

## ENSEÑANZA DE COMPUTACION EN CARRERAS DE INGENIERIA

### **Autores**

- GROSSI, María Delia ([mdgrossi@mara.fi.uba.ar](mailto:mdgrossi@mara.fi.uba.ar))
- JIMÉNEZ REY, M. Elizabeth ([ejimenez@mara.fi.uba.ar](mailto:ejimenez@mara.fi.uba.ar))
- SERVETTO, Arturo Carlos ([aserve@mara.fi.uba.ar](mailto:aserve@mara.fi.uba.ar))
- PERICHINSKY, Gregorio ([gperi@mara.fi.uba.ar](mailto:gperi@mara.fi.uba.ar))

*Paseo Colón N° 850, 4° Piso, Tel. 4343-0891 Int. 142*

*Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires*

### **Resumen**

Este trabajo propone una estrategia para la enseñanza de Computación, asignatura de formación básica obligatoria para todos los alumnos de las carreras de Ingeniería (excepto Ingeniería en Informática e Ingeniería Electrónica) sobre la base de nuestra experiencia docente.

Se describen las condiciones de enseñanza para ejemplificar los problemas que debemos resolver, se presenta la metodología utilizada para optimizar el aprendizaje de los alumnos, se expone la forma de evaluación aplicada y los resultados que se han conseguido y se pretende conseguir. Se fundamenta la propuesta desde la perspectiva educativa de mejorar la calidad de la enseñanza.

### **Palabras clave**

Informática Educativa. Enseñanza de Algoritmia y Programación.

## **Introducción**

El objetivo de este trabajo es comunicar nuestra experiencia en la 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, y proponer una estrategia para el mejoramiento y la eficientización de la práctica de la enseñanza que tenga en consideración cómo aprende el alumno.

“La enseñanza es una situación social que como tal se encuentra sometida a las variaciones de las interacciones entre los participantes así como a las presiones exteriores y a las definiciones institucionales de los roles” [3].

### *Caracterización de la Materia*

Sus objetivos son:

- Brindar al alumno una visión global de la Computación de manera que comprenda el aspecto científico de la actual sociedad informatizada y conozca herramientas y vocabulario de profesionales de informática de manera tal de poder interactuar con ellos interdisciplinariamente [5].
- Lograr que el alumno se compenetre con las tecnologías y herramientas fundamentales de la Computación de manera que aprenda a usar la computadora como herramienta de trabajo, conociendo su precisión, capacidad y limitaciones.
- 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.
- 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.

Y sus contenidos:

Alcance de las ciencias de la Computación. Técnicas para representar y almacenar información y forma en que las máquinas digitales manipulan los datos. Arquitectura de computadoras. Software de Sistema, de Aplicación y de Traducción. Lenguajes de Programación. Algoritmia y Programación básicas.

En la institución esta materia es de formación básica obligatoria.

### *Caracterización de Alumnos*

Los alumnos que cursan la materia pertenecen a distintas carreras de Ingeniería, y pueden hacerlo en distintas etapas de avance en sus planes de estudio. Algunos alumnos provienen de escuelas técnicas y con conocimientos y experiencia previa en programación, otros poseen experiencia en la operación de computadoras para uso personal.

### *Condicionantes*

La relación docente alumno al inicio de los cursos alcanza a un docente auxiliar cada treinta alumnos, y el índice de deserción llega generalmente al treinta por ciento.

La materia es cuatrimestral, y tiene una carga horaria de cuatro horas por semana

### *Problema*

La coexistencia de alumnos de distintas carreras y niveles de formación dificulta el abordaje y elaboración de contenidos por parte de los docentes, así como la constitución de grupos de trabajo por parte de los alumnos.

La algoritmia representa para los alumnos 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.

### **Antecedentes**

En la enseñanza de la materia durante catorce cuatrimestres se buscó permanentemente mejorar el método de enseñanza, la organización y la selección de contenidos para conseguir un mejor aprendizaje por parte de los alumnos.

En principio, se trató que el alumno se compenetrara en las aplicaciones de mercado horizontal (procesamiento de texto, planillas de cálculo, manejo de bases de datos de oficina) así como en la arquitectura de la computadora, el software de sistema y la programación básica de manera que el alumno aprendiera a emplearla como herramienta de trabajo, y conociera sus alcances y limitaciones. También se presentaban los conceptos fundamentales de los campos disciplinares de la Informática para que los alumnos se familiarizaran con las técnicas y vocabulario de la disciplina, y en su futuro profesional pudieran interactuar con profesionales informáticos sin inconvenientes.

Pero al ser la materia cuatrimestral y de cuatro créditos (cuatro horas de clase por semana) la enseñanza de programación quedaba acotada a no más de cuatro clases, que no resultaban suficientes para que los alumnos aprehendieran conceptos básicos y desarrollaran la habilidad de programar. Como muchos alumnos venían ya con algún manejo de software de aplicación de mercado horizontal y el Centro de Estudiantes ofrecía y ofrece distintos cursos de operación de software de este tipo, se modificó el contenido y la planificación de la materia suprimiéndose la aproximación al software de mercado horizontal y se extendió el tiempo y profundizó la práctica de programación. Así fue como siguiendo un enfoque lineal se trató de que los alumnos se aproximaran a la informática partiendo de un concepto fundamental: la algoritmia. Para presentar estructuras de control y plantear esquemas de solución se empleaba el pseudocódigo, y luego se codificaba en Pascal.

En virtud del tiempo disponible para la enseñanza, se vio la necesidad de optimizar el tiempo destinado a las prácticas de programación, produciendo un nuevo cambio. Por un lado, en la forma de abordar los contenidos: se descartó el pseudocódigo para comenzar directamente con Pascal, haciéndose mucho énfasis en el método de refinamientos sucesivos, y por otro lado, en la integración de las clases teórico-prácticas: las clases teóricas se desarrollan en forma interrelacionada con las clases prácticas. En las clases teóricas los conceptos relativos al área de conocimiento se enseñan en el orden en que fueron concibiéndose en el tiempo y se los contextualiza con los problemas tecnológicos de contorno. Para ello, se presentan fundamentos teóricos y se plantean problemas que se resuelven en clase en forma grupal o individual.

## **Enseñanza**

### *Acerca de los Contenidos*

La institución, a través del Programa Sintético establece el contenido mínimo a ser enseñado.

A partir de ese contenido mínimo, como estrategia de organización, se deben tomar ciertas decisiones relativas a cuáles son las estrategias oportunas para conseguir organizar globalmente la enseñanza de la materia, de tal modo que los alumnos puedan ir adquiriendo el conocimiento del modo más eficaz posible [2].

La selección del contenido se realiza privilegiando los conceptos y técnicas de campos de la disciplina que contribuyan al logro de los objetivos de la materia.

La enseñanza de algoritmia y programación comprende el descubrimiento de algoritmos como proceso de resolución de problemas y la representación de los algoritmos para que puedan ser comunicados a una computadora. Se considera que los alumnos pueden incorporar mucho mejor nociones, por ejemplo, sobre traducción e interpretación de programas si tienen previamente una base de conocimiento de cómo es el lenguaje de máquina, o conceptos de variable y constante, si ya saben cómo se representan los datos en una computadora, o cómo procesar archivos si tienen nociones básicas de arquitectura de computadoras y sistemas operativos [1]. Estos conocimientos los ilustran acerca de las bases sobre las cuales deben trabajar y los concientiza del potencial de la computadora como herramienta de trabajo y de las limitaciones del software que produzcan, además de cumplir el objetivo de familiarizarlos con conceptos y técnicas de la disciplina que en su futuro profesional los habilitaría a interactuar interdisciplinariamente con profesionales en informática sin problemas de comunicación.

### *Unidades de Programa*

- El algoritmo como objeto de a Computación. Alcance de la disciplina. El software y sus categorías. Desarrollo de programas. Modelo de un programa tipo.
- Capacidades de computadoras y componentes arquitectónicos asociados.
- Primitivas de programación en Pascal: de documentación, de asignación, de entrada y salida de datos, expresiones, estructuras de control selectivas, repetitivas y de invocación de subprogramas. Subprogramas del lenguaje.
- Ambientes integrados de desarrollo de software. Escritura de un programa en lenguaje Pascal. La memoria central de una computadora y los tipos de datos básicos o primitivos.
- La unidad central de procesamiento. Instrucciones en lenguaje de máquina. Ejecución de programas. Comunicación con periféricos.
- Organización de dispositivos de almacenamiento masivo en una computadora. Archivos de texto.
- El sistema operativo, funciones, interfaces con usuarios. Redes y software de comunicación.
- Acoplamiento de subprogramas. Uso de parámetros. Principios de modularización de programas.
- Tipos de datos estructurados, restricciones de dominio (subrangos), registros y arreglos. Búsqueda y ordenamiento de elementos en arreglos.

## *Acerca de la Secuenciación*

La secuenciación del contenido de la asignatura es el segundo problema a resolver debido a la gran importancia que tiene en el proceso de aprendizaje de los alumnos. Teniendo en cuenta el principio elemental que establece que el conocimiento sólo se puede adquirir sobre la base de las estructuras de conocimiento previamente poseídas, se escalona la modificación de dichas estructuras para que progresivamente los alumnos puedan ir incorporando el conocimiento [2]. Las subunidades temáticas se desarrollan de acuerdo a un cronograma, teniendo en cuenta las siguientes premisas:

- Se presentan las subunidades temáticas en un continuo que va de lo simple a lo complejo o de lo general a lo particular [2].
- “El continuo de lo simple a lo complejo se forma añadiendo partes de ideas”. Se parte de una idea simple, la algoritmia como método de resolución de problemas, porque tiene menos elementos y requiere estructuras de conocimiento menos desarrolladas, y se incorporan progresivamente ideas cada vez más complejas que van enriqueciendo dichas estructuras, y la idea va complejizándose. El continuo de lo general a lo particular se genera con la subdivisión de ideas. Esta subdivisión hace que una idea más detallada, como la de primitiva en Pascal, se aplique a pocos objetos o acontecimientos, mientras que un concepto genérico, como el de problema a resolver representado por el enunciado, se aplica a más objetos o acontecimientos. Un concepto general (primera descomposición o refinamiento del problema a resolver en Prólogo, Resolución y Epílogo) posee menos atributos que un concepto particular (primitiva en Pascal, último nivel de refinamiento), lo general es más simple que lo particular ya que requiere menos y más sencillas discriminaciones. El método tradicional para desarrollar algoritmos es el denominado "divide y vencerás", que consiste en descomponer el problema principal en 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 expresadas por primitivas de programación (refinamientos sucesivos o diseño descendente -porque parte de ideas generales y desciende hasta los detalles).

“La secuencia de lo simple a lo complejo hace que el alumno tenga siempre presente el concepto y el sentido de las ideas que intenta aprender, le permite aprender al nivel de complejidad mas apropiado y más significativo en cada estadio de aprendizaje y se evita, en gran medida, la necesidad de recurrir a prerrequisitos de aprendizaje que muchas veces tienen que presentarse a un nivel de complejidad inapropiado y poco significativo para el alumno” [2].

- Las ideas iniciales, las más simples o generales, son un compendio de las ideas siguientes (sección declarativa: recursos, sección algorítmica: herramientas que manipulan recursos para obtener la solución). Se presenta “un pequeño número de ideas fundamentales y representativas a través de las cuales se transmite la esencia de todo el contenido; las ideas se eligen de tal modo que el resto del contenido del curso proporcione más detalles o más conocimiento acerca de ellas” [2].
- Los sucesivos desarrollos se realizan sobre la base de un único tipo de contenido. La dimensión que sirve de base a la organización del contenido del curso es conceptual. Los conceptos constituyen el tipo de contenido estructurable utilizado para organizar el curso. “Los conceptos son clasificaciones arbitrarias de objetos, acontecimientos o símbolos que comparten características comunes. Conocer un concepto es ser capaz de seleccionar aquellos rasgos de un objeto o acontecimiento que desde la perspectiva de un campo de conocimientos, definen un conjunto de entidades. Saber un concepto implica no solo un conocimiento descriptivo de sus atributos sino también el dominio de ciertos procesos mentales que permiten identificar, reconocer o clasificar algo” (por ejemplo,

individualizar, aislar, y ordenar las distintas actividades para conseguir una solución). “Las estructuras conceptuales describen relaciones supraordenadas, subordinadas y coordinadas entre ideas. Una elaboración conceptual es ir añadiendo conceptos más detallados”. Se sigue un orden histórico en el campo de las Ciencias de la Computación como estrategia para secuenciar la enseñanza porque los principios más simples son los primeros a los que llega una ciencia y a partir de ellos se van descubriendo principios cada vez más complejos. La estructura organizante elegida permite secuenciar la enseñanza. Los desarrollos se van desdoblado en niveles de igual complejidad. Cada nivel puede tener varias subunidades temáticas pero sólo tras haber completado un nivel se puede pasar al siguiente [2].

### *Cronograma*

<b>Clase</b>	<b>Teoría</b>	<b>Práctica</b>
<b>1</b>	Algoritmo y Programa. Categorías del Software. Desarrollo de Programas: (a) Análisis y Documentación del Problema a Resolver; (b) Definición de Recursos y Esquematización, Refinamientos y Prueba del Algoritmo; (c) Formalización del Programa, y (d) Depuración y Prueba del Programa. Modelo de un Programa Tipo (Prólogo-Resolución-Epílogo).	Estructura de un programa en lenguaje Pascal. Recursos Básicos de un Programa en Lenguaje Pascal: Constantes y Variables Enteras, Reales y de Caracteres. Enunciados de Documentación Interna. Instrucciones de Asignación, de Lectura desde Teclado, de Escritura en Pantalla y Estructuras de Selección Simple (IF-THEN, IF-THEN-ELSE).
<b>2</b>	Arquitectura Básica de una Computadora. Organización de la Memoria Central. Sistema de Numeración Binario. Unidades de Información en una Computadora: Órdenes, Valores Enteros, Valores Fraccionarios y Símbolos.	Anidamiento de Selecciones Simples, y Selección Múltiple (CASE). Comparación y Equivalencia. Especificación del Primer Trabajo Práctico
<b>3</b>	Representación de Valores Enteros: Complemento a Dos.	Estructuras de Repetición con Control Inicial (WHILE) y con Control Final (REPEAT). Comparación y Equivalencia.
<b>4</b>	Representación de Valores Fraccionarios: Punto Flotante y Representación de Valores Enteros en Exceso.	Condiciones Compuestas de Control de Repetición. Lógica Binaria.
<b>5</b>	Representación de Símbolos: ASCII. Numeración Hexadecimal como Abstracción de la Binaria.	Entrega del Primer Trabajo Práctico.
<b>6</b>	Órdenes en Lenguaje de Máquina y Almacenamiento de Programas. PRIMER PARCIAL: Evaluación sobre Sistemas de Numeración y de Representación de Datos.	PRIMER PARCIAL: Evaluación sobre Estructuras de Control.

7	<p>Organización y Funcionamiento de la Unidad Central de Procesamiento (CPU). El Ciclo de Máquina. Mejoras de Rendimiento: Memoria Caché y Canalización.</p> <p>Devolución del Primer Parcial y discusión general sobre problemas conceptuales detectados.</p>	<p>Subprogramas como Recurso de Programación y como Estructuras de Control de Transferencia-Retorno. Acoplamiento Implícito (Variables Globales) y Explícito (Parámetros) de Subprogramas. Reglas de Alcance. Parámetros Formales y Reales. Parámetros por Valor y por Referencia. Procedimientos y Funciones.</p> <p>Devolución del Primer Parcial y Resolución de la evaluación en forma conjunta en el pizarrón</p>
8	<p>Comunicación con Periféricos: Puertos y Controladores. Acceso Directo a Memoria (DMA) y Buffers. Interrupciones y Entorno de Programación.</p>	<p>Modularización Cohesiva de Programas. Especificación del Segundo Trabajo Práctico.</p>
9	<p>Organización de Dispositivos de Almacenamiento Masivo. Discos Magnéticos, Ópticos y Magnetoópticos. Confiabilidad de la Información Almacenada: Detección y Corrección de Errores.</p> <p>SEGUNDO PARCIAL: Evaluación sobre Instrucciones y Funcionamiento de Máquinas.</p>	<p>EVALUACIÓN DIAGNÓSTICA sobre Subprogramas.</p>
10	<p>Categorías de Sistemas de Computación: Macro y Supercomputadoras, Minicomputadoras y Computadoras Personales. Tiempo Compartido y Terminales. Mono y Multiprocesamiento. Conexión Remota y Redes.</p> <p>Devolución del Segundo Parcial y Discusión general sobre problemas conceptuales detectados.</p>	<p>Archivos de Texto. Organización de Líneas y Caracteres de Control. Asignación de Archivos. Parametrización de Nombres de Archivo. Lectura de Secuencias de Datos desde un Archivo de Texto.. Devolución de la Evaluación Diagnóstica y Resolución de la evaluación en forma conjunta en el pizarrón</p>
11	<p>Sistemas Operativos: Interfaces (Por Comandos y GUI) y Recursos (De Hardware y de Software), Carga y Arranque, Administración de Recursos. Multiprogramación y Procesamiento Interactivo (o en Primer Plano) y en Segundo Plano.</p>	<p>Creación de Archivos de Texto y Escritura de Datos. Agregado de Datos a un Archivo de Texto Existente. Archivos como Parámetros. Búsqueda y Substitución de Patrones en Archivos de Texto.</p>
12	<p>Sistemas Operativos: Supervisión y Seguridad de Recursos. Accesorios del Sistema: Software de Comunicación y Herramientas de Software (Testeo y Desfragmentación de Discos, y Programación de Tareas).</p>	<p>Definición de Datos (TYPE): Restricciones de Dominio de Variables Enteras (Rangos), Asociación de Variables (Registros) y Arreglos. Cadenas de Caracteres como Arreglos. Estructura de Control Predeterminado de Repeticiones (FOR).</p>

<b>13</b>	Lenguajes de Programación. Paradigmas. Ambientes de Desarrollo. Compilación e Interpretación. Depuradores. Enlace de Módulos Objeto.	Búsqueda y Substitución de Patrones en Cadenas de Caracteres. Arreglos como Parámetros. Carga de Arreglos desde Archivos de Texto. Búsqueda de Elementos en Arreglos. Almacenamiento de Arreglos en Archivos de Texto.
<b>14</b>	Activación de Subprogramas y Administración de la Memoria por parte del Compilador. Almacenamiento de Arreglos y Procesamiento de Archivos.	Ordenamiento de Arreglos. Búsqueda Óptima de Elementos en Arreglos Ordenados. Operaciones Binarias con Arreglos Ordenados: Unión, Intersección y Diferencia. Entrega del Segundo Trabajo Práctico..
<b>15</b>	TERCER PARCIAL: Evaluación Integradora sobre Software de Sistema, Lenguajes de Programación y Conceptos de Pascal Directivas de Compilación y Puesta a Punto de Programas. Programación Concurrente.	SEGUNDO PARCIAL: Evaluación Integradora de la parte Práctica.
<b>16</b>	Devolución Tercer Parcial y Discusión general sobre problemas conceptuales detectados.	Devolución del Segundo Parcial y Resolución de la evaluación en forma conjunta en el pizarrón.

### *Acerca de la Metodología*

Los conceptos se introducen mediante organizadores previos que sirven de anclaje entre los conocimientos previos y los nuevos a ser incorporados [7], con posterior presentación teórica de conceptos fundamentales y resolución metódica de problemas tipo y ensayos sobre objetivos y luego, para el trabajo, de los estudiantes, el método utilizado para la enseñanza es por descubrimiento, es decir, los problemas se plantean de manera que el alumno descubra cuál o cuáles son las herramientas más apropiadas para resolverlos.

Para cada subunidad temática del programa se plantea una guía de ejercicios de contrastación e interpretación de resultados, que generen dudas y motiven la profundización del conocimiento a través de la bibliografía. Los alumnos pueden desarrollarlos en papel o mediante la computadora, en forma independiente de los trabajos prácticos.

Los trabajos prácticos están diseñados para que puedan ser resueltos por grupos de tres alumnos con el objetivo de crear las condiciones necesarias para que el alumno obtenga la formación deseada que es el trabajo cooperativo donde se comparten decisiones y responsabilidades y se resuelven los problemas de manera racional [1].

### *Ejemplo de Segundo Trabajo Práctico (Integrador)*

Desarrollo de un programa en lenguaje Pascal que permita determinar los valores  $x_1, x_2, \dots, x_n$  (por el método de eliminación gaussiana) que satisfacen el siguiente sistema de  $n$  ecuaciones lineales:

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n = b_1$$

$$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n = b_2$$



...

$$a_{n1} x_1 + a_{n2} x_2 + \dots + a_{nn} x_n = b_n$$

La ecuación puede escribirse en forma vectorial como  $A x = b$  donde:

- A es una matriz de  $n \times n$ , no singular, (el determinante de A es distinto de cero) cuyos elementos son los coeficientes de las incógnitas, y
- x y b son vectores, cuyos elementos son las incógnitas y los términos independientes, respectivamente.

Se verifica que para el desarrollo de la solución los alumnos la descompongan y definan subprogramas cuanto menos para:

- Lectura y escritura de arreglos
- Buscar el pivote
- Intercambiar filas
- Reducir a la forma triangular superior
- Sustituir hacia atrás

### *Acerca de las Evaluaciones*

“Los resultados de la enseñanza constituyen el criterio de valor de los posibles métodos en función de las diferentes condiciones” [2].

La evaluación es una función didáctica que se estructura con las funciones de enseñanza y de aprendizaje a la manera de un mecanismo interno de control.

Durante el cuatrimestre, a los efectos de contemplar diferentes momentos del aprendizaje de los alumnos, la evaluación tiene las siguientes funciones: diagnóstica, formativa y sumativa.

La cátedra prepara guías correspondientes a las diferentes unidades temáticas, en las cuales se abordan en forma concisa y conceptual los temas teóricos y prácticos que se tratan en las clases. Cada una de estas guías contiene dos grupos de ejercicios, unos referidos a la parte teórica y otros referidos a la parte práctica.

En la parte práctica, los docentes, al finalizar cada clase, indican a los alumnos cuáles son los ejercicios que deben plantear para la próxima clase en forma individual. Estos ejercicios tienen por objeto que los alumnos adquieran las habilidades básicas para poder aplicar cada herramienta en forma adecuada. En la clase siguiente, y en función de las dificultades que los alumnos hayan tenido para resolver los ejercicios, se analizan en forma conjunta en el pizarrón. Durante el cuatrimestre, los alumnos deben elaborar dos trabajos prácticos, uno que se debe entregar a mediados del cuatrimestre y otro que se debe entregar al finalizar el mismo. Estos trabajos prácticos se realizan en forma paralela a los ejercicios de las guías y consisten en actividades generativas. Una actividad generativa busca la capacidad de aplicar principios y conceptos centrales a la resolución de problemas concretos. El primer trabajo práctico está diseñado para que, dado un problema, los alumnos apliquen el conocimientos adquirido en el área de algoritmia y programación hasta estructuras de control inclusive; en tanto que el segundo trabajo práctico, es integrador y los alumnos deben aplicar todos los conocimientos adquiridos acerca de algoritmia y programación para resolver un problema, Los alumnos pueden controlar sus avances en el Trabajo Práctico en las fechas de consulta. La calificación del trabajo práctico es conceptual y se evalúan las siguientes categorías:

- Funcionamiento, es decir, que a través del programa desarrollado se llegue al objetivo deseado,
- Método, es decir, que se utilice el método de refinamientos sucesivos y que se haga uso de la documentación interna del programa.
- Presentación, es decir, que se entregue el trabajo práctico en forma completa.

La resolución de los ejercicios de las guías y de los trabajos prácticos en forma controlada, por parte de los alumnos permite a los docentes hacer un seguimiento del aprendizaje de éstos.

En la parte práctica, el primer parcial, tiene por objeto evaluar el aprendizaje de los alumnos hasta Estructuras de Control inclusive. Dado el caso en que el alumno no lo apruebe, podrá recuperarlo con la aprobación del segundo parcial pues éste integra los temas del primer parcial.

Luego del primer parcial se trata la subunidad temática “Subprogramas”. Los subprogramas son una herramienta importante para el desarrollo de algoritmos y programas. “La programación modular, que consiste en dividir el programa en módulos, es uno de los métodos de diseño más flexibles y potentes para mejorar la productividad de un programa” [4]. A los efectos de evaluar en forma individual el grado de comprensión por parte de los alumnos de esta unidad temática, se realiza una Evaluación Diagnóstica sobre este tema. La calificación de esta evaluación es conceptual.

El segundo parcial tiene una finalidad sumativa, se realiza al finalizar el cuatrimestre y permite detectar si los alumnos han adquirido la capacidad para diseñar y programar la resolución de un problema de tipo técnico – científico para que luego pueda ser ejecutado por una computadora.

Tanto en los dos parciales como en la Evaluación Diagnóstica, se evalúa la resolución de algoritmos y su representación en Lenguaje Pascal.

Los aspectos que se evalúan son:

- Optimalidad del algoritmo (el algoritmo que desarrolla el alumno debe ser general y no ambiguo, las herramientas se deben utilizar en forma adecuada y en el caso de modularizar el problema, la información debe transferirse a través de parámetros)
- Utilización de la metodología de refinamientos sucesivos y de la documentación interna
- Funcionamiento del programa para el enunciado presentado (aquí se tiene en cuenta el análisis significativo de la información )

En los parciales y en la evaluación diagnóstica, se plantea a los alumnos un problema de complejidad semejante a los desarrollados en el curso, donde deben aplicar conceptos y principios fundamentales dentro de una situación propuesta representativa de los contenidos generales del área a evaluar. El problema se debe resolver en clase, en dos horas, en forma individual y en papel. Es a libro cerrado, y los docentes escriben en el pizarrón la sintaxis de las estructuras en Lenguaje Pascal que fueron enseñadas.

La corrección se realiza en forma pormenorizada [6].

*Esquema de Categorías para la Corrección de los Parciales y de la Evaluación Diagnóstica:*

Sobre una nota máxima de 10 puntos, la categorización para cada objetivo a evaluar es:

Funcionamiento del programa	Optimalidad del algoritmo	Utilización de la metodología de refinamientos sucesivos y de la documentación interna
-----------------------------	---------------------------	--

3.33	3.33	3.33
------	------	------

El día en que se entregan las evaluaciones los docentes resuelven el problema dado en el pizarrón en forma conjunta con los alumnos, con el objeto de hacer notar los errores observados y de reforzar aquellos temas que denoten una mala interpretación.

En la parte teórica, luego de presentados los conceptos fundamentales de cada unidad temática, los alumnos deben plantear en clase y con ayuda del docente, los problemas teóricos referidos a la unidad temática. De acuerdo a la percepción que tenga el docente acerca del grado de asimilación de los temas, éstos se vuelven a tratar.

Si bien la teoría tiene un eje que es la algoritmia, hay unidades temáticas bien diferenciadas, cuyos conceptos se pueden evaluar en forma independiente. Por esta razón se toman tres parciales. El contenido de cada evaluación está formado por preguntas y ejercicios, cuya respuesta y resolución denoten el manejo de los conceptos y la aplicación de los conocimientos.

La corrección de los parciales teóricos es pormenorizada y depende de los temas evaluados [6].

#### *Ejemplo de conformación de la calificación de las Evaluaciones teóricas*

En el caso en que sean preguntas teóricas, se fija un porcentaje a cada pregunta

Este porcentaje tiene en cuenta:

- Análisis significativo de la información: 20%
- la asimilación y relación de conceptos: 80%

En el caso en que se deba desarrollar un ejercicio, se ponderan los siguientes puntos:

- Análisis significativo de la información: 10%
- Metodología utilizada para la resolución: 80%
- Arribo al resultado: 10%

#### *Conformación de las Calificaciones de la parte teórica y de la materia*

La calificación total sumativa de cada estudiante correspondiente a la parte teórica se obtiene hallando el promedio de las calificaciones obtenidas en los parciales teóricos.

La calificación total sumativa de cada estudiante correspondiente a la materia se obtiene hallando el promedio de las calificaciones finales obtenidas en la parte práctica y en la parte teórica.

#### *Régimen de Aprobación*

Aquellos alumnos que aprobaron la parte teórica, la parte práctica y los Trabajos Prácticos, pueden promocionar la materia, de acuerdo con el régimen promocional de cursado y aprobación de la misma. Aquellos alumnos que no aprobaron el Segundo Parcial de la parte práctica, tienen una oportunidad para recuperarlo en fecha de Recuperación de la Evaluación Parcial. Aquellos alumnos que aprobaron la parte práctica y los Trabajos Prácticos y desaprobaron todos o alguno de los parciales teóricos, deberán rendir una Evaluación Integradora Final en la cual se realizará una evaluación teórico-conceptual de toda la asignatura o solamente de los módulos desaprobados según el caso. El alumno dispone de tres oportunidades durante un año y medio para aprobar la materia.

## Conclusión y Trabajos Futuros

Los autores de este trabajo consideran que las innovaciones introducidas en la enseñanza de la materia han contribuido a mejorar la calidad de la enseñanza, pero detectan que aun subsisten falencias en el aprendizaje de los alumnos. Se percibe un bajo porcentaje de alumnos que realizan consultas en las clases prácticas y teóricas, como también, dificultades que presentan los alumnos en el proceso de comprensión, construcción y de apropiación de ideas, que se visualiza en la falta de optimalidad de las soluciones desarrolladas para resolver los problemas que se plantean en las evaluaciones.

Los pasos a seguir tenderán a sustituir la exposición teórica de las subunidades temáticas por un proceso didáctico puramente por descubrimiento de manera a seguir produciendo innovaciones a los efectos de alcanzar los resultados que pretenden conseguirse, fundamentalmente, el logro de la mayor interactividad con el alumno y la mejora y mantenimiento de su motivación.

## Referencias

- [1] “Diseño del Curriculum de una Asignatura de Carreras de Informática”. Servetto, A. V Congreso Internacional de Ingeniería Informática. Departamento de Computación. Facultad de Ingeniería. Universidad de Buenos Aires. 133 - 140. 1999.
- [2] “La psicología del aprendizaje y los modelos de diseño de enseñanza: la teoría de la elaboración”. Traducción de Juan José Aparicio-Frutos del libro “La psicología en la formación inicial del profesorado”. Boletín del I.C.E. de la Universidad Autónoma de Madrid. 1994.
- [3] “Enseñanza, Curriculum y Profesorado”. Contreras Domingo, J. AKAL Universitaria, España. Cap. 1. 1990.
- [4] “Programación en Turbo Pascal” Versiones 5.5, 6.0 y 7.0. Joyanes Aguilar, L. McGraw-Hill. 1993.
- [5] “Introducción a las Ciencias de la Computación”. Brookshear, G. Addison-Wesley Iberoamericana. 1995.
- [6] “Diseño y desarrollo curricular”. Zabalza, Miguel A., Narcea. 1991.
- [7] “Teoría del aprendizaje significativo”. Palomino N., W. <http://www.didacticahistoria.com/psic/psic02.htm>. 2001.