

Creación de Programas: Proceso Evolutivo de Polya e Inteligencia Exitosa de Sternberg

Elizabeth JIMÉNEZ REY (ejimenezrey@yahoo.com.ar)

Gregorio PERICHINSKY (gperichinsky@acm.org)

Departamento de Computación. Facultad de Ingeniería (Sede Paseo Colón). UBA.
(C1063ACV) Av. Paseo Colón 850, 4° Piso, Tel. 4343-0891 Int. 140/142, Fax 4345-7261.
Ciudad Autónoma de Buenos Aires. República Argentina.

Resumen

En este trabajo se propone integrar el método evolutivo de Polya para la resolución de problemas con las tres formas de la inteligencia exitosa de Sternberg (*analítica, creadora y práctica*) al ámbito de la creación de programas. El presente artículo conforma una línea de trabajo en el área de Educación, dentro del Proyecto de Investigación Acreditado UBACYT I015 “Manufactura Integrada por Computadora en Sistemas Complejos para el Desarrollo Social, Industrial y de Tecnología”.

En publicaciones anteriores se consideró el uso del Mapa Conceptual para abordar la enseñanza de la materia Computación en la Facultad de Ingeniería de la UBA. El Mapa Conceptual, como representación del proceso de creación de programas, esquematiza conceptos y procedimientos (*memoria semántica*) posibilitando al Docente el desarrollo del contenido (*el qué y el cómo enseñar*) en forma evolutiva e incremental y, como herramienta estratégica educativa, permite implementar el proceso de aprendizaje humano (*modelo de la memoria de dos almacenes*), facilitando al Alumno la comprensión del conocimiento procedimental y operativo, asistiéndolo en la aplicación del método científico de resolución de problemas a través de cada fase del proceso de creación de un programa (*inteligencia analítica*).

Se focaliza la enseñanza y el aprendizaje de la resolución de problemas con la computadora como un medio para entrenar a los alumnos en la elaboración de nuevas y más eficientes

estrategias para decidir y actuar, partiendo de un pensamiento autónomo, creativo, crítico y productivo.

Palabras Clave

Método de Polya, Inteligencia Triárquica, Creación de Programas, Mapa Conceptual.

Escenario Educativo

La materia de grado Computación en la Facultad de Ingeniería de la Universidad de Buenos Aires es considerada de formación básica obligatoria para alumnos de todas las carreras de Ingeniería, excepto para aquellos que cursan Ingeniería Electrónica e Ingeniería en Informática. Los alumnos pueden cursarla en distintas etapas de avance en sus planes de estudio, pues la única materia correlativa posterior es Análisis Numérico. Algunos alumnos provienen de escuelas técnicas y tienen conocimiento y experiencia previa en programación pero la mayoría de los alumnos nunca programó. La materia es cuatrimestral y tiene una carga horaria de cuatro horas por semana.

La coexistencia de alumnos de distintas carreras y niveles de formación dificulta el abordaje y elaboración de contenidos por parte del Docente, así como la constitución de grupos de trabajo por parte del Alumno. La algoritmia representa para los estudiantes un nuevo paradigma para resolver problemas y les produce un fuerte impacto en su predisposición al aprendizaje que, en muchos casos, se traduce en falta de motivación o en rechazo.

Son los objetivos de la asignatura:

- Concientizar al Alumno de la importancia de la Algoritmia como paradigma de resolución de problemas y de la Programación como práctica y ejercitación en la resolución de problemas (*enseñanza basada en problemas*).
- Promover en el Alumno el desarrollo de la capacidad de abstracción, la capacidad de relacionar esquemas de solución con la resolución de problemas algorítmicos, haciendo hincapié en el método científico (*desarrollo de habilidades analíticas, creativas y prácticas*).
- Analizar, Sistematizar, Programar y Procesar distintos problemas de tipo técnico-científicos para que dichos conocimientos resulten útiles al Alumno tanto en el desarrollo de la carrera como en la actividad profesional.
- Sustentar la estrategia de enseñanza y de aprendizaje de los conceptos y procedimientos en los recursos tecnológicos como mediadores de los procesos y las habilidades cognitivas a inducir en el alumno.
- Comprender conceptos y técnicas de la disciplina que en su presente académico y su futuro profesional habilite al Alumno a interactuar en forma interdisciplinaria con pares y profesionales en informática sin problemas de comunicación. [5]

Contexto de Enseñanza y Aprendizaje

Las computadoras electrónicas constituyen una herramienta esencial en muchas áreas: industria, gobierno, ciencia, educación ... en realidad, en casi todos los campos de nuestras vidas. Uno de los objetivos fundamentales de las Ciencias de la Computación es la resolución de problemas con la computadora.

La práctica de la Ingeniería, como disciplina tecnológica, requiere capacidad creativa para inventar soluciones a los problemas y para utilizar múltiples criterios en la evaluación de las soluciones; es decir, el ingeniero debe ser capaz de identificar las características básicas de los problemas que tenga que resolver y diseñar buenas soluciones a los problemas. [5]

La enseñanza y el aprendizaje de la materia Computación se aborda focalizando la estrategia educativa en el principio procedimental para la creación de programas, el cual se sustenta en el

proceso de resolución de problemas del matemático George Polya que consta de cuatro fases: 1) Comprender el problema. 2) Idear un plan (*formular una estrategia general*). 3) Ejecutar ese plan (*formular una prueba detallada*). 4) Mirar hacia atrás (*verificar los resultados*), como se muestra en la Figura 1.

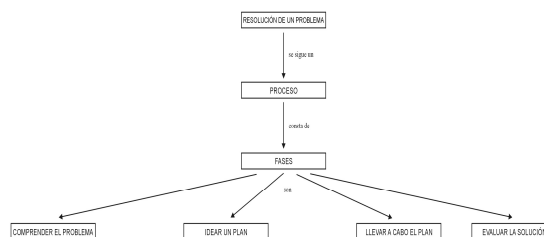


Figura 1

En el marco del Modelo de Polya aplicado al ámbito de la construcción de programas, el Docente inicia al Alumno en el conocimiento operativo necesario para la creación de un programa Pascal (parte práctica de la materia). El Docente interactúa con el Alumno y juntos navegan en forma evolutiva a través de las fases del proceso para desarrollar un programa hasta completarlas: 1) Análisis. 2) Diseño. 3) Codificación. 4) Evaluación, como puede visualizarse en la Figura 2.

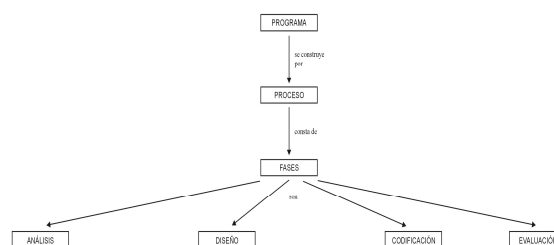


Figura 2

En Anexo se representa el esquema detallado del proceso, en el contexto de la creación de programas, por medio de un Mapa Conceptual. Se presenta el Mapa Conceptual como el

instrumento que, utilizado en forma iterativa durante el curso para la enseñanza y el aprendizaje de cada nueva herramienta de programación a manera de ciclo, permite el desarrollo del contenido (*el qué y el cómo enseñar y aprender*) en forma evolutiva e incremental, teniendo como eje la mediación electrónica y atendiendo a los procesos y habilidades cognitivas implicados. [1]

La Teoría Triárquica de la Inteligencia de Robert Sternberg (1949-), considera tres aspectos [1]:

**A.-La relación de la inteligencia con el mundo interno del individuo:
los componentes de la inteligencia**

La teoría pretende describir la relación de la inteligencia con el mundo interno del individuo a través de los componentes o procesos mentales que están implicados en el pensamiento. Estos componentes son de tres tipos: 1) metacomponentes, 2) componentes de ejecución, y 3) componentes de adquisición del conocimiento. Los metacomponentes son los procesos ejecutivos usados para planificar, controlar y evaluar la solución de los problemas o tareas. Los componentes de ejecución son los procesos de nivel más bajo de todos los procesos implicados en hacer efectivos (o ejecutar) todos los mandatos de los metacomponentes. Y los componentes de adquisición del conocimiento son aquellos procesos utilizados, en primer lugar, para aprender a solucionar los problemas. Estos tipos de componentes son altamente interdependientes como se puede ver en la Figura 3.

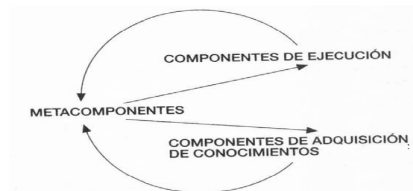


Figura 3

**B. La relación de la inteligencia con la experiencia individual:
facetas de la inteligencia humana**

Los tres tipos de componentes de procesamiento de la información se aplican a tareas y situaciones que varían en cuanto a los niveles de experiencia individual para con ellos. Cuando la tarea se presenta por primera vez es novedosa. Por medio de sucesivas experiencias se va automatizando. En la Figura 4 se muestra la relación entre tareas novedosas y automatizadas.

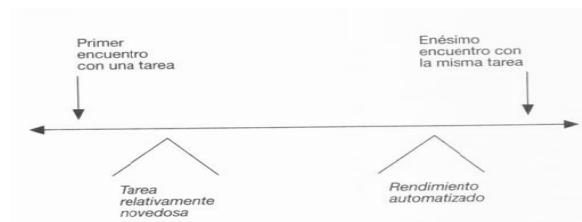


Figura 4

C. El contexto de la inteligencia: funciones del autogobierno mental

En esta teoría, la inteligencia cotidiana, de cada día, se define «como la finalidad adaptativa a, selección de, y modificación de los ambientes del mundo real que son relevantes en nuestra vida y para nuestras aptitudes». Primero se elige la adaptación y, si ésta falla, se opta por la selección o moldeamiento. Se puede observar en la Figura 5 las relaciones entre adaptación, selección y moldeamiento.

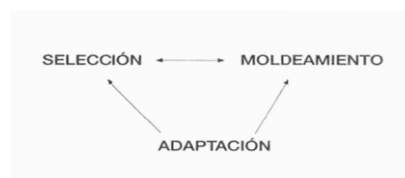


Figura 5

La utilización del Mapa Conceptual para la creación de programas promueve la participación activa del alumno en la construcción de su conocimiento, permitiendo el despliegue de habilidades analíticas en la resolución de problemas con la computadora, a través de [4]:

- Análisis de situaciones problemáticas en términos de los metacomponentes: 1) Definir

la naturaleza de un problema 2) Seleccionar los componentes o pasos para resolverlo 3) Seleccionar una estrategia para ordenar los componentes en la resolución del problema 4) Seleccionar una representación mental para la información 5) Distribuir los recursos mentales 6) Monitorear la solución.

- Identificación de analogías, comparaciones, relaciones: concretas-abstractas, implícitas (inferidas)- explícitas; de procesos inductivos, deductivos.

- Consideración de la experiencia previa necesaria (requisitos) o grado de familiaridad para encarar el material en términos de conocimiento (información pertinente) y habilidades (búsqueda, selección, codificación, uso).

Polya opina que el conocimiento del dominio es importante para resolver problemas en un dominio particular [2]:

- Una buena provisión y acumulación de conocimiento bien organizado es un recurso importante para resolver problemas. En cualquier materia, hay hechos importantes: 1) Tener algunos hechos claves sobre como comenzar 2) Saber las conexiones del dominio así como se sabe las rutas más directas hacia su casa en su pueblo 3) Cuanto más extenso es su conocimiento y bien organizado, hay más oportunidad de encontrar lo que se necesita.

- Comenzar con las partes pertinentes del conocimiento probablemente será muy relevante para el problema.

- Intentar la generalización de la prueba, la especialización, la analogía, la estructura global del dominio.

Diseño de la Estrategia de Uso del Recurso

Constituyen el *conocimiento previo* del Alumno, adquirido en clase introductoria a través de actividades grupales para producir un *aprendizaje significativo*:

- el Modelo de Polya (Figura 6) representa el método científico para la resolución de un problema cualquiera, un ciclo iterativo y evolutivo (los alumnos resolvieron un problema matemático para incorporarlo). *Metacognición, pensar sobre el pensamiento.*

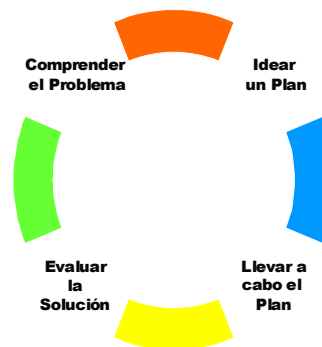


Figura 6

- el Modelo de Programa Tipo representa el punto de partida para la escritura de un programa en lenguaje Pascal (los alumnos analizaron el texto de una receta como aplicación del Modelo de Polya para incorporarlo). *Cognición situada, Reflexión desde la acción.*

{SECCIÓN DECLARATIVA}

{Definición del Objetivo}

{Definición de Recursos}

{SECCIÓN ALGORÍTMICA}

{Desarrollo de la Solución}

{Prólogo}

{Resolución}

{Epílogo}

El Mapa Conceptual lo utilizará el Docente en cada clase presencial para representar el proceso de creación de un programa Pascal (*conocimiento declarativo*) y para que el Alumno experimente cómo debe proceder para desarrollar un programa Pascal a partir del enunciado de un problema (*conocimiento procedimental*).

En la primera clase, el Docente usa el Mapa Conceptual (*aprendizaje situado*) para desarrollar en forma interactiva con el Alumno el primer programa Pascal, a partir del Modelo de Programa Tipo (*procesamiento de arriba*

abajo), utilizando estructuras de control secuenciales y tipos de datos simples. El conocimiento nuevo se integra al mapa conceptual el cual se modifica (*proceso de elaboración*). El Alumno almacena en su memoria a largo plazo el nuevo conocimiento esquematizado en el mapa. [6]

El material (detalle de cómo navegar o transitar por cada una de las fases del proceso hasta completarlas) se carga en el entorno virtual para que el Alumno lo utilice como guía para la escritura de programas con estructuras de control secuenciales (*saber hacer haciendo*).

Se propone a los alumnos la resolución en computadora, en forma grupal o individual, para la siguiente clase, de situaciones problemáticas nuevas (*afrentamiento de tareas nuevas*) que generen distintos esquemas de solución. A medida que aumenta la experiencia con nuevos tipos de situaciones, la novedad decrece y la nueva tarea a resolver llega a ser una medida apropiada de las habilidades que ya han sido automatizadas (*automatización del procesamiento de la información*). Se promueve el despliegue de habilidades creativas que favorezcan el desarrollo de procesos de *insight*, aptitud para enfrentarse bien con tareas nuevas o situaciones desconocidas, en las instancias de desarrollo de los Trabajos Prácticos o Evaluaciones Parciales. [3]

Polya redujo los elementos críticos del proceso a tres: 1) Materiales inherentes al problema (*¿cuál es el problema?*). 2) Conocimiento disponible (*¿qué conocimiento está a su disposición?*). 3) Problemas auxiliares (*¿qué otros problemas similares o paralelos ayudarían? ¿se ha resuelto un problema así en un área diferente pero contenida en el dominio?*). [2]

En la clase siguiente, el mapa conceptual se recupera de la memoria a largo plazo del Alumno (*recuperación del conocimiento declarativo y de procedimientos*), se repasa y refuerza los conocimientos adquiridos en la clase anterior y se repite el proceso de utilización del mapa conceptual como soporte para desarrollar un programa Pascal integrador

utilizando ahora estructuras de control secuenciales (*conocimiento previo*) y estructuras de control selectivas (*conocimiento nuevo*). El Alumno almacena en su memoria a largo plazo el nuevo conocimiento esquematizado en el mapa (*integración de la nueva información, significativa, elaborada y organizada, en las redes de la memoria a largo plazo, haciendo explícitas las relaciones entre el material nuevo y el previo*). [7]

En las siguientes clases se repite el proceso integrando al Mapa Conceptual las estructuras de control repetitivas (Primer Módulo) y las estructuras de control de transferencia-retorno y los tipos de datos estructurados (Segundo Módulo). *Mejoramiento de la memoria con más procesamiento del mismo tipo*.

Los problemas a desarrollar en los Trabajos Prácticos y las Evaluaciones Parciales permiten al Alumno la aplicación de la información en distintos contextos. Se favorece el despliegue de la inteligencia práctica en el Alumno optimizando sus fortalezas y compensando sus debilidades. *Transposición Didáctica*.

En la Teoría Triárquica, las partes, aunque distinguibles, trabajan juntas de manera integrada: una componencial relaciona la inteligencia con la parte interna del individuo, una experiencial relaciona la inteligencia tanto con el mundo interno como con el externo y la contextual relaciona la inteligencia con el mundo externo del individuo. [3]

Impacto Psicológico en la Métrica del Desarrollo

Cuando el modelo estático es conceptual, es decir, de máximo nivel de abstracción, y los mecanismos de abstracción al planificar la clasificación, la agregación y la generalización, el uso de cada uno de estos mecanismos debe ser considerado en la medida de la complejidad del código[8][9].

La abstracción de la clasificación se usa para definir un concepto como una clase de objetos de la realidad que se caracteriza por las propiedades comunes. Una abstracción de agregación define una nueva clase a partir de un juego de otras clases o tipos que

representan sus componentes de parte. Una abstracción de la generalización define una relación de subconjunto entre los objetos o entidades de dos o más clases.

Por consiguiente, los parámetros que pueden ser considerados para una medida de complejidad serían cantidad de C de las clases (la clasificación), cantidad total de atributos UN (la agregación) y cantidad parcial de CA de los atributos complejo (la clasificación y agregación), cantidad total de relaciones o asociaciones entre las clases R (la agregación) y cantidad parcial de asociaciones como clasifica CR (la clasificación y agregación) y cantidad de clases que ellos son los subconjuntos de la generalización G.

La complejidad del sistema puede medirse como

$$C + C * VA + C * VR + C * VG = C * (1 + VA + VR + VG)$$

donde VA, VR y VG son factores que aumentan la complejidad en la función de los atributos, relaciones y generalizaciones, y es calculado

$$VA = C/A * (1 + CA/A) \quad VR = R/C * (1 + CR/R) \\ VG = G/C$$

Al medir funciones se tiene que un Programa <P>, es un quintuplo de atributos característicos.

Las características de este programa son la Longitud (el número de líneas de código), Funcionalidad (cumple los requisitos funcionales), y la Complejidad.

Otra característica a tener en cuenta es la reusabilidad o cuánto un programa es genuinamente nuevo.

La primera conclusión es que el esfuerzo debe tomar en la redundancia de cuenta y complejidad; la productividad como una verdadera funcionalidad y esfuerzo; y finalmente el costo de la reusabilidad y complejidad.

La complejidad involucra el problema de la aplicación o la complejidad de cálculo (Problem Solving), la estructura del software

lleva a cabo el algoritmo y el esfuerzo para entender el problema del software.

Siguiendo a la Ciencia de Halstead para la Métrica del Software se comienza el Programa <P>, con una colección de rótulos (tokens), tal que:

n1 = el número de operadores monádicos.

n2 = el número de operando monádicos.

N1 = las ocurrencias totales de operadores.

N2 = las ocurrencias totales de operandos.

La longitud de <P> = N = N1 + N2

El vocabulario de <P> = n = el n1 + el n2

El volumen de <P> = V = N log2 x n

Donde el Volumen es que el total de comparaciones necesitó escribir <P>.

Nivel <P> = L = V * / V

Donde V * es el tamaño mínimo de aplicación de <P>.

La dificultad = 1 / L

Según la teoría de Halstead, se puede calcular una estimación L * de L como

$L * = 1 / D = 2 / n1 \ n2 \ x / N2$

$N = n1 \ x \ log2 \ n1 + n2 \ x \ log2 \ n2$

El esfuerzo de <P> = E = V / L * = n1 N2 log2 N n / 2 n2

Los estudios psicológicos dentro de la Ciencia de Halstead exigió $\beta = 18$ donde β permite computar el T de tiempo de programación igual al Esfuerzo / 18 segundos, aunque los valores de β varían entre un mínimo de 5 y un máximo de 25.

Donde el número de β es una limitación de la mente humana para hacer las comparaciones diferenciadas por segundo.

Para el Modelo del Costo desarrollado por Böehm y la fórmula exponencial para computar el Esfuerzo, la Planificación, Provisión de personal y Eficiencia, con factores de ajuste.

Para determinar el peso se debe multiplicar por tres niveles de complejidad.

Si nosotros computamos un factor técnico como el factor de complejidad de pesos cn(f) o ciclomático.

Mc Cabe propuso que un quintuplo del Programa <P> de atributos con una complejidad característica medidas a través de la descomposición en el orden jerárquico de

profundidad del camino de tipos: Principal (Requerido), Secuencia y Repetitivo y Anidado.

De esta manera computar el Número Ciclomático muy conocido: el $cn(f) = e - n + 2$ donde e son los arcos y n son nodos de un flujograma o digrafo del Programa [8][9].

Una forma especial de computar el número ciclomático es el número esencial como una suma de árboles de subdigrafos del digrafo del Programa.

Se puede demostrar la relación entre el Número Ciclomático y la Garantía de calidad del Software.

Se usa frecuentemente una forma empírica de Halstead dentro de la Ciencia del Software para encontrar una expresión de la ecuación de regresión polinómica que relaciona las faltas, tamaño, densidad y esfuerzo:

$D = A_0 + A_1 L + A_2 L^2$ donde L (o logaritmo de L) son líneas de código y cada A_i depende del promedio de operandos y operadores en cada línea de código y D las faltas que predicen la densidad.

Una forma de implementar en un curso es disponer de un sistema distribuido con cámaras para que se fijen atributos psicológicos de acuerdo al comportamiento de los alumnos. De allí se han obtenido matrices de comportamiento para usarlas como factores de ajuste en la interacción Docente - Alumno.

Conclusiones y Trabajo Futuro

La utilización del Mapa Conceptual como material educativo estratégico, en cuanto a la actividad cognitiva del alumno:

- Promueve el aprendizaje constructivista porque los alumnos adquieren el conocimiento declarativo y procedimental, el saber y el saber hacer, a través de la resolución de problemas, reflexionando en el hacer y desde el hacer. *Cognición situada*.
- Permite explicitar, clarificar, dar sentido, elaborar, organizar, estructurar la información, proveer recursos, entrenar y motivar al Alumno para facilitar los procesos de atención, percepción,

almacenamiento y recuperación, repaso y refuerzo del conocimiento. *Modelo de la Memoria de Dos Almacenes*.

- Activa el desarrollo de habilidades cognitivas **analíticas**: *definición* de la naturaleza de un problema (observación, comprensión, interpretación, comparación, relaciones concretas-abstractas, relaciones implícitas-explicitas, identificación de vacíos en la información, “identifique el problema”), *selección* de una representación mental para el problema, de los componentes o pasos para resolverlo (clasificación), de una estrategia para ordenar los componentes en la resolución del problema (ordenación), *distribución* de los recursos mentales (“actúe con lo que se tiene”), *implementación* del plan estableciendo relaciones de jerarquías entre los componentes (“evalúe las perspectivas”), *monitoreo* de la solución (“reitere el problema”), *síntesis*, *razonamiento* (deductivo, inductivo, crítico), *reflexión* metacognitiva (“¿está atascado! reimponga la solución”); **creativas**: *identificación* de componentes automatizados (entrenamiento en resolución de problemas típicos simples, “ataque de los problemas auxiliares”), *relación* de la información nueva con la disponible o existente (reconstrucción global, “busque problemas relacionados”), *búsqueda* de nuevos elementos o extensiones, *identificación* de relaciones no evidentes, *aplicación* de la técnica a situaciones novedosas (capacidad resolución de problemas originales complejos, interpretación constructiva), *creación*, *invención*, *exploración*, *descubrimiento*, *imaginación*, *suposición*; **prácticas**: *aplicación* del conocimiento a problemas de la vida cotidiana, *detección* de los aspectos prácticos del nuevo conocimiento, *implementación* de informaciones recientemente adquiridas, *identificación* de las fortalezas y debilidades para enfrentar las dificultades y tratar con ellas.

Se considera reformular la propuesta en un futuro cercano para migrar a un Mapa Conceptual Hipermedial que contribuya más efectivamente al despliegue de procesos y habilidades cognitivas en el Alumno.

Referencias

- [1] *La Teoría Triárquica de la Inteligencia de Sternberg aplicada a la Creación de Programas*. Jiménez Rey, E., Perichinsky, G. 2008. Aceptado para la presentación como poster. X Workshop de Investigadores en Ciencias de la Computación 2008. Facultad de Ingeniería. UNLPam. General Pico, La Pampa, Argentina. 5 y 6 de mayo.
- [2] *Información sobre Solución de Problemas: El Problema se resuelve Aplicando el método de Polya a un Trabajo de Referencia*. Perichinsky, G. (desarrollado de los escritos de George Polya). 2007. Base de Conocimiento Proyecto I015. UBACYT.
- [3] *Síntesis de la Teoría Triárquica de la Inteligencia Humana formulada por R .J. Sternberg*. Jiménez Rey, E. (desarrollado de los escritos de Robert Sternberg). 2007. Base de Conocimiento Proyecto I015. UBACYT.
- [4] *Pensamiento analítico, creativo y práctico*. Malbrán, M. (síntesis desarrollada para el Seminario sobre Psicología Cognitiva Aplicada a la Informática Educativa). 2007. Maestría en Informática Aplicada en Educación. Facultad de Informática. UNLP.
- [5] *Un Enfoque Procedimental para la Enseñanza de Computación en Carreras de Ingeniería*. Jiménez Rey, E. 2005. Proceedings de las Primeras Jornadas de Educación en Informática y TICs en Argentina, JEITICs 2005. (pp 35-39). Bahía Blanca, Provincia de Buenos Aires, Argentina.
- [6] *Teorías del aprendizaje*. Schunk, D. 1997. Capítulo 5. Prentice – Hall Hispanoamérica. México.
- [7] *Capacidad de resolución de problemas*. Chi, M. y Glaser, R. 1986. En: Sternberg, R. Las capacidades humanas. Un enfoque desde el procesamiento de la información. Labor Universitaria. Barcelona.
- [8] Fenton, N. and Pfleeger, S. *Software Metrics*. PWS Publishing Company. 1997.
- [9] Pfleeger, S. "Software Engineering: Theory and Practice". Prentice Hall. 1998.

Anexo

