grande	3	-	-	-	-	-
					Total:	81,76

Con los valores de TPF y DH, se calculó:

$$\text{Esfuerzo} = \text{TPF} * \text{DH} = 81,76 \text{ TPF} * 1,83 \text{ hh/PF} = 149,62 \text{ hh}$$

Duración:  $\text{Esfuerzo} / \text{HD} / \text{DM} = 149,62 / 5 / 20 = 1,49$  meses (meses de 20 días de 5 hs. cada uno).

### 3.5 Duración en meses de cada método

La duración de los proyectos de los alumnos se compara con la duración estimada proporcionada por cada uno de los métodos utilizados, que se muestran en la tabla 15.

**Tabla 15: Duraciones obtenidas vs duración real**

Promedio duración real de los proyectos de alumnos	CWADEE	Casos de uso	Webmo	RESC
1,48	1,36	3,97	1,79	1,49

La duración calculada por RESC 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.

#### **4. Conclusiones**

Se realizó un análisis comparativo de métodos de estimación aplicados a proyectos de desarrollo de una asignatura de grado cuyo objetivo es la introducción a la programación Web.

Se seleccionaron los métodos Basado en Puntos de Casos de Uso, Webmo, CWADEE y RESC. La duración calculada por RESC 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.

RESC presenta como ventaja que puede aplicarse en una etapa temprana dado que requiere información posible de extraer de proyectos anteriores. El método, si bien se basa fuertemente en el juicio de experto, permite sistematizar un procedimiento que ofrece un valor estimado con bastante precisión.

CWADEE se destaca por basar su información no en las funcionalidades sino en el modelo de datos del proyecto a estimar, por tanto también es un método de estimación temprano.

Se concluye que a mayor especificidad del método mejor es el ajuste a la duración real. En este caso los métodos CWADEE, Webmo y RESC, están enfocados en proyectos Web, en tanto el método de Puntos de Casos de Uso, es más general.

Dentro de los métodos específicos, a mayor disponibilidad y pertinencia de datos históricos de proyectos, se aproxima más a la duración real.

Como trabajo futuro se propone aplicar los mismos métodos en proyectos realizados en el ámbito de una empresa de software o área de sistema de otras organizaciones, a fin de validar los resultados obtenidos y verificar la adecuación de los factores que proveen los métodos en este tipo de proyectos.

#### **5. References**

- Albrecht A. (1979) "Measuring Application Development Productivity". In: Proc of IBM applications.
- Dapozo G., Greiner C., Medina Y., Ferraro M., PedrozoPetrazzini, O. and Lencina A. (2014) "Métodos de estimación de software. Un análisis desde un enfoque evolutivo". In: III Jornadas de Investigación en Ingeniería del NEA y países limítrofes. UTN - Facultad Regional Resistencia. ISBN: 978-950-42-0157-1.
- García, M. C. and Garzás, J. (2010). "Método de Estimación de Puntos de Caso de Uso". In: <http://www.233gradosdeti.com/estimacion-puntos-caso-de-uso/>, acceso: 04/ 2014.
- McConnell, S. (2006) "Software Estimation: Demystifying the Black Art" (Developer Best Practices). Microsoft.

- Moløkken, K., Jørgensen, M., Tanilkan, S.S., H, Gallis, Lien, A.C. and Hove, S.E. (2004) "A survey on software estimation in the Norwegian industry". In: Proceedings of the 10th International Symposium on Software Metrics. pp. 208–219.
- Nasir M., Ahmad F. (2006) "An Empirical Study to Investigate Software Estimation Trend in Organizations Targeting CMMI". In: 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.
- Ochoa, S. F., Pino, J. A., Andrade, D. (2007). "Estrategias para Estimar el Esfuerzo de Desarrollo de Proyectos Web en Escenarios Inmaduros". In: Revista de la Pontificia Universidad Católica del Ecuador No 81. P. 125-172
- Piattini, et al. (2008) "Medición y Estimación del Software". AlfaOmega Editores.
- Reifer, D. J. (2000) "Web Development: Estimating Quick-to-Market Software." In: Software, IEEE Computer Society, November/December, pp. 57-64.
- Reifer, D. J. (2002) "Estimating Web Development Costs: There Are Differences". In: CROSSTALK The Journal of Defense Software Engineering.
- Remón, C. A. and Thomas, P. (2010) "Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso". In: XVI Congreso Argentino de Ciencias de la Computación.