

UN ENFOQUE PROCEDIMENTAL PARA LA ENSEÑANZA DE COMPUTACIÓN EN CARRERAS DE INGENIERÍA

Autor

M. Elizabeth JIMÉNEZ REY (ejimenez@mara.fi.uba.ar)

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.

Capital Federal. República Argentina.

Resumen

En esta propuesta se focaliza la estrategia de enseñanza y de aprendizaje de la materia Computación en el principio procedimental para la creación de programas, el cual se sustenta en el proceso de resolución de problemas cuyas fases fueron definidas por el matemático George Polya a finales de los años cuarenta. Se propone la representación de dicho 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 iterativamente 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) en forma evolutiva e incremental.

Palabras Clave

Enfoque procedimental, proceso de resolución de problemas, propuesta educativa.

Presentación

En este trabajo se presenta una propuesta de enseñanza de la materia de grado Computación en la Facultad de Ingeniería de la Universidad de Buenos Aires, considerada de formación básica obligatoria para alumnos de todas las carreras de Ingeniería, excepto para aquellos que cursan Ingeniería Electrónica e Ingeniería en Informática. El diseño de la propuesta a través de un enfoque procedimental se fundamenta en el modelo prescriptivo de solución de problemas de Polya que distingue 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) [6]. En el marco de este modelo aplicado al ámbito de la construcción de programas, se describe el contenido de la materia a través de las fases que se deben completar para crear un programa: 1) Análisis. 2) Diseño. 3) Codificación. 4) Evaluación. Se representa el modelo mediante un mapa conceptual que se utiliza como herramienta para la enseñanza y el aprendizaje significativo de los conceptos y procedimientos de la asignatura. Se explica de qué manera el mapa conceptual puede contribuir al logro del “aprendizaje significativo” en los alumnos postulado por David Ausubel en su teoría constructivista de la asimilación cognoscitiva a principios de los años sesenta.

Contextualización

Una descripción de ingeniería es esencialmente una descripción de problemas y de su resolución. Un problema proviene del deseo de lograr la transformación de un estado de cosas en otro e involucra algo más que hallar una solución: requiere encontrar una forma preferible de lograr la transformación deseada. En la mayor parte de los problemas hay muchas soluciones posibles. Una norma de referencia para seleccionar entre varias soluciones se llama criterio. El ingeniero debe ser capaz de identificar las características básicas de los problemas que tenga que resolver. La capacidad creativa necesaria para inventar soluciones y el criterio utilizado en su evaluación, significan que la práctica de la ingeniería tiene más de arte de lo que podría haberse supuesto. [8] En este marco referencial, se considera que los alumnos que cursan la materia Computación, como futuros ingenieros, tendrán que aprender a resolver problemas de tipo técnico-científicos utilizando la computadora. La solución a la que debe arribar el alumno no es la única posible porque se plantea la resolución de problemas creativos, es decir, que deben resolverse mediante la construcción original de la solución y del propio proceso de resolución por parte del alumno, lo cual requiere autonomía en la toma de decisiones. Se hace hincapié en que la resolución del problema debe ser metódica y en que la solución encontrada debe ser correcta y óptima. [3]

Caracterización del Escenario de Enseñanza

La materia es cuatrimestral y tiene cuatro créditos. La única materia correlativa de Computación es Análisis Numérico, por lo cual, los alumnos que cursan la materia pueden hacerlo en distintas etapas de avance en sus planes de estudio pero, en casi la totalidad de los casos, los alumnos la cursan en el primer año de su carrera, en el cual aún han desarrollado poco su capacidad de abstracción.

Se concibe la enseñanza como un proceso por medio del cual el docente transmite al alumno un contenido con la intencionalidad de que lo aprenda de acuerdo con un objetivo.

Son los objetivos básicos de la materia [1] :

- 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, como promotores del desarrollo de la capacidad de abstracción, la capacidad de relacionar esquemas de solución con la resolución de problemas algorítmicos, con hincapié en el método científico para lograr ese objetivo.
- Enseñar al alumno de Carreras de Ingeniería el Análisis, la Sistematización, Programación y Procesamiento de distintos problemas de tipo técnico-científicos a fin de que dichos conocimientos le resulten de utilidad ya sea en el desarrollo de la carrera como así también en su actividad profesional.

Se concibe el conocimiento como el conjunto de los conceptos esenciales y de las formas propias en que esos conceptos se relacionan para constituir la estructura conceptual del objeto en estudio.

Son los principios básicos de la materia [4] :

Conceptuales

1. La Computación es la disciplina que busca establecer una base científica para resolver problemas mediante la computadora.
2. El concepto fundamental de la Computación es el concepto de algoritmo.
3. Un algoritmo es un conjunto finito de instrucciones ejecutables, no ambiguas, que dirige a una actividad que termina o resuelve un problema en tiempo limitado.
4. Una computadora debe tener la capacidad de recibir entradas, procesar información, almacenar y recuperar información y producir salidas para poder ejecutar algoritmos.

Procedimentales

1. La creación de programas se basa en el proceso de resolución de problemas cuyas fases fueron definidas por Polya: Análisis, Diseño, Codificación y Evaluación (modelo de un programa tipo).
2. El descubrimiento de un algoritmo para resolver un problema equivale en lo esencial al descubrimiento de una solución para ese problema.
3. La estrategia más común para desarrollar algoritmos es la denominada “divide y vencerás” que consiste en descomponer el problema principal en una lista de problemas más simples, que a su vez se siguen descomponiendo hasta llegar a un nivel en que los problemas sean resolubles por acciones simples de la computadora (método de refinamientos sucesivos).
4. El algoritmo conceptual debe transformarse en un conjunto claro de instrucciones a ser representadas sin ambigüedad en forma de programa mediante un lenguaje de programación (primitivas de programación) para que se lo pueda comunicar a la computadora.

Propuesta Educativa

Qué enseñar

El eje epistemológico o el principio fundamental y estructurante de la materia es la noción de descubrimiento de algoritmos como método de resolución de problemas y la representación de algoritmos en forma de programa para que puedan ser comunicados a una computadora. Lo que el docente debe enseñar y lo que el alumno debe aprender es cómo retener, comprender y usar activamente los principios conceptuales y procedimentales disciplinares para que se produzca la transferencia del conocimiento

(conocimiento generador) [5] . En la Figura 1 se muestra una instancia del Mapa Conceptual para la enseñanza de creación de programas.

FASE DE ANÁLISIS

Se enseñará a los alumnos cómo deben leer el enunciado para poder realizar un análisis significativo del problema a resolver. Los alumnos deberán aprender a clarificar cual es el objetivo del enunciado para poder plantear bien el problema a resolver, es decir, tienen que comprender **qué** es lo que tienen que resolver.

FASE DE DISEÑO

Se enseñará a los alumnos **cómo** deben idear un plan para resolver el problema. Los alumnos deberán aprender cómo definir los recursos que se necesitarán para resolverlo y cómo descubrir el algoritmo que resuelve el problema. Se transmite así la forma de conocer propia de la Computación para resolver problemas mediante el diseño del algoritmo. En este sentido, el contenido a desarrollar y trabajar con los estudiantes será: *El algoritmo como objeto de la Computación. Alcance de la Computación considerando algunos de sus objetos de estudio desde el punto de vista de los algoritmos. Nociones básicas de Categorías de Computadoras y de Categorías de Software.*

Para definir recursos será necesario aprender a especificar sus nombres y dominios y a determinar cómo serán utilizados, es decir, verificar cuáles de ellos estarán asociados a valores que deben ingresar al algoritmo, cuáles a valores intermedios o de procesamiento y cuáles a valores resultantes que debe proporcionar el algoritmo. En este sentido, el contenido a desarrollar y trabajar con los estudiantes será: *Tipos de datos básicos o primitivos. Tipos de datos compuestos o estructurados. Archivos de texto.*

Para encontrar y desarrollar la solución será necesario aprender cómo se aplica la estrategia más común para romper la complejidad del problema planteado y desarrollar algoritmos, “divide y vencerás”, que consiste en descomponer el problema a resolver en una lista de problemas más simples, que a su vez se siguen descomponiendo hasta llegar a un nivel en que los problemas sean resolubles por acciones simples de la computadora (primitivas de programación no ejecutables). Este proceso se denomina también de refinamientos sucesivos o de diseño descendente. Será necesario aprender a individualizar, aislar y ordenar las distintas actividades para conseguir una solución. Se hace hincapié en la utilización de los enunciados de documentación interna y de la indentación de las sentencias para lograr la inteligibilidad de la solución. En este sentido, el contenido a desarrollar y trabajar con los estudiantes será: *Primitivas de programación en Pascal. Capacidades de una computadora. Organización de dispositivos de almacenamiento: principal (la memoria central de una computadora) y secundario (la memoria auxiliar de una computadora). Subprogramas del lenguaje (como recurso de programación y como estructura de control de transferencia/retorno). Acoplamiento de subprogramas. Uso de parámetros. Principios de modularización de programas.*

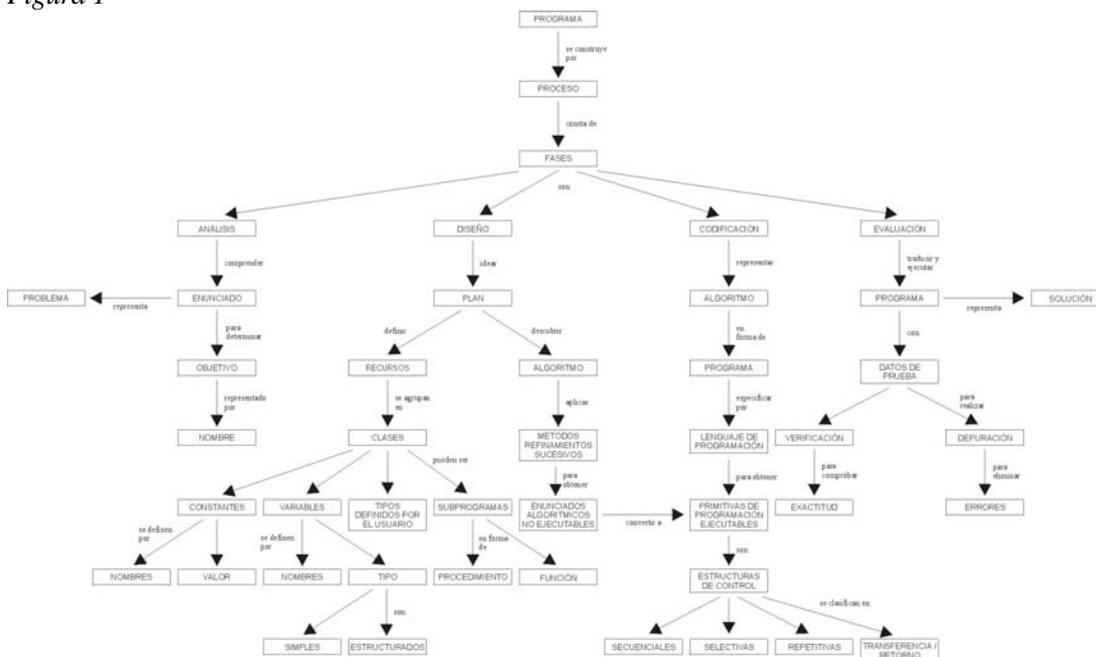
FASE DE CODIFICACIÓN

Se enseñará a los alumnos cómo representar el algoritmo en forma de programa para que pueda ejecutarlo la computadora. Los alumnos deberán aprender a transformar el algoritmo conceptual en un conjunto claro de instrucciones (primitivas de programación ejecutables) y representarlas sin ambigüedad a través de un lenguaje de programación (en este curso, el lenguaje Pascal). En este sentido, en el contenido se tratará: *Ambientes integrados de desarrollo de software. Escritura de un programa en lenguaje Pascal. Desarrollo de Programas. Modelo de un Programa Tipo. La unidad central de procesamiento. Instrucciones en Lenguaje de máquina. Ejecución de programas. Comunicación con periféricos.*

FASE DE EVALUACIÓN

Se enseñará a los alumnos cómo evaluar el programa en cuanto a su exactitud y a su potencial como herramienta para resolver otros problemas (probar y depurar el programa). Los alumnos deberán aprender a comprobar que la solución obtenida es buena ejecutando el programa con distintos valores testigo de los datos para los que se conoce o se infiere fácilmente cuál es el resultado correcto.

Figura 1



Cómo enseñar

La creación de programas constituye el núcleo de la materia desde el cual se integran los diferentes conceptos de computación asociados al desarrollo del curso. En cada clase expositiva, el Docente plantea al Alumno la resolución de un Problema con la computadora a través de un Enunciado para introducir la enseñanza de una nueva herramienta de programación.

Para enseñar a los Alumnos el método científico de resolución de problemas con la computadora, el Docente elabora (en la primera clase) y utiliza (en la primera clase y en las siguientes, en forma iterativa) un Mapa Conceptual para la creación de programas. El Mapa Conceptual representa la estructura conceptual que sirve de base al Docente para organizar el conocimiento durante el desarrollo del curso y sirve de guía al Alumno de manera tal que tenga siempre presente que dichas fases no son pasos que deban seguirse linealmente al tratar de resolver un problema, sino más bien fases que han de completarse evolutivamente en algún momento durante el proceso de resolución para encontrar la solución satisfactoria. El Mapa Conceptual posibilita el pensamiento “cooperativo” entre el Docente y el Alumno, quienes reflexionan juntos y constantemente en estrecha interacción (inteligencia repartida en el aula).

Para cada clase expositiva, el Docente selecciona un nuevo Problema (organizador previo) cuya resolución plantea a los Alumnos, de tal manera que en las Fases de Análisis y Diseño de la creación del programa, descubran que no pueden encontrar la solución con las herramientas de programación conocidas hasta ese momento (conocimiento previo). Se incrementa así la estabilidad y claridad de los conocimientos que ya tiene el Alumno antes de presentar el conocimiento nuevo. El procedimiento es “adaptativo” porque la interacción permite al Docente y al Alumno realizar la vigilancia epistemológica de la comprensión de los principios conceptuales y procedimentales, reforzando la estructura cognoscitiva del Alumno. Surge la necesidad de utilizar una nueva herramienta de programación (recurso o instrucción). Se provoca así el conflicto cognitivo y se produce la diferenciación progresiva. El Docente conecta sólidamente el conocimiento nuevo con el conocimiento existente (anclaje conceptual) en la estructura cognoscitiva del Alumno y promueve su recepción activa, produciendo la reconciliación integradora. Se integra el conocimiento nuevo al Mapa Conceptual, el cual se amplía y complejiza. La enseñanza y el aprendizaje del contenido es “incremental” pues se lo secuencia y organiza de manera que

cada aprendizaje sucesivo se conecte con lo presentado anteriormente y se produzca a través del esfuerzo creciente y constante del Alumno por lograrlo. [7]

El Mapa Conceptual ayuda al Docente a enseñar y presentar de manera explícita, directa, continua e integrada los principios conceptuales y procedimentales de la asignatura. La “sencillez” del Mapa Conceptual como material académico posibilita al Alumno la fácil comprensión del “qué debo conocer” y “cómo debo proceder” en cada fase de creación de un programa, permitiendo reducir la oposición entre lo teórico y lo práctico, entre lo formal y lo concreto, entre lo puro y lo aplicado.

Se usa el Mapa Conceptual como marco pedagógico para tratar de lograr el aprendizaje reflexivo de los Alumnos, es decir, el aprendizaje que gira en torno al pensamiento. Se pretende que los Alumnos aprendan reflexionando sobre lo que aprenden. [5]

Conclusiones

Se propuso el ensayo de una nueva alternativa para la Enseñanza de Computación en Carreras de Ingeniería que podría calificarse como una propuesta educativa “ágil” por la similitud que presenta con el desarrollo ágil de software (en contraposición al desarrollo tradicional o no ágil), el cual se caracteriza por ser incremental, cooperativo, sencillo y adaptativo. La filosofía “ágil” se resume en el "Manifiesto Ágil" (Utah, Febrero de 2001).

En trabajos futuros se planea profundizar en la metodología de enseñanza y aprendizaje propuesta, por ejemplo, exponiendo cómo se presenta a los alumnos el organizador previo en la primera clase expositiva, extendiendo la construcción del mapa conceptual para la enseñanza de la arquitectura de computadoras, detallando el desarrollo de cada nueva subunidad temática mediante la aplicación de la estrategia, explicando las modalidades de clases para el aprendizaje de los alumnos, presentando la propuesta evaluativa, explorando cómo el saber hacer del experto (Docente) puede contribuir al saber hacer del novato (Alumno) para el descubrimiento del algoritmo, indagando acerca del desarrollo de nuevas formas de enseñar para promover la motivación intrínseca de los alumnos o analizando algunos heurísticos solucionadores de problemas de Polya para la resolución de problemas con la computadora.

Bibliografía

- [1] Grossi, M. D., Jiménez Rey, E., Servetto, A., Perichinsky, G. 2003. *Enseñanza de Computación en Carreras de Ingeniería*. Actas IX Congreso Argentino de Ciencias de la Computación (pp 1252-1263). Universidad Nacional de La Plata.
- [2] Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J. 2002. *Agile software development methods. Review and analysis*. Espoo 2002. VTT Publications 478. Finland.
www.agilealliance.com/articles/index
- [3] Alda, F. L., Hernández, M. D. 1998. *Resolución de problemas*. En: “Cuadernos de Pedagogía” (Nº. 265, pp 28-32). FONTALBA.
- [4] Brookshear, J. G. 1995. *Introducción a las ciencias de la COMPUTACIÓN*. ADDISON-WESLEY IBEROAMERICANA.
- [5] Perkins, D. 1992. *La Escuela Inteligente: Del adiestramiento de la Memoria a la Educación de la Mente*. GEDISA.
- [6] Nickerson, R., Perkins, D., Smith, E. 1987. *Enseñar a pensar: Aspectos de la aptitud intelectual*. PAIDOS.
- [7] Joyce, B., Weil, M. 1985. *Modelos de Enseñanza*. En: “Los organizadores previos: mejorar la eficacia del estudio y de otros modos de presentación de información” (capítulo 5, pp 89-107). ANAYA.
- [8] Krick, E. V. 1980. *Introducción a la ingeniería y al diseño en ingeniería*. LIMUSA.