

Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso

Cristian A. Remón¹, Pablo Thomas²

¹ Dpto. I+D Maker Electrónica, Mar del Plata, Argentina
cremon@makerelectronica.com.ar

² III-LIDI, Instituto de Investigación en Informática
Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina
pthomas@lidi.info.unlp.edu.ar

Resumen. Las diversas metodologías de estimación de esfuerzo en el proceso de desarrollo de software han surgido a raíz de fracasos y aciertos en los proyectos. La evolución de los métodos de estimación no ha sido como consecuencia de su invalidez o resultados erróneos, sino de la misma evolución de la tecnología, del acotamiento de los márgenes de error y de las exigencias del mercado en busca de productos de mayor calidad. Este trabajo muestra que la metodología de Puntos de Casos de Uso propuesta por Frohnhoff y Engels, con claras y fehacientes muestras de éxito en momentos de su desarrollo, necesita de ajustes y/o mejoras. En este sentido, se presentan casos de estudio concretos.

Palabras Claves: Casos de Uso, Puntos de Caso de Uso, Estimación de esfuerzo.

1 Introducción

El uso software en la sociedad se ha incrementado notablemente, por lo que es necesario el desarrollo de software de calidad.

La calidad del software es medida, primordialmente, por la completitud de los requerimientos definidos en etapas iniciales del ciclo de vida de desarrollo y por el proceso aplicado durante ese ciclo de vida.

Para lograr este objetivo, todo el proceso de desarrollo de software debe ser administrado bajo un efectivo plan de proyecto [1], el cual garantiza que se concluye en costo y tiempo de acuerdo a los requerimientos definidos previamente.

El desarrollo de software es una actividad económica, y por lo tanto está sujeto a restricciones económicas además de las inherentemente técnicas, las cuales se encuentran explícitamente dentro del plan de proyecto [2].

Las actividades o tareas que se deben ejecutar para lograr el producto estipulado, requiere de la inversión de esfuerzo, el cual se estima en función de los requerimientos obtenidos en la etapa de elicitación. Los elementos involucrados en la estimación de proyectos de software son: tamaño, esfuerzo invertido, tiempo de desarrollo, tecnología utilizada, entre otros. El esfuerzo invertido es un elemento

fundamental, ya que a partir de este valor, como parte del costo del proyecto, se deriva el margen de ganancia que se obtiene por el producto terminado.

Uno de los métodos utilizados para estimar el esfuerzo de desarrollo de software se basa en modelos de Casos de Uso [3], técnica ampliamente difundida para describir la interacción de los usuarios con un sistema de software. Los diseñadores de UML (Unified Modeling Language) recomiendan a los desarrolladores seguir un proceso de modelado de Casos de Uso a ser utilizado como documento de entrada para el diseño y la etapa de testeo [4]. Gustav Karner [6] tomó el modelo de Casos de Uso para mejorar la técnica de Puntos de Función, en la estimación de esfuerzo en proyectos de software.

Las metodologías de estimación de esfuerzos radican en dos objetivos claros: estimar el esfuerzo en una etapa preliminar de desarrollo y reducir relativamente la magnitud del error.

En este trabajo se propone relevar 12 casos de estudio: 9 tesinas de grado de proyectos de software presentados por los aspirantes a ingenieros en informática de la Facultad de Ingeniería de la Universidad FASTA de la ciudad de Mar del Plata, y 3 proyectos de la industria de software. A partir de estos casos, se analizan los resultados obtenidos aplicando la metodología mejorada de Puntos de Caso de Uso propuesta por Frohnhoff y Engels [5], en comparación con los resultados obtenidos en la estimación realizada en cada uno de los casos de estudio.

En el punto 2 se describe la metodología propuesta por Gustav Karner para la estimación de esfuerzo. En el punto 3 se detallan los ajustes realizados por Frohnhoff y Engels al método de Gustav Karner. Luego en el punto 4, se realiza una descripción y los valores obtenidos de la aplicación del método de Frohnhoff y Engels, a los casos de estudios aquí propuestos. En el punto 5 se realiza un análisis y simulación del método de Frohnhoff y Engels, utilizando los valores medios obtenidos en el punto 4, con un enfoque principal de la variable A-factor del método mencionado. Finalmente en el punto 6 se presentan las conclusiones y trabajos futuros.

2 Metodología de Puntos de Casos de Uso (UCP) de Gustav Karner

La metodología de los puntos de casos de uso es una derivación de la metodología de puntos de función propuesta por Albrecht. Karner [6] basa su metodología en la utilización de casos de uso como dato de entrada para calcular el esfuerzo en horas-hombre (hh) que son necesarias para el desarrollo de un proyecto de software. El método de estimación del esfuerzo utiliza cuatro variables principales:

2.1 Clasificación de los Actores Involucrados

Los actores involucrados en los casos de uso se clasifican de acuerdo a su característica intrínseca y la forma en que interactúan con el sistema. Un actor *simple* (*peso=1*) es aquel que representa una interfaz de programación o API (ej.: capa de abstracción); un actor *medio* (*peso=2*) es aquel que interactúa mediante un protocolo

(ej.: TCP/IP, HTTP, FTP) y un actor *complejo* (*peso=3*) es aquel que interactúa por medio de una interfaz gráfica. A cada actor de acuerdo a esta clasificación le corresponde un valor el cual se denomina *peso* (El autor de la metodología no especifica el origen de los pesos en ninguno de los casos).

2.2 Clasificación de Caso de Uso

Los casos de uso son clasificados de acuerdo a la cantidad de transacciones que poseen, incluyendo las transacciones de escenarios alternativos y excluyendo las extensiones o inclusiones de otros casos de uso. Un caso de uso *simple* (*peso=5*) es aquel que posee 3 o menos transacciones; uno *medio* (*peso=10*) es el que posee de 4 a 7 transacciones; y un caso de uso *complejo* (*peso=15*) es el que posee más de 7 transacciones. Nuevamente, a cada caso de uso le corresponde un *peso*.

2.3 Factor de Complejidad Técnica del Proyecto de Software

Los factores técnicos (T) están definidos por las influencias técnicas que puedan afectar el proceso de desarrollo del sistema a construir. Cada factor técnico posee un grado de complejidad, que oscila entre 0 y 5, donde 0 significa un valor irrelevante o nulo y 5 determina un valor con alto grado de influencia. Cada factor técnico posee un valor de *peso*. El peso total de ese factor de influencia técnica se obtiene con el producto entre el valor de complejidad asignado y el peso que le corresponde al factor.

2.4 Factores de Entorno del Proyecto

Los factores de entorno (E) indican la influencia del grupo humano involucrado en el proyecto sobre el sistema a desarrollar. De manera similar a los factores técnicos, los factores de entorno poseen un grado de influencia que oscila entre 0 y 5, donde 0 significa un valor irrelevante o nulo y 5 determina un valor con alto grado de influencia. Cada factor de entorno posee un valor de *peso*. El peso total de ese factor de influencia técnica se obtiene con el producto entre el valor de influencia asignado y el peso que le corresponde al factor de entorno.

Para determinar la estimación de los Puntos de Caso de Uso se deben cumplir los siguientes pasos:

1. Clasificar los actores para determinar el valor de UAW (Unadjusted Actor Weight)
 $UAW = \text{Sumatoria de todos los pesos de los actores identificados}$
2. Clasificar los casos de uso para determinar el valor de UUCW (Unadjusted Use Case Weight)
 $UUCW = \text{Sumatoria de los pesos de los casos de uso}$
3. Determinar el valor del UUCP (Unadjusted Use Case Point)
 $UUCP = UAW + UUCW$

4. Determinar el valor de TCF (Technical Complexity Factor)

$$TCF = 0,6 + (0,01 * \Sigma (T_1...T_{13}))$$

5. Determinar el valor de EF (Environment Factor)

$$EF = 1,4 + (-0,03 * \Sigma (E_1...E_8))$$

6. Determinar el valor de AUCP (Adjusted Use Case Point)

$$AUCP = UUCP * TFC * EF$$

Karner establece un factor de 20hs hombre por punto de caso de uso para realizar la estimación de un proyecto de software [6].

$$UCP = AUCP * 20$$

El valor de Use Case Point (UCP) obtenido indica el esfuerzo de horas hombre que se deben invertir para desarrollar el proyecto de software.

3 Ajuste de Frohnhoff y Engels sobre el Método UCP

Frohnhoff y Engels realizaron pruebas de estimación sobre 15 proyectos de software de distintos sectores industriales, comparando el esfuerzo estimado aplicando el método UCP de Karner contra el esfuerzo real del proyecto, obteniendo como resultado una desviación estándar de un 42% [5].

Una desviación de esta magnitud no es aceptable en la industria del software porque tiende a una sobrestimación del esfuerzo real requerido.

Los autores relevaron las causas de estos desvíos con los líderes de proyectos de los diferentes casos de estudio, arribando a dos hipótesis principales:

- Ausencia de estandarización de casos de uso
- El modelo de proceso de desarrollo de software aplicado:
 - procesos estándar (Rational Unified Process (RUP), V-model)
 - procesos limitados
 - procesos con estrictas políticas de desarrollo

Poniendo énfasis sobre la hipótesis de los modelos de procesos, Frohnhoff y Engels enfocaron los ajustes sobre la influencia que posee la definición de requerimientos en el proceso de desarrollo y el esfuerzo invertido en el proyecto. El modelo propuesto para el cálculo del esfuerzo de proyecto (PE) es el siguiente:

$$PE = (PF * M-factor) * (T-factor * A-factor)$$

A-factor (Application factor): está definida por la clasificación de los casos de uso, que representan los requerimientos funcionales del proyecto. Este valor es análogo al UUCP del método original.

Se estandariza el nivel de complejidad de cada caso de uso en función de sus escenarios principales, transacciones y cuadros de diálogo:

- Simple: ≤ 3 escenarios principales, transacciones y cuadros de dialogo (5 puntos)
- Medio: ≤ 7 escenarios principales, transacciones y cuadros de dialogo (10 puntos)
- Complejo: ≥ 8 escenarios principales, transacciones y cuadros de dialogo (15 puntos)

Los pesos de la clasificación son tomados de acuerdo al método original de UCP, n es la cantidad de casos de uso, resultando:

$A\text{-factor} = (\text{Suma de } 1 \text{ a } n \text{ del peso de los casos de uso}) + UAW \text{ (valor original del método UCP)}$

T-factor: Representa los requerimientos no funcionales. Toma las variables y pesos originales del método de UCP estandarizando los valores de influencia de complejidad. Se incluye una guía para indicar el grado de influencia con valores 0 (irrelevante), 3 (influencia media) o 5 (influencia alta). Se denomina G al grado de influencia, W al peso del factor técnico de complejidad, resultando:

$$T\text{-factor} = 0,58 + \sum_{i=1}^{13} (W_i * G_i * 0,01)$$

M-factor (Management Factor): Se reducen las variables de influencia de entorno (E) del método original de 8 a 6. Se excluyen el factor E1 (conocimiento de RUP) y el factor E3 (conocimiento del paradigma OO), justificando que no poseen una influencia significativa ya que hay un alto porcentaje de los proyectos que se desarrollan utilizando procesos UML y bajo paradigmas OO. Se incluye una guía para indicar el grado de influencia con valores 0 (irrelevante), 3 (influencia media) o 5 (influencia alta).

Los autores agregan una nueva variable de influencia M7 que representa al modelo de proceso a utilizar, tal como se muestra en la Tabla 1.

Tabla 1. M7 como nueva variable y sus diferentes grados de influencia

M-factor	Grado de influencia	Peso
M7- Modelo de Proceso	0: procesos limitados	-4,5
	3: procesos estándar (RUP, V-model)	
	5: procesos con estrictas políticas de desarrollo	

Luego, la nueva clasificación de M-factor según Frohnhoff y Engels, queda establecida como se presenta en la Tabla 2.

Tabla 2. Clasificación de M-factor

M-factor	Descripción	Peso
M1	Experiencia en la aplicación	-0,5
M2	Capacidades del líder de proyecto	0,0
M3	Motivación	2,5
M4	Requerimientos Estables	2,0
M5	Grupo de desarrolladores Part-Time	-1,0
M6	Dificultad en el lenguaje de programación	0,0
M7	Modelo de proceso	-4,5

Finalmente, considerando G como el grado de influencia y M como el peso del valor de M-factor, resulta:

$$M\text{-Factor} = 0,8875 - \sum_{i=1}^7 (M_i * G_i * 0,025)$$

La estandarización de los pesos de M-factor causó un ajuste de escalabilidad en el factor de productividad (PF) de 28,7 a 35 hh/UCP.

Frohnhoff y Engels aplicaron el nuevo método ajustado sobre los 15 proyectos mencionados al inicio de esta sección, obteniendo como resultado una desviación de un 20% entre el esfuerzo estimado y el esfuerzo real del proyecto; en comparación con la desviación de un 42% obtenida utilizando el método Karner [5].

4 Aplicación en Casos de Estudio

El método de UCP ajustado por Frohnhoff y Engels fue aplicado en 12 casos de estudio, 9 tesinas de grado de proyectos de software (procesos limitados según la clasificación M-factor) de la Facultad de Ingeniería de la Universidad FASTA de la ciudad de Mar del Plata, Argentina; y en 3 proyectos de la industria de software de la empresa Maker Electrónica, Argentina (www.makerelectronica.com.ar).

En estos casos de estudio se aplicaron las mismas políticas de desarrollo (procesos estándar según la clasificación M-factor). El esfuerzo que se debía aplicar para su desarrollo fue estimado utilizando la metodología propuesta por Frohnhoff y Engels, además de la experiencia en el dominio del gerente del proyecto y el grupo de desarrolladores, arribando a interesantes conclusiones luego de finalizados los proyectos.

4.1 Descripción de Proyectos

A continuación se describen muy brevemente los proyectos utilizados:

- CE1- Gestión de cuentas corrientes en cooperativa de remises: Sistema informático para gestionar las cuentas corrientes e información de las personas que poseen un vínculo contractual con la cooperativa.
- CE2- Gestión Técnica y Panel de Control de equipos AVL: Diseñar e implementar un sistema de software que permita la administración de toda la información relevante y ciclo de vida de los equipos posicionamiento satelital AVL (Automatic Vehicles Localisation) producidos por la empresa Maker Electrónica. Centralizar la información y unificar el sistema de Administración de Empresas cliente vigente.
- CE3- AVL-Desktop: Diseñar e implementar un sistema de escritorio en donde se visualice en un mapa digital, la posición de la flota de vehículos de una empresa en tiempo real. El seguimiento satelital de móviles se realiza bajo una aplicación Web; el proyecto AVL-Desktop es una extensión de la aplicación Web AVL.
- CE4- Sistema GAMA [9]: Sistema informático que contribuye a la gestión de la atención médica ambulatoria de pacientes en el Instituto Nacional de Epidemiología (INE) con sede en Mar del Plata, aplicable a otras entidades de salud.
- CE5- SCRUM [10]: Solución informática para la Unión de Rugby de Mar del Plata, la cual le permita gestionar de manera integral su actividad, a través del desarrollo de una aplicación Web y una aplicación de Escritorio.

- CE6- DO-RE-MI [11]: Sistema de Gestión para un conservatorio de Música.
- CE7- SLT [12]: El objetivo del proyecto es brindar al usuario final una solución informática que administre en forma integral y eficiente la logística de una empresa de transporte.
- CE8- SIARER [13]: Sistema Informático de Análisis de Recursos Energéticos.
- CE9- MARATHON [14]: Sistema de gestión deportiva y planificación de entrenamiento.
- CE10- ZONDA [15]: Sistema Informático de Análisis de Recursos Energéticos Renovables.
- CE11-C.T.E.N.P [16]: Sistema Informático para gestionar las tareas de la Escuela Nacional de Pesca de la ciudad de Mar del Plata.
- CE12-GySP [17]: Sistema Informático para gestionar y supervisar proyectos de software.

4.2 Resultados obtenidos

Los valores obtenidos de la aplicación del método de Frohnhoff y Engels se presentan en la tabla 3. Los valores de la estimación de esfuerzo de los casos de estudio están expresados en horas/hombre.

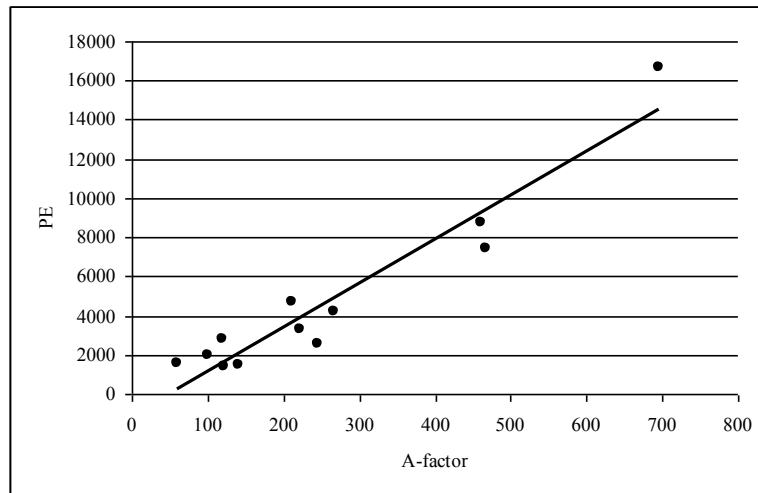
Tabla 3. Valores obtenidos según el método de Frohnhoff y Engels.

Caso de Estudio	Tipo Proyecto	A: Esfuerzo Real (HH)	A-factor	T-factor	M-factor	B: PE(HH)	Desfase (B-A) /A*100
CE01	Gestión	750	211	0,915	0,700	4730,09	530,68%
CE02	Industria	173	120	0,800	0,825	2772,00	1502,31%
CE03	Industria	305	59	0,910	0,850	1597,28	423,70%
CE04	Gestión	5085	461	0,940	0,575	8720,97	71,50%
CE05	Gestión	2025	266	0,925	0,488	4198,23	107,32%
CE06	Gestión	1236	696	0,980	0,700	16710,96	1252,02%
CE07	Gestión	3762	468	1,070	0,425	7448,81	98,00%
CE08	Simulación	4577	101	1,005	0,550	1953,97	-57,31%
CE09	Gestión	1810	246	0,905	0,325	2532,42	39,91%
CE10	Simulación	3282	140	0,940	0,325	1496,95	-54,39%
CE11	Gestión	1244	221	0,940	0,450	3271,91	163,015
CE12	Gestión	1641	121	1,015	0,325	1397,02	-14,87%
<i>Media</i>		<i>2157,50</i>	<i>259,17</i>	<i>0,95</i>	<i>0,54</i>	<i>4671,98</i>	<i>116,55%</i>

5 Análisis de A-factor

En la Figura 1 se analiza la relación entre el esfuerzo del Proyecto (PE) y el valor de A-factor, en la cual se puede visualizar la tendencia de incremento del valor de PE a medida que el valor de A-factor aumenta en rangos mínimos.

Figura 1. Tendencia lineal del Esfuerzo de Proyecto con respecto al valor A-factor.



Se toman los valores medios de los casos de estudio utilizados para ejemplificar esta tendencia.

$$PE = (35hh * M\text{-factor}) * (T\text{-factor} * A\text{-factor}) = (35hh * 0,54) * (0,95 * 259) = 4649hh$$

A continuación se utilizan los valores medios de la Tabla 3 para demostrar, mediante un ejemplo concreto, la sensibilidad de la variable A-factor con respecto al contexto del proyecto, donde se obtiene como resultado una sobrestimación del esfuerzo requerido.

Si se agrega un nuevo caso de uso de una complejidad media, le corresponde un peso de 10, y en consecuencia la fórmula para el cálculo quedaría de la siguiente manera:

$$PE = (35hh * M\text{-factor}) * (T\text{-factor} * A\text{-factor}) = (35hh * 0.325) * (0.905 * 269) = 4819hh$$

El ingreso de una nueva funcionalidad al proyecto incrementó el esfuerzo estimado en un 3,6%, con una diferencia de 170hh. Esto significa que la nueva funcionalidad de complejidad media posee una inversión de 170hh para finalizarla.

Se puede observar la sensibilidad del valor de A-factor a raíz de la inclusión de un requerimiento de mediana complejidad, el cual incurrió en una sobrestimación que se aleja de los márgenes aceptables de dispersión, optimista o pesimista, en cuanto a la estimación de esfuerzo de proyecto.

Así como en su momento Frohnhoff y Engels analizaron el método de Gustav Karner y propusieron una mejora enfocándose en las variables tecnológicas y de entorno, el trabajo de análisis aquí expuesto sugiere una mejora de éste método enfocándose en la variable A-factor, la cual es sensible al contexto de los requerimientos funcionales de un proyecto de software.

La mejora propuesta es lograr una normalización de los casos de uso que derive en una reclasificación cualitativa y cuantitativa de los mismos; y de esta manera fortalecer la influencia de la variable A-factor.

6 Conclusiones y Trabajos Futuros

Lo expuesto en este trabajo indica que la aplicación del método de Puntos de Caso de Uso propuesto por Frohnhoff y Engels, sobre doce casos de estudio reales, genera desviaciones excesivas en la estimación de esfuerzo de un proyecto de software, debido a la sensibilidad de la variable A-factor, la cual ha sido mostrada con un ejemplo concreto en el punto 5.

La clasificación cuantitativa y cualitativa otorgada por Frohnhoff y Engels a los casos de uso, no es parámetro convincente a la hora realizar una estimación de esfuerzo en una etapa preliminar del proyecto, ya que la misma es causal de la debilidad que radica en la variable A-factor dentro de éste método. Consecuentemente debido a ésta característica y a los resultados mostrados en este trabajo, es necesario realizar un ajuste de esta variable

Así como Frohnhoff y Engels enfocaron su mejora en los aspectos tecnológicos y de entorno, éste trabajo sugiere aplicar una mejora en el entorno de los requerimientos funcionales del proyecto, tomando como punto de partida la normalización de casos de usos, que es la información de entrada de esta variable, para luego abordar la reclasificación de la complejidad de los casos de uso.

Los casos de uso son una herramienta útil e importante en el desarrollo de proyecto de software. Representan el punto de partida para la diagramación de un sistema orientado objetos, los puntos bases de la planificación del proyecto, y la documentación de entrada en la validación de requerimientos y en etapas de testeo.

Finalmente, como trabajo futuro se prevé:

- La normalización de los casos de uso utilizados, con el objetivo de poder realizar una re-clasificación de complejidad del caso de uso.
- Analizar los valores y grado de influencia de A-factor sobre cada proyecto.
- Realizar nuevas pruebas de campo con los ajustes aplicados.

Bibliografía

1. Kusumoto, S., Matsukawa, F., Inoue, K., Hanabusa, S., Maegawa, Y.: Estimating Effort by Use Case Points: Method, Tool and Case Study. In: Proceedings of the 10th International Symposium on Software Metrics, pp. 292--299 (2004)
2. Sommerville, I.: Ingeniería de Software, 6ta. Edición. Addison, Wesley (2002)
3. Anda, B., Dreiem, H., Sjøberg, D.I.K., Jørgensen, M.: "Estimating Software Development Effort Based on Use Case – Experience from Industry. In: M. Gogolla,

- C., Kobryn , C. (Eds.) UML 2001. LNCS, vol. 2185, pp. 487-502. Springer-Verlag (2001)
4. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object Oriented Software Engineering: A Use Case Driven Approach. (ACM Press) Addison-Wesley (1992)
 5. Frohnhoff, S., Engels, G.: Revised Use Case Point Method - Effort Estimation in Development Projects for Business Applications. In: Proceedings of the 11th International Conference on Quality Engineering in Software Technology, Potsdam, dpunkt.verlag (2008)
 6. Karner, G.: Metrics for Objectory, Degree thesis, Universidad de Linkoping, Suecia (1993)
 7. Coleman, D.: A Use Case Template: draft for discussion. Fusion Newsletter (1998)
 8. Cockburn, A.: Writing Effective Use Case. Boston, Addison-Wesley (2001)
 9. Cucchi Colleoni, A., Di Croce, V., Sansevero, R.: Proyecto de Tesis: GAMA. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2007)
 10. Ghigliani, J., Fernández, G.: Proyecto de Tesis: SCRUM. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2005)
 11. Cosia, V., Villen, C.: Proyecto de Tesis: DO.RE.MI. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2009)
 12. Ferrari, M., Ortiz, S., Seoane, L., Ullo, M.: Proyecto de Tesis: SLT. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2007)
 13. Abadie, E., Bressán, J., Guzmán, E.: Proyecto de Tesis: SIARER. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2010)
 14. Gáspari, F., Remón, C.: Proyecto de Tesis: MARATHON. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2006)
 15. Albornoz, D., Posse, J., Speratti, N.: Proyecto de Tesis: ZONDA. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2010)
 16. Aroca, A., D'Angelo, J.: Proyecto de Tesis: CTENP. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2007)
 17. Pavón, J., Rueda, J.: Proyecto de Tesis: GySP. Facultad de Ingeniería, Universidad FASTA, Mar del Plata (2006)