

Estrategias para la enseñanza de Algorítmica y Programación

Zulema Beatriz Rosanigo
brosanigo@infovia.com.ar

Alicia Beatriz Paur
apaur@ar.inter.net

Facultad de Ingeniería – UNPSJB
Roca 115 – 2do. Piso – Trelew – Te-Fax: 02965 42 84 02

Resumen

El objetivo de este trabajo es compartir las estrategias que nos dieron buen resultado para salvar algunas de las dificultades que encontramos en el proceso de aprendizaje de los alumnos ingresantes para los conceptos básicos de algorítmica y programación.

Palabras claves: programación – algoritmo – enseñanza problemática

Introducción

En todas las carreras de informática existe alguna materia de programación en el primer año, en nuestro caso esta primer materia es Algorítmica y Programación que se dicta en el segundo cuatrimestre, luego de Análisis Matemático I, Álgebra y Geometría e Introducción a la Computación. Tal como acontece en la mayoría de las primeras materias de una carrera universitaria, confluyen diferentes situaciones que dificultan el aprendizaje:

- Falta de conciencia estudiantil
- Desinterés por sus estudios en general
- Apatía por la materia en particular
- Dificultades para interpretar consignas
- Dificultades para expresarse
- Carencia de hábitos de lectura
- Carencia de disciplina para estudiar
- Escasa retención de los conocimientos adquiridos
- Bajo grado de reflexión, de independencia y de generalización.

Estas situaciones sumadas a las inherentes a la asignatura provocan alta deserción y bajo rendimiento que se refleja en una relación aprobados / inscriptos muy baja.

Algorítmica y Programación es una materia de vital importancia en la formación del alumno, el éxito o fracaso en ella influye decisivamente en las restantes, ya que no quedan casi posibilidades de seguir avanzando en la carrera si no se logran las competencias requeridas. Por lo tanto es imperativo buscar las mejores estrategias que ayuden a efectivizar el proceso de enseñanza-aprendizaje y la calidad de los resultados.

En este sentido, a lo largo de muchos años venimos observando, investigando, proponiendo y probando estrategias que den mejor resultado para la enseñanza-aprendizaje de la programación. Estas estrategias abarcan desde la forma didáctica de impartir los conocimientos hasta la distribución de las horas de clase.

Marco teórico

En este apartado daremos el marco teórico de las metodologías aplicadas en nuestra cátedra, que por sus características, y luego de en un proceso de refinamiento, selección y experimentación, consideramos que son las óptimas.

Enseñanza basada en problemas

La enseñanza basada en problemas consiste en el planteo y solución de problemas en cuya resolución se produce el aprendizaje. Es un método de enseñanza donde los alumnos se sitúan sistemáticamente ante problemas, cuya resolución debe realizarse con su activa participación, y en el que el objetivo no es sólo la obtención del resultado, sino además su capacitación para la resolución independiente de problemas en general. Se fundamenta en el contexto del constructivismo, mediante el cual el estudiante busca y selecciona la información, razona e integra los conocimientos previos y adquiridos, dando finalmente una solución al problema planteado, tal y como se va a enfrentar en su actividad profesional.

No es lo mismo un ejercicio que un problema. Mientras que en el ejercicio se aplica un método o fórmula conocida, en el problema en cambio, no resulta evidente el camino a seguir, requiere explorar y relacionar con los conocimientos previos. Para que un enunciado de un ejercicio sea considerado un “problema”, debe provocar adquisición y transferencia de conocimientos. Se da un verdadero problema cuando desde una situación quiero llegar a otra y no conozco el camino que me puede llevar de una a otra.

Los conocimientos previos y la actividad constituyen los pilares fundamentales del aprendizaje. El conocimiento no se transfiere de forma directa, el aprendizaje requiere la participación activa del alumno, provocando el “autodescubrimiento”.

Desde el punto de vista didáctico, en la selección de problemas a proponer a los estudiantes, debe tenerse en cuenta no solo la naturaleza de la tarea, sino también los conocimientos que la persona requiere para su solución y las motivaciones para realizarla.

Enseñanza problémica

Es un método de enseñanza basado en problemas que hace hincapié en provocar una situación problémica. Según Portuondo Padrón (2000), la *situación problémica* es un estado psíquico de dificultad que surge en el alumno cuando, en la tarea que está resolviendo, no puede explicar un hecho nuevo mediante los conocimientos que posee, al aplicarlos consecuentemente provocan una contradicción. Está caracterizada por:

- existencia de algo nuevo en la actividad intelectual
- tendencia a la motivación en el sujeto del aprendizaje

- presencia de la contradicción (contradicción entre los conocimientos nuevos a asimilar y los ya asimilados o contradicción entre los conocimientos nuevos o existentes y la propia vida).

Los métodos de enseñanza problémica más utilizados son:

- La *exposición problémica*: la esencia de este método radica en que el profesor, al transmitir los conocimientos, crea una situación problémica y muestra la vía para solucionar determinado problema, muestra la veracidad de los datos, “descubre” las contradicciones presentes en la situación y “descubre” la lógica del razonamiento para solucionar el problema planteado.
- *Elaboración conjunta problémica*: está basado en la interacción profesor-alumno, de forma tal que el docente logre, a través de los alumnos, la situación problémica.
- *Búsqueda parcial o heurística*: se caracteriza porque el profesor organiza la participación de los estudiantes para la realización de determinadas tareas del proceso de investigación.
- *Investigativo*: permite relacionar al alumno con las técnicas y métodos propios de la actividad investigativa, desarrollando el pensamiento creador. Lo fundamental, en este método, es la actividad de búsqueda independiente de los estudiantes dirigida a la solución de un problema.
- *Por problemas*: los estudiantes se introducen en el proceso de búsqueda de solución de problemas nuevos para ellos, aplicando conocimientos ya asimilados y adquiriendo independientemente otros, lo que le permite obtener y desarrollar la actividad creadora.

Métodos participativos

Los métodos y técnicas participativas se definen como las vías, procedimientos y medios sistematizados de organización y desarrollo de la actividad del grupo de estudiantes, sobre la base de concepciones no tradicionales de la enseñanza, con el fin de lograr el aprovechamiento óptimo de sus posibilidades cognoscitivas y afectivas. Algunos de estos métodos y técnicas son, por ejemplo, la discusión en grupos pequeños, el juego de roles, la mesa redonda, la técnica de la rejilla, el panel, y otros que tienen una utilización cada vez más amplia en la enseñanza.

Según Cueto (1985), el aprendizaje grupal implica ubicar al docente y al estudiante como seres sociales, integrantes de grupos, buscar el abordaje y la transformación del conocimiento desde una perspectiva de grupo, valorar la importancia de aprender a interaccionar en grupo y a vincularse con los otros, aceptar que aprender es elaborar el conocimiento, ya que éste no está dado ni acabado; implica, igualmente, considerar que la interacción y el grupo son medio y fuente de experiencias para el sujeto, que posibilitan el aprendizaje, reconocer la importancia de la comunicación y de la dialéctica en las modificaciones sujeto grupo.

El método así concebido ha de garantizar, además, el desarrollo no solo de conocimientos y estrategias cognoscitivas en el estudiante, sino también de importantes cualidades de trabajo en grupo; fomentar su interés y motivación por el conocimiento y por su propio desarrollo como personalidad, como sujeto de su actividad social.

Enfoque metodológico adoptado

En los primeros años, el énfasis en la enseñanza estuvo puesto en transmitir conceptos y procedimientos algorítmicos y en realizar prácticas para aplicar esos conceptos. Luego fuimos recurriendo a métodos más participativos.

Desde hace varios años estamos aplicando un entrelazamiento de las distintas formas que puede darse la enseñanza problémica, la enseñanza basada en problemas y los métodos participativos.

Las clases son teórico- prácticas aunque formalmente están divididas en teorías y prácticas: En las clases prácticas, los alumnos trabajan solos o en grupo sobre una guía de ejercicios propuestos, con dificultades diversas, y los docentes actúan de guías, decidiendo cuándo intervenir y qué sugerencias dar si ven que hay un bloqueo o lo notan muy perdido. En general, de cada práctico, se toma algún ejercicio para construir entre todos la solución, o para analizar y comparar las de distintos alumnos. Se busca crear espacios de reflexión acerca de las dificultades que plantea cada problema y las estrategias de resolución empleadas.

Las actividades en laboratorio comienzan a mitad de cuatrimestre una vez adquirida suficiente habilidad para la resolución de problemas algorítmicos. Primeramente se da una clase guiada en el ambiente de desarrollo del lenguaje de programación adoptado, buscando familiarizarlo con el lenguaje y las herramientas de edición, compilación y depuración. Debido al número de alumnos y a los recursos disponibles, se trabaja en subgrupos en diferentes horarios. Luego se deja que sea el alumno quien experimente y resuelva los problemas, supervisando y guiándolo en lo que necesite.

En las clases de “teoría” se introducen los conceptos nuevos siguiendo alguna de las formas descritas en los párrafos siguientes, y luego se plantean diferentes variantes de problemas que aplican el concepto y lo relacionan con otros, construyendo la solución entre todos.

Determinados temas se plantean como una situación problemática en que hay una contradicción entre lo que se espera, de acuerdo a los conceptos conocidos y a lo que acontece. Por ejemplo:

$$\text{¿Es } 4 * 0.1 = 0.1 + 0.1 + 0.1 + 0.1?$$

Genéricamente, dado un $r \in \mathbb{R}$ y $n \in \mathbb{N}$, podemos asegurar la siguiente igualdad?

$$n * r = \underbrace{r + r + \dots + r}_n = \sum r$$

La contradicción surge porque matemáticamente es lo mismo, pero computacionalmente no, debido al error de redondeo propio de la implementación de los números reales.

En otros casos se aborda un tema a partir de un problema cercano a la vida cotidiana o profesional que constituyan de alguna manera un desafío.

También hay casos en que la discusión de un nuevo tópico consiste en una introducción teórica, presentando el concepto y dando las definiciones que se requieran, seguida de numerosos ejemplos comentados y una amplia práctica de aplicación y de corroboración de resultados esperados o no.

Algunas clases de “teoría” se dedican a resolver en forma conjunta un enunciado de problema planteado y elegido por los alumnos. Los alumnos realizan sus aproximaciones hacia la solución y el docente analiza dinámicamente si está bien o no, permitiendo las equivocaciones para aprender desde el error. Si el grado de error es muy grosero, se insta con preguntas a que reflexionen sobre la propuesta antes de seguir avanzando. Si el error es fácilmente corregible mediante un refinamiento, se lo acepta para luego corregirlo en el refinamiento. Son clases muy ricas, porque el actor principal es el alumno, quien además debe defender su propuesta frente a otras de sus compañeros.

La distribución de los temas de la materia determina también una forma de enseñanza cíclica en la que los mismos conceptos se abordan de forma recurrente a lo largo del curso, reafirmando, completando, relacionando, aplicándolo. Muchos conceptos se aplican primero intuitivamente o con una explicación simple de lo que se trata pero suficiente como para poder aplicarlo, y luego, en una etapa posterior se completa y formaliza. Tal es el caso de descomposición de problemas y refinamiento sucesivos.

Elección de la forma de representación el algoritmo

Es importante que el alumno desarrolle capacidades que le permitan encarar la resolución de un problema de manera eficaz e independiente del lenguaje de programación. Si bien se comienza a resolver problemas simples, luego serán cada vez más complejos. De los distintos mecanismos para representar algoritmos, hemos adoptado el pseudocódigo, un lenguaje algorítmico menos rígido que un lenguaje de programación, que permite centrar la atención en la resolución del problema y no en los detalles propios de la rigidez de un lenguaje de programación.

El pseudocódigo es una notación algorítmica textual caracterizada por:

- Permitir la declaración de los datos (constantes y/o variables) manipulados por el algoritmo.
- Disponer de un conjunto pequeño y completo de palabras reservadas que permitan expresar: las acciones elementales, las primitivas de composición de acciones y la definición de acciones con nombre (funciones y procedimientos).

Algunos critican al pseudocódigo porque lo consideran como la traducción de un lenguaje de programación al castellano, más precisamente lo llaman “Pascal en castellano”. Aunque algunas formas de pseudocódigo pueden llegar a serlo, en general no es así. Esta similitud con Pascal se da porque Pascal fue un lenguaje pensado para enseñar y por lo tanto tiene características útiles para expresar con naturalidad la solución.

Nosotros definimos y usamos un lenguaje algorítmico que respeta las construcciones de la programación estructura con las características sintácticas más favorables para la legibilidad de los programas, como por ejemplo: simbología diferente para cosas diferentes, estructuras de control con terminador de estructura, etc.

La elección del lenguaje de programación

Aunque el propósito principal es que el estudiante **aprenda a programar**¹, es conveniente que pueda **codificar**² los algoritmos en algún lenguaje de programación e implementarlo utilizando la herramienta de desarrollo disponible y que se considere adecuada.

¿Qué lenguaje utilizar? Obviamente deberá responder al paradigma de programación que se utiliza para enseñar. En nuestro caso, hemos adoptado el lenguaje PASCAL, por ser un lenguaje simple y que respeta las reglas de programación estructurada, fácil de utilizar y aprender. Para el tipo de complejidad de los problemas que se resuelven en primer año, la representación en este lenguaje es casi natural, cuenta con suficientes construcciones básicas, tipos de datos y constructores de tipo que permiten representar el modelo naturalmente. Existen varios entornos de programación para este lenguaje que facilitan las tareas de edición, compilación y depurado, cuidando mucho al programador contra sus propios errores y omisiones.

A pedido de docentes de cursos superiores, también hemos analizado la posibilidad de utilizar el lenguaje C, pero a pesar de los argumentos atendibles de su parte, no consideramos conveniente su utilización en un curso formativo en la programación por varias razones, pero fundamentalmente porque ciertos conceptos que se están aprendiendo quedarían distorsionados:

¹ **Programar** es un proceso mental complejo dividido en varias etapas. Su finalidad es comprender claramente el problema que se va a simular o resolver mediante la computadora, y entender también con detalle cuál será el procedimiento mediante el cual la máquina llegará a la solución deseada

² La **codificación** es una etapa posterior a la programación y consiste en escribir en un lenguaje de programación la solución ya encontrada mediante la programación.

- El concepto de función sin efecto lateral es factible y natural hacerla respetar en Pascal pero es antinatural querer respetarlo en C
- Los conceptos de comunicación entre subprogramas a través de parámetros de entrada, salida o entrada-salida, obligan a trabajar anticipadamente con punteros, con todo lo que esto implica. En Pascal se puede hacer el paralelo con el pseudocódigo “los parámetros de salida y de entrada-salida se especifican en la cabecera con el prefijo var, mientras que los de entrada no requieren ninguna aclaración”. Querer establecer un paralelo entre el concepto y la implementación en C, obligaría a modificar la llamada, la cabecera y el cuerpo de la función.
- El manejo de la entrada-salida y de los archivos es más complejo y menos natural que en Pascal.
- La forma natural de tratar un problema con posibles condiciones de error en C, es tratar primero los casos erróneos para retornar enseguida si se diera esa situación, para luego quedarse con el caso normal. Pero no es la forma natural en que se piensa la solución del problema. Para estos caso, la codificación en C exigiría reformular la solución, o de lo contrario, enseñar la programación basada en el lenguaje.

Otro lenguaje de programación analizado fue ADA, que tiene ventajas similares a las de Pascal pero no es tan simple como él.

La característica más rechazada por el alumnado del uso de lenguajes de este tipo es la interfaz con el usuario que les resulta “antigua”, puramente textual. Estamos analizando la conveniencia o no de utilizar entornos con interfaz gráfica.

Principales errores encontrados en la resolución de problemas

Muchos errores de programación pueden evitarse si conocemos y atacamos las razones que lo motivan.

Los errores más comunes que cometen los alumnos en la resolución de problemas son:

- Confunde el uso de los operadores lógicos, utiliza conjunción en lugar de disyunción o viceversa, o no reconoce bien cual corresponde para el problema.
- Errores sintácticos: Utiliza primitivas o variables no definidas, utiliza variables sin inicializar, hace asignaciones o comparaciones inválidas.
- Mal uso de la entrada-salida: Confunde mostrar un mensaje para aclarar al usuario lo que se espera que entre con la operación efectiva de entrada, confunde validar con mostrar un mensaje aclaratorio, no valida la entrada en los casos necesarios
- Errores en funciones y procedimientos: Define variables como parámetros de una función y en el cuerpo de la misma usa otras, no hay concordancia entre los parámetros formales y reales, ingresa en el cuerpo de un subprograma las variables que pasa como parámetros, confunde y mezcla los conceptos de función y procedimientos, confunde la invocación con la declaración.
- Errores en las estructuras de control: Ciclos sin fin, no avanza nunca dentro del ciclo, no inicializa variables utilizadas como contadores o acumuladores, no controla cuando la secuencia está vacía al comenzar el ciclo, selecciona mal la estructura de control más adecuada al problema, plantea mal la condición en la estructura, utiliza un condicional en lugar de un mientras o viceversa.

- Mal manejo de los índices en arreglos: Accede a posiciones no definidas, confunde o no respeta el orden de los índices en arreglos de más de una dimensión, confunde arreglo y registro
- Errores en la solución: No muestra los resultados obtenidos, no resuelve el problema planteado. mala interpretación del enunciado, mal planteo del problema, falta casos de tratar, establece la solución general pero no la refina, resuelve puntos parciales pero no arma la solución global

Estrategias adoptadas

Distribución horaria: Al principio dábamos módulos de cuatro horas dos de teoría y dos de práctica en dos días, pero los alumnos se retiraban antes y tampoco ejercitaban fuera de clase. Desde hace cuatro años, distribuimos las ocho horas de clases semanales en cuatro días de dos horas, quedando clases teórico-prácticas alternadas con clases de ejercitación, y logrando mayor participación de los alumnos y menos “olvidos” de una semana para la otra.

Guía de trabajos prácticos: Desde hace unos cinco años, se introdujo un práctico inicial que se puede hacer sin haber comenzado a ver conceptos propios de la materia, pero que lo ejercita para la resolución de problemas: situaciones de sentido común, de interpretación de enunciados como por ejemplo analizar situaciones acerca de las reglas de un juego, de relaciones lógicas y matemáticas, identificar la incógnita, los datos y restricciones, identificación de estructuras de control que aparecen en el lenguaje natural, expresiones con conectivos lógicos o matemáticos, enunciados del tipo “al menos uno”, “mayor que 2 pero no tan grande como 15”, entretenimientos y acertijos, etc.

La resto de la guía incluye una variedad de problemas que cubren todos los temas y en los que se plantean problemas similares con diferente forma de enunciado, enunciados similares pero con connotaciones diferentes, varias variantes de un mismo problema, problemas a resolver, problemas para reconocer la o las soluciones correctas, para analizar cual es mejor, buscar errores, enunciados de situaciones bastante cercanas a la realidad con diferente grado de dificultad. En general los enunciados no explicitan las estructuras de datos que requieren sino que debe “descubrirlos”.

El objetivo de esta variedad de enunciados es que resuelva situaciones diferentes, preste atención a cada palabra del enunciado y se familiarice con distintas maneras de expresar en lenguaje natural lo que en lenguaje algorítmico podría representarse de una misma forma.

Evaluaciones: Se evalúa a través del desarrollo de trabajos prácticos y parciales. Las evaluaciones contemplan de manera integrada la adquisición de conocimientos, la formación de actitudes, el desarrollo de la capacidad de análisis, habilidades para encontrar la información y resolver problemas reales, expresión oral y escrita.

Los trabajos prácticos obligatorios consisten fundamentalmente en resolución de problemas algorítmicos algunos de los cuales lo codificarán, desarrollando programas específicos.

El primer parcial es conceptual y evalúa la adquisición de los conceptos básicos para la formulación de algoritmos. Se requiere de su aprobación y no se califica con nota numérica. En el segundo y tercer parcial se evalúa resolución de problemas mediante formulación de algoritmos en los que debe aplicar estructuras de datos y de control adecuadas; se califica con nota numérica.

La incorporación del parcial conceptual antes de las evaluaciones que incluyen desarrollar algoritmos ha permitido mejorar mucho los resultados de ellas: los alumnos afianzan los conceptos que utilizarán para resolver los problemas y hacen un primer parcial sin el temor de la “hoja en blanco” que se produce cuando aún no se tiene práctica suficiente en la resolución algorítmica.

Relativas a los temas: Se trabaja con los conceptos de función y procedimiento desde los primeros algoritmos: Al principio, cuando aún no se ha dado el tema y se detecta que sería útil contar con una cierta operación que no es una primitiva para el procesador, “imaginamos” que la posee y especificamos qué hace y cómo nos comunicamos: qué recibe, qué devuelve, le damos un nombre y la utilizamos respetando la especificación acordada. Esto ha contribuido a disminuir los errores que se encontraban relativos a la invocación y definición, a afianzarse en el uso de funciones y procedimientos, a comenzar a modularizar en forma natural.

La especificación con precondiciones y poscondiciones de las funciones y procedimientos, y la comunicación conceptual de parámetros de entrada, entrada-salida y salida también ha ayudado a mejorar los errores que se encontraban relativos al tema.

El dedicarle mayor tiempo al análisis del problema, de los datos de entrada y las condiciones que deben cumplir y la definición de operaciones especiales para su validación ha mejorado su visión para resolver el problema y disminuido los errores frecuentes que ocurrían sobre el tema.

Conclusiones

La enseñanza a través de la resolución de problemas es actualmente el método más invocado para poner en práctica el principio general de aprendizaje activo. Lo que en el fondo se persigue con ella es transmitir en lo posible de una manera sistemática los procesos de pensamiento eficaces en la resolución de verdaderos problemas.

Una metodología para la enseñanza basada en problemas durante el proceso docente educativo será funcional en la medida que se tenga en cuenta los otros elementos del proceso: objetivos, contenidos, métodos, medios y formas organizativas.

Aprender a programar es una tarea compleja pero esencial tanto para la adquisición de conocimiento como el desarrollo de las capacidades de pensamiento.

Se han enumerado un conjunto de estrategias que nos están dando buen resultado y esperamos que puedan ser útiles para otras personas.

Bibliografía

CUETO DEL A.M. y A.M. FDEZ (1985). "El dispositivo grupal". Ediciones Búsqueda. Buenos Aires, 1985.

MAJMUTOV, M.I. (1984) “*La Enseñanza Problemática*”. La Habana. Cuba. Editorial Pueblo y Educación. 1984.

PORTUONDO PADRÓN, Roberto, VÁZQUEZ CEDEÑO, Rosa. (2000) “*Algunos aspectos de la enseñanza problemática*” Apuntes del Seminario de postgrado “Metodología de la enseñanza de la Ingeniería y la Tecnología”

RIZO, Cabrera Celia y Luis Campistrous: Didáctica y Solución de Problemas, Edición Especial , II Congreso Internacional Didáctica de las Ciencias, 2002.

ROSANIGO, Zulema Beatriz (2005) Algorítmica y Programación – Notas de cátedra.

RUEDA, Sonia V., GARCÍA Alejandro J. (2005) “Análisis y Comprensión de Problemas Curso de nivelación para ingresantes a carreras de Ciencias e Ingeniería de la Computación” JEITICS 2005 - Primeras Jornadas de Educación en Informática y TICS en Argentina