

# Estrategia Didáctica y Recursos Tecnológicos para la Enseñanza de los Sistemas de Numeración

Marcia Mac Gaul, Marcela F. López, Eduardo F. Fernández

Institución: Universidad Nacional de Salta. Facultad de Ciencias Exactas, Salta, Argentina

[mmacgaul@cidia.unsa.edu.ar](mailto:mmacgaul@cidia.unsa.edu.ar), [marcelaflopez@gmail.com](mailto:marcelaflopez@gmail.com), [effe@cidia.unsa.edu.ar](mailto:effe@cidia.unsa.edu.ar)

## Resumen

En el año 2010 se implementa un nuevo Plan de Estudio de la Licenciatura en Análisis de Sistemas. Este plan se formula atendiendo a las recomendaciones establecidas por RedUNCI y los estándares formulados por el Ministerio de Educación, en el marco del proceso de acreditación que sigue la CONEAU. En este escenario, las nuevas asignaturas básicas tienen por objetivo profundizar los conceptos algorítmicos de la Programación Estructurada, aplicándolo transversalmente en todos los contenidos de la asignatura. Se presenta entonces, el análisis diagnóstico en el que se indaga la relevancia del problema de enseñar contenidos clásicos de ciencias de la Computación, desde un enfoque de resolución de problemas computacionales a través del diseño algorítmico. Se describe la aplicación *Diagramar*, utilizada para la construcción, depuración y documentación de algoritmos a partir de componentes. Se ilustra la metodología utilizada en las cátedras aplicándola a un caso práctico en el contexto de Sistemas de Numeración. El enfoque algorítmico transversalmente asociado a éste y a otros contenidos de la asignatura procura potenciar las competencias de abstracción en diversas situaciones problemáticas propias de la Computación y la reutilización y adecuación de componentes para un diseño algorítmico de calidad en un contexto de aprendizaje mediado por TICs.

*Palabras claves:* Sistemas de Numeración – Metodología de enseñanza – Algoritmos – Enfoque modular – Recursos tecnológicos.

## Abstract

In 2010 implemented a new Plan of Study Bachelors in Systems Analysis. This plan is formulated on the recommendations established by RedUNCI and standards formulated by the Ministry of Education, as part of the accreditation process that follows the CONEAU. In this setting, the new basic subjects are

aimed at deepening the algorithmic concepts of Structured Programming, applying across all content of the course. Then presents the diagnostic analysis which investigates the relevance of the problem to teach classical content of Computer Science, from a computational problem solving by algorithmic design. Is described implementation *Diagramar*, used for building, debugging and documentation of algorithms from components. It illustrates the methodology used in the chairs by applying it to a case study in the context of numbering systems. The cross-linked algorithmic approach to this and to other contents of the subject seeking to enhance the skills of abstraction in various problem situations own Computer and reuse and adaptation of components to algorithmic design quality in the context of ICT mediated learning.

*Keywords:* Numbering Systems - Teaching Methodology - Algorithms - Modular Approach - Technology Resources.

## 1. Introducción

En el año 2010 se implementa un nuevo Plan de Estudio de la Licenciatura en Análisis de Sistemas (LAS), en reemplazo de su antecesor del año 1997. La reformulación de la carrera se basa en las recomendaciones establecidas por la RedUNCI y los estándares formulados por el Ministerio de Educación, en el marco del proceso de acreditación que sigue la CONEAU. Como resultado de este proceso, la carrera se acredita por Resolución 696/11 de la CONEAU.

En LAS 2010, las dos asignaturas cuatrimestrales de primer año, correspondientes al área de Algoritmos y Lenguajes, son Elementos de Programación y su correlativa Programación, las que sustituyen, respectivamente, a Elementos de Computación e Introducción a la Programación de LAS 1997. Siguiendo las recomendaciones y estándares mencionados, las nuevas asignaturas básicas tienen por objetivo profundizar los conceptos algorítmicos de la Programación Estructurada. La estrategia didáctica

para tal fin se asienta sobre la resolución de problemas computacionales a través del diseño algorítmico.

La Res. Ministerial 786/09, por la que el Ministro de Educación aprueba los contenidos curriculares básicos, la carga horaria mínima, los criterios de intensidad de la formación práctica y los estándares para la acreditación de LAS y otras carreras informáticas, señala en el punto II.5: *en el plan de estudios los contenidos deben integrarse horizontal y verticalmente. Asimismo deben existir mecanismos para la integración de docentes en experiencias educativas comunes.* Por lo tanto, este trabajo reúne las experiencias docentes de las dos asignaturas iniciales, dos de cuyos autores son responsables de sendas cátedras, quienes además dirigen Proyectos de Investigación y Desarrollo<sup>22</sup> acreditados por el Consejo de Investigaciones de la misma Universidad, e integraron un Proyecto de Investigación Científica y Tecnológica Orientado (PICTO)<sup>23</sup> acreditado en la Agencia Nacional de Promoción Científica, Tecnológica y de Innovación (ANPCyT). Desde estos espacios de investigación se trabaja con un equipo integrado por una pedagoga y docentes-investigadores que se desempeñan en las dos asignaturas informáticas del primer año universitario.

Se sostiene que la docencia universitaria y la investigación son actividades complementarias. Los docentes-investigadores, en su rol de profesores aplican estrategias didácticas elaboradas a partir de los resultados de la investigación y en ésta, usan el aula como un laboratorio a partir del cual se registran datos, se efectúan conjeturas, se contrastan resultados y se alcanzan algunas conclusiones que se trasladan a la práctica docente, enriqueciéndola.

La transferencia del grupo de I+D+i se concreta, principalmente, en el desarrollo de una aplicación que interpreta algoritmos, permitiendo su ejecución automática con fines de prueba y documentación. Este software educativo denominado *Diagramar*, se utiliza en el marco de una estrategia didáctica, en la que los contenidos – incluso los correspondientes a las matemáticas aplicadas a la Computación – se abordan desde un enfoque algorítmico.

La investigación indaga los procesos cognitivos de los estudiantes iniciales de Programación. Se reconocen las dificultades que presentan para el estudio de los Sistemas de Numeración. En particular, los obstáculos de origen epistemológico que se presentan cuando el conocimiento del Sistema Decimal actúa como obstáculo para el conocimiento de otros sistemas de

numeración: Binario, Octal y Hexadecimal. Se advierte la necesidad de abstracción que requiere mantener estable, tanto la cantidad como la representación hablada y escrita, de números expresados en distintos sistemas numéricos. Las dificultades de representación, entre otros, provienen de verdaderos conflictos con los saberes anteriores. Dice Brousseau [1]:

*“...describir el conocimiento, comprender su uso; explicar qué ventajas procura en relación a los usos anteriores, con qué prácticas sociales se relaciona, con qué técnicas y, si es posible, con qué condiciones matemáticas; señalar esas concepciones en relación a otras posibles, especialmente las que le sucedieron, para comprender las limitaciones, las dificultades y finalmente las causas del fracaso de esta concepción, pero al mismo tiempo las razones de un equilibrio que parece haber durado un tiempo lo suficientemente largo; buscar posibles resurgimientos, retornos imprevistos, si no bajo la forma inicial, al menos bajo formas similares, y las razones de ello.”*

Al respecto del equilibrio mantenido a lo largo de un tiempo, debe considerarse que en el Sistema Decimal los números tienen un nombre que los identifica. Por ejemplo el *trece*, que en notación matemática es  $(13)_{10}$  o simplemente 13 como es usual. En otro sistema de numeración, tal como el octal, la magnitud 13 se representa como  $(15)_8$ . La denominación apropiada para este número octal es *uno-cinco*, explicitando los caracteres constitutivos del número y no *quince* como sugeriría erróneamente su lectura en sistema decimal.

Este trabajo presenta las posturas metodológicas que se sostienen para la enseñanza de los Sistemas de Numeración, desde las cuales, mediante la aplicación de una estrategia basada en el desarrollo algorítmico de problemas vinculados al contenido, con apoyo de recursos tecnológicos, entre los que se encuentra *Diagramar*, se propone profundizar la abstracción de los contenidos respecto a la temática desarrollada.

## 2. Análisis diagnóstico

La realidad de los ingresos masivos está afectada de una serie de problemas que impactan directamente en la calidad educativa. Entendido como un fenómeno multidimensional, parte de ellos corresponden a razones exógenas a la Universidad. Sin embargo, otros son propios y merecen especial atención, por lo que se enuncian a continuación como una aproximación diagnóstica que sustenta las acciones desarrolladas.

### 2.1. El alumno inicial

En el trienio 2011-2013, las carreras informáticas, LAS y TUP (Tecnatura Universitaria en Programación),

<sup>22</sup> Proyectos CIUNSa. N° 1638 (2007-2009) “Una estrategia metodológica: el uso de las NTICs en el ingreso masivo universitario” y CIUNSa. N° 1865/3 (2010-2013) “Entornos Virtuales para la Articulación entre el Nivel Medio y carreras universitarias de Ciencias Exactas”.

<sup>23</sup> PICTO N° 36724 (2008-2010).

registran una pre-inscripción cuya media es de 458 estudiantes. Estos alumnos cursan la asignatura Elementos de Programación. Al finalizar el cuatrimestre, los que registran su inscripción definitiva, son, en valores medios, 212 ingresantes y 59 recursantes. La diferencia se explica desde un abandono medio del 40%, que se corresponde con un valor histórico superior al 30% observado en los últimos diez años y coincidente con el mismo fenómeno en carreras similares de otras universidades nacionales.

El alumno que concurre es, en su gran mayoría, un egresado reciente del nivel anterior. No siempre dispone de procesos cognitivos que le permita un grado de abstracción adecuado para permanecer en el primer año. Tampoco cuenta con hábitos ni estrategias de estudio que le permita consolidar su inclusión en el sistema superior.

## 2.2. Los objetivos y los contenidos

El principal objetivo de Elementos de Programación es que el alumno resuelva problemas computacionales aplicando algoritmos fundamentales. Los contenidos se distinguen en tres grandes ejes:

- Conceptos iniciales de la Programación, con énfasis en el diseño de algoritmos.
- Elementos de computación básicos, que se asientan sobre formalizaciones de la Matemática Aplicada: Sistemas de Numeración y Álgebra de Boole.
- Contenidos complementarios e introductorios orientados a la alfabetización informática.

Respecto a los conceptos básicos de la Computación que se apoyan sobre formalizaciones matemáticas, interesa explicar las razones por las que estos contenidos, específicamente los Sistemas de Numeración, se prevén en la asignatura del área de Algoritmos y Lenguajes y no en las del área de Ciencias Básicas, entre las cuales está Matemática para Informática, asignatura paralela a Elementos de Programación. De menor a mayor importancia, estas razones son:

1. El porcentual de carga horaria que cada área tiene asignada en el Plan de Estudio de LAS. El mayor porcentaje corresponde a Ingeniería de Software, Bases de datos y Sistemas de Información (30,58%), seguida por las áreas de Algoritmos y Lenguajes y de Ciencias Básicas, cada una con un 17,36%. De contemplarse el contenido en cuestión dentro de las Ciencias Básicas, que integra los contenidos matemáticos clásicos para la formación de un Programador y Analista, se produciría una redefinición de porcentuales, en la que Algoritmos

y Lenguajes tendría menor incidencia que Ciencias Básicas.

2. Su estrecha vinculación con asignaturas posteriores del área de Arquitectura, Sistemas Operativos y Redes.
3. La orientación algorítmica que se le puede asignar a la práctica de Sistemas de Numeración.

## 2.3. La metodología y los recursos tecnológicos

El ingreso se atiende con un plantel docente integrado por profesores y auxiliares. Los primeros a cargo del dictado de contenidos en su mayoría teóricos, que se imparten como clases magistrales a un numeroso grupo de alumnos, durante las instancias presenciales. No obstante, el uso de recursos tecnológicos imprime una dinámica más participativa de los estudiantes frente a la ejecución de algoritmos clásicos cuya teoría sustentan. Las clases prácticas se desarrollan en diversas comisiones de cursado paralelo, cada una de ellas a cargo de un Jefe de Práctica y Auxiliares Docentes Alumnos. Parte de estas prácticas corresponden a actividades de programación en laboratorios de computadoras.

Desde los proyectos de investigación se orienta hacia la construcción de un espacio en el que las competencias tecnológicas de los jóvenes se apliquen al ámbito educativo; en contraposición a otras estrategias que las ignoran, propiciando exclusivamente la cultura del papel. Desde el año 2008, las cátedras cuentan con sus respectivos entornos virtuales para la enseñanza y el aprendizaje (EVEA), montados sobre la plataforma MOODLE, en el servidor e-cidia, perteneciente al Centro de Investigación y Desarrollo en Informática Aplicada de la UNSa. El diseño de estos EVEA se sustenta a partir del debate y la selección de nuevas estrategias didácticas acordadas por el cuerpo docente. Luego de seis años de experiencia, se puede afirmar que los docentes fueron tomando, paulatinamente, nuevos roles de tutoría más propios de los escenarios educativos actuales [2,3,4,5,6,7]. La modalidad *extended-learning* adoptada prevé la asistencia de los estudiantes a instancias presenciales y la participación en actividades montadas sobre la plataforma, dirigidos principalmente al seguimiento de los aprendizajes en un marco de construcción grupal y colaborativo.

La aplicación *Diagramar* [8], utilizada desde el año 2009, es un intérprete gráfico que permite crear, editar y ejecutar un Diagrama N-S<sup>24</sup> y visualizar la prueba de escritorio que se obtiene como resultado de ejecutar el diagrama. *Diagramar* dinamiza el trabajo colaborativo en el EVEA, ya que permite socializar la solución de

<sup>24</sup> Nassi-Schneiderman

los algoritmos presentados en la materia, a partir de los archivos gráfico y ejecutable que la herramienta genera y que se debaten en el entorno virtual.

A modo de ejemplo, se listan algunos problemas presentados para su solución algorítmica, en el contexto de Sistemas de Numeración:

*Dado un número entero, indicar la mínima base de cada sistema de numeración al que puede pertenecer.*

*Dados dos números naturales  $P$  y  $Q$  pertenecientes al sistema de numeración de base  $\beta < 10$ , donde  $P \geq Q$ , realizar la resta  $P - Q$  por el método del complemento.*

*Convertir números enteros de cualquier base a la base 10 por el Método de Suma Ponderada.*

*Convertir números enteros de la base 10 a cualquier base por el Método de la División Reiterada.*

*Convertir un número entero  $N$ , aplicando el Método Directo. Los datos de entrada son  $N$ , la base del sistema de partida,  $BP$  y la base del sistema de llegada,  $BL$ . Estos dos últimos números son naturales mayores que 1 y menores que 10 y  $BP < BL$ . Importante: el algoritmo no debe convertir  $N$  a sistema decimal y de allí al sistema de base  $BL$ . Debe reproducir el procedimiento del Método Directo.*

## 2.4. La relevancia del problema

El Plan de Estudio de LAS vigente desde el año 2010, posee como principal diferencia con su antecesor (1997), que la asignatura inicial de Computación, modifica el orden de los contenidos impartidos según los ejes mencionados anteriormente. El contenido central es la resolución de problemas computacionales a través del diseño algorítmico. En el plan anterior, este contenido se desarrollaba en la segunda mitad del cuatrimestre, luego de presentar, entre otros, Sistemas de Numeración y Álgebra de Boole. La principal fortaleza de este diseño curricular, era la de trabajar con menor cantidad de alumnos un contenido históricamente reconocido como complejo. Esto era así debido a que un primer parcial eliminatorio, que evaluaba Sistemas de Numeración y Álgebra de Boole, reducía notablemente el número de estudiantes habilitados para continuar el cursado. Sin embargo, la principal debilidad radicaba en el escaso tiempo disponible para que un estudiante inicial construyera estructuras lógicas de razonamiento, adecuadas para

resolver exitosamente una variedad de problemas computacionales clásicos, tales como ordenamiento, búsqueda, inserción, eliminación y otros que son fundamentales para introducirse en la Programación. Esta debilidad se acentuaba al momento de aplicar estos algoritmos fundamentales a la resolución de problemas más complejos, cuyas narrativas están en un contexto de realidad y por lo tanto, se necesita la adaptación de estos componentes básicos y su combinación con otros elementos lógicos de la diagramación.

Por otro lado, se pudo constatar a partir de distintos dispositivos de evaluación, que los contenidos asimilados por los alumnos respecto al tema Sistemas de Numeración, consistía en la mecanización de métodos carentes de profundización conceptual de la temática.

Para revertir esta situación en la nueva asignatura, los contenidos Resolución de Problemas Computacionales y Algoritmos fundamentales, se ubican en las cuatro primeras unidades del programa, mientras que Sistemas de Numeración y Álgebra de Boole se postergan para los últimos momentos del cursado. Este nuevo cronograma permite asignar al tema Algoritmos, cuatro meses de dedicación, propiciando así un abordaje más gradual de contenidos y técnicas. Para atender la dificultad de trabajar este tema con un alumnado masivo e inicial, se habilitan espacios de apoyo tutorial desde el entorno virtual, promoviendo el uso de la aplicación *Diagramar*, para concretar la reutilización de los algoritmos clásicos en la solución de problemas asociados con los temas subsiguientes de Sistemas de Numeración y Álgebra de Boole. De esta forma, la cátedra adopta un enfoque algorítmico en el abordaje de todos los contenidos de la materia, orientado a un proceso de aprendizaje gradual y contextualizado a problemas propios de la disciplina. Para la especificación algorítmica se utilizan principalmente los diagramas N-S; los cuales permiten realizar una representación gráfica del diseño de programas bajo el paradigma estructurado (también se especifican algoritmos usando pseudocódigo, pero mantener un vocabulario controlado y una adecuada indentación para las estructuras de control resulta algo complejo para el estudiante inicial). La diagramación está basada en el diseño *top-down* que propone como estrategia para la resolución de un problema, dividirlo en subproblemas cada vez más pequeños y simples hasta obtener instrucciones elementales destinadas a la construcción de un programa. A esta metodología de diseño, se suma un importante concepto asociado a la reutilización de código. Su abordaje permite no solo trabajar sobre aspectos técnicos fundamentales para la construcción de software de calidad, sino que propicia un espacio para la reafirmación de conocimientos a través de la reutilización de las propias producciones de los estudiantes y, fundamentalmente, de algoritmos elementales que dimos en llamar *componentes*, los

cuales se encuentran definidos y documentados en forma previa e integran una galería de la aplicación *Diagramar*.

Surge de la investigación, que tradicionalmente, el diseño de los algoritmos se realiza en el aula presencial con papel y lápiz, herramientas que no permiten trabajar en profundidad las diferentes actividades involucradas, tales como la prueba y la documentación. A esto se suma la dificultad manifestada por los estudiantes para realizar satisfactoriamente las pruebas de sus algoritmos, lo que genera un alto grado de dependencia con el docente, cuyo rol se desvirtúa, constituyéndose en sólo un *probador de código*. Por otra parte, los docentes advierten que los estudiantes no comprenden cómo los diagramas que diseñan pueden transformarse en los programas reales que se ejecutan en una computadora. En general, no pueden captar la dinámica de funcionamiento de aquello que escriben en el papel, percibiendo al diagrama planteado como una descripción estática y no como un proceso dinámico en el que “suceden cosas” durante la ejecución de las instrucciones allí planteadas. En este aspecto, el docente que sigue prácticas tradicionales, realiza grandes esfuerzos para mostrar el comportamiento dinámico de los algoritmos, utilizando la pizarra, sin obtener mayores éxitos.

A medida que el uso de la tecnología se fue incorporando como un recurso que colabora a mejorar la calidad de la propuesta educativa, los estudiantes manifiestan la necesidad de contar con un software que permita no sólo el diseño de algoritmos, sino también su ejecución, depuración y documentación y, sobre todo, les otorgue más autonomía en su aprendizaje. A partir de estos requerimientos, el análisis en profundidad de los procesos cognitivos de los estudiantes al resolver problemas computacionales y los resultados de la encuesta aplicada sobre atributos de calidad, se modela *Diagramar*, adecuado para el aprendizaje de conceptos y técnicas asociados al diseño de algoritmos que considera las necesidades manifestadas por los alumnos-usuarios.

El docente-investigador usa este software con el objetivo de indagar más profundamente el proceso de aprendizaje de sus alumnos. La aplicación permite pausar la ejecución y el profesor aplica este recurso cuando identifica momentos de ruptura entre lo que el estudiante cree que el algoritmo realiza y lo que la aplicación procesa.

## 2.5. Breve descripción de la aplicación

*Diagramar* posee las siguientes herramientas (ver Figura 2).

- Prueba de Escritorio, con la que se ejecutan los Casos de Prueba,

- Casos de Prueba que ofrece una serie de recomendaciones para ejecutar la prueba, procurando ejercitar diversos caminos de ejecución y
- Documentación, que permite al usuario editar las anotaciones marginales de su trabajo de diagramación generando un documento que se vincula al gráfico del diagrama desarrollado.

Estas funcionalidades se incorporaron al software para dar respuesta a un requerimiento común a todos los estudiantes entrevistados; en sus propias palabras, “*cuando veo un problema parecido bien resuelto, lo uso para diagramar la solución del nuevo problema*”. Además, permite que el alumno se habitúe a la consulta y producción de documentación asociada al desarrollo de programas.

Desde el punto de vista de la aplicación, el usuario ve la construcción de un nuevo algoritmo como una tarea muy sencilla ya que se realiza a través de la selección del elemento a insertar y de la zona del diagrama en la que debe ser ubicado. Cada objeto insertado admite la configuración de sus propiedades a través del panel ubicado en la zona derecha de la pantalla. Los conceptos de modularidad y reutilización de código se trabajan a través de una galería de componentes estándar para ser utilizados por el usuario. Los componentes se dividen conceptualmente en grupos según el tipo de variables que utilizan: simples, indizadas unidimensionales o vectores y bidimensionales o matrices.

El aspecto central de la aplicación reside en la posibilidad que brinda para la depuración del algoritmo, efectuando la ejecución con diferentes velocidades de inspección de las instrucciones, una ejecución paso a paso, la visualización de la prueba de escritorio, la documentación de los casos de prueba, la documentación de la justificación del diseño y el almacenamiento del algoritmo diseñado. También permite su exportación como archivo de imagen bajo diferentes formatos, que facilita la socialización de las producciones a través del EVEA de la cátedra. La experiencia indica que los debates relativos a temas clásicos de la algoritmia se enriquecen a partir de la posibilidad de compartir en el entorno virtual los diagramas desarrollados por estudiantes y docentes. La comunidad somete a consideración sus producciones y aporta comentarios respecto a las realizaciones de los otros, en una dinámica en la que los archivos ejecutables de la aplicación y los archivos gráficos de los diagramas, son recursos imprescindibles para el planteo de los debates.

### 3. Metodología para la creación de algoritmos computacionales

La metodología se aplica a partir de la conceptualización y definición de aquello que se entiende como problema computacional. Entendemos que, la resolución de problemas computacionales implica el descubrimiento de un algoritmo. Juan José Cueto [9] indica que “un algoritmo es un método para la solución de un problema” ... “por lo tanto debemos entender la importancia de estudiar el método de solución “algoritmo” para comprender cómo se está solucionando el problema.” En una primera aproximación al concepto de algoritmos se puede decir que un algoritmo es “un conjunto de instrucciones para ejecutar una tarea” (J. Glenn Brookshear). El mismo autor afirma que “una vez descubierto un algoritmo para efectuar una tarea, la realización de ésta ya no requiere entender los principios en que se basa dicho algoritmo, pues el proceso se reduce a seguir las instrucciones”. Entonces, la resolución de un problema implica el desarrollo de un esfuerzo intelectual que produce un algoritmo, el cual, al ser aplicado, genera un resultado, es decir, una salida en función de una determinada entrada. Sobre la base de este sustento teórico, la metodología que proponemos para resolver un problema y obtener así una especificación algorítmica probada, consta de las siguientes fases:

#### 3.1. FASE I: Comprender el problema

Enfatizamos entre los estudiantes la necesidad de trabajar a partir de una formulación del problema clara, concisa, precisa, que evite todo tipo de ambigüedades. Cuanto mejor esté formulado un problema mayor será la facilidad para encarar su solución. Se propone trabajar esta fase en tres subfases.

**Fase I.1) Reconocer términos:** es posible que en la formulación del problema existan palabras cuyo significado no sea conocido o interpretado en el contexto de la consigna. Este desconocimiento opera en los estudiantes como obstáculo para la cabal comprensión del problema y por tanto, lo desorienta hacia los posibles caminos de resolución. Es imprescindible despejar toda duda respecto de los términos desconocidos, ya sea a través de una investigación bibliográfica o recurriendo al redactor del problema para refinar los aspectos no interpretados. Se ilustra con un ejemplo de cada caso.

Problema A: “Dada una cierta cantidad de datos correspondientes a notas de estudiantes, se desea determinar el promedio y la moda de los mismos”. En este caso, habría que asegurarse de comprender el significado y proceso de cómputo del *promedio* y la *moda*. No tener certeza acerca de estos términos obliga a recoger información a través de alguna fuente, por ejemplo, un libro de estadísticas.

Problema B: “Dados los datos de facturación correspondientes a un año, en una cadena de heladerías, se desea determinar los montos mensuales y estacionales de facturación, distinguidos por cada local de la cadena”. La formulación del problema es clara para quien lo redactó, posiblemente el gerente de la cadena de heladerías. Sin embargo, el analista puede no descifrar la expresión *estacional*. Una posible interpretación es considerar las estaciones primavera, verano, otoño e invierno; pero también es posible pensar en otra separación lógica propia del rubro del negocio, digamos por ejemplo, diciembre-marzo, abril-agosto y septiembre-noviembre. Lo conveniente, en estos casos de incertidumbre, es consultar una fuente fidedigna directamente asociada con el destinatario de la solución computacional que se construya.

La metodología de la cátedra al respecto de la fase I.1 consiste en habilitar nuevas entradas al diccionario que provee MOODLE, para circunstancias como la que ilustra el problema A, en el que una mínima investigación bibliográfica resuelve la situación; o postear en el foro de Asistencia Temática lo que el docente, en calidad de “cliente” precisa respecto de los términos no reconocidos del problema B. Todo foro de asistencia académica está configurado “por grupos separados”, lo que permite a diferentes docentes adoptar libremente su criterio al momento de evacuar las dudas sobre los términos en discusión. Un profesor cuyo grupo está avanzado al momento de resolver problemas como los del nivel B, puede complejizar la situación problemática solicitando la entrada de datos de inicio y fin de cada período estacional.

**Fase I.2) Identificar Entradas, Salidas y Condiciones:** en esta fase, se trabaja sobre el reconocimiento de la E/S y las condiciones sobre esos datos como restricciones que sobre ellos operan. También es importante aclarar que la formulación del problema no siempre tiene definidos los identificadores (denominaciones elegidas para los datos) de la E/S. Por ello, una de las primeras tareas de esta fase, además de identificar la E/S, es asignar un identificador válido para cada dato. Si por ejemplo, se requiere de un dato edad, un posible identificador válido es justamente “edad”, “E”, o cualquier otra denominación que permita comprender la naturaleza del valor almacenado bajo dicho identificador. Desde el punto de vista pedagógico, se orienta hacia la elección criteriosa de identificadores y a una inmediata documentación de respaldo.

**Fase I.3) Diseñar casos de prueba:** un caso de prueba está constituido por una colección de datos de entrada y las condiciones o restricciones que sobre ellos operan, necesarios para obtener un resultado válido como salida. Una salida es correcta si el resultado es el esperado para los datos provistos como entrada. Esta es una importante fase dentro del análisis porque contribuye a que el alumno, mediante situaciones concretas de manipulación de datos, los opere

reflexivamente a través de un proceso que constituye la primera aproximación algorítmica de la solución.

Por ejemplo, retomando el Problema A relativo al cálculo del promedio y moda de un conjunto de notas de estudiantes, un caso de prueba sería:

Entrada: Se poseen 10 notas (condición para cada dato nota: número natural entre 1 y 10). Las notas son 5, 6, 2, 8, 3, 9, 6, 1, 10, 6.

Salida: Promedio = 5,6 ; Moda = 6

La función principal del docente durante esta fase es la orientación para que el estudiante reconozca otro u otros casos de prueba que ejerciten posibilidades diferentes. Por ejemplo:

Entrada: Se poseen 10 notas (condición para cada dato nota: número natural entre 1 y 10). Las notas son 5, 6, 5, 8, 3, 9, 6, 5, 10, 6.

Salida: Promedio = 6,3 ; Moda = 5 y Moda = 6 (existe más de una moda).

Al inicio de la asignatura el estudiante desarrolla un primer trabajo práctico orientado a la interpretación de consignas de diferente complejidad, pero sólo a los efectos de cumplir con la Fase I. Los casos de prueba se dejan en reserva para retomarlos luego de concretar las correspondientes especificaciones algorítmicas que se ejecutan sobre la aplicación *Diagramar*.

### 3.2. FASE II: Seleccionar componentes

Esta fase se corresponde con la de *concebir un plan*. Para el diseño de esta estrategia es necesario conocer todos los aspectos vinculados al problema, en especial, aquellos recursos que pueden combinarse y contribuir a la solución del mismo. Estos recursos son los que se denominaron *Componentes*. Poseer una galería de componentes facilita la selección y adaptación de los mismos con el fin de ensamblarlos coordinadamente para construir la solución. En este contexto metodológico, definimos **Componente** como un proceso elemental realizado por un autómata. Su característica principal es la de poseer una única funcionalidad, claramente definida.

### 3.3. FASE III: Diseñar el algoritmo

Los sistemas informáticos deben contar con la capacidad de cumplir tres tareas básicas: entrada (captación de datos), procesamiento y salida (transmisión de los resultados). Una primera idea sobre lo que es un algoritmo, es justamente ese conjunto de tareas ensambladas correctamente para que, a partir de una cierta entrada se pueda brindar la salida que resuelva el problema formulado.

### 3.4. FASE IV: Ejecutar pruebas de escritorio

Esta fase es crucial para lograr un desarrollo autónomo del estudiante que diseña sus primeros algoritmos. Por ello, se define en primera instancia cuáles son los objetivos de la prueba:

La prueba es un proceso de control con la intención de descubrir un error.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Cualquier producto de ingeniería (y de muchos otros campos) puede ser probado de una de dos formas: 1) conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa; 2) conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que “todas las piezas encajan”; o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Como es sabido, el primer enfoque se denomina *prueba de la caja negra* y el segundo *prueba de la caja blanca*. La orientación docente y las prestaciones de *Diagramar* están dirigidas a que la fase de prueba se desarrolle usando el enfoque de caja blanca.

## 4. Aplicación de la metodología a un caso práctico: la conversión de Sistemas de Numeración

En la unidad 2 del programa se presentan, entre otros, dos componentes que se aplican a la solución de diversos problemas algorítmicos que usan tipos de datos simples: *Separación\_Dígitos\_Entero* y *Composición\_Número\_Entero*. En la unidad 6 se presentan los sistemas de numeración binario, octal y hexadecimal y los métodos de conversión entre ellos. Para este trabajo interesa desarrollar sucintamente el Método de Suma Ponderada de potencias de la base y el Método de la División Reiterada.

**Conversión de sistemas de numeración:** desde que se inicia la unidad 6, el tema de operaciones en una misma base se presenta haciendo hincapié en los algoritmos asociados a dichos contenidos y se propone a los alumnos que luego de algunas prácticas de operaciones usando números concretos, formulen los correspondientes algoritmos que permitan sumar, obtener complementos y multiplicar en una base menor o igual a diez.

En lo que se refiere a conversión de sistemas también se mantiene y profundiza el mismo enfoque. Se presenta el tema estableciendo el concepto de base  $\beta$  y el de conjunto de caracteres,  $C$ , que contiene los dígitos que, individualmente o en combinaciones entre ellos, representan todo número del sistema. Se establece que la conversión se puede realizar de un sistema de cualquier base a otro de cualquier base, siendo la primera la base de partida y la segunda la base de llegada.

Se considera entonces a  $C$  como el conjunto de caracteres del sistema de numeración de partida, cuya base es  $\beta$  ( $\beta > 1$ ), y a  $\alpha_i \in C$  los elementos del conjunto  $C$  o dígitos de dicho sistema. Se establece además que  $C'$  es el conjunto de caracteres del sistema de numeración de llegada, cuya base es  $\beta'$  y que  $\alpha'_j$  los elementos del conjunto  $C'$ .

**Método de la Suma Ponderada:** se presenta el número a convertir como una suma ponderada de potencias de la base  $\beta$

$$N_{\beta} = (\alpha_n \alpha_{n-1} \dots \alpha_1 \alpha_0)_{\beta} = (\alpha_n \beta^n + \alpha_{n-1} \beta^{n-1} + \dots + \alpha_1 \beta^1 + \alpha_0 \beta^0)_{\beta}$$

Y se establece la equivalencia de cada dígito de base  $\beta$  en un número de la base  $\beta'$  denominado  $(\gamma_0)_{\beta'}$  por ser una combinación de dígitos y no necesariamente un dígito en el sentido atómico de pertenencia al conjunto  $C'$ .

$$(\alpha_0)_{\beta} \equiv (\gamma_0)_{\beta'} ; (\alpha_1)_{\beta} \equiv (\gamma_1)_{\beta'} ; \dots ; (\alpha_n)_{\beta} \equiv (\gamma_n)_{\beta'}$$

Como así también la equivalencia de la propia base del sistema de partida  $\beta$ , en el sistema de numeración de base  $\beta'$ . Teniendo en cuenta que  $\beta$  se expresa en base 10, resulta:

$$(\beta)_{10} \equiv (\beta^*)_{\beta'}$$

Entonces, se hace notar a los alumnos que el número convertido se obtiene a partir de una suma ponderada:

$$N_{\beta} = (\alpha_n \beta^n + \alpha_{n-1} \beta^{n-1} + \alpha_{n-2} \beta^{n-2} + \dots + \alpha_1 \beta^1 + \alpha_0 \beta^0)_{\beta} \equiv (\gamma_n (\beta^*)^n + \gamma_{n-1} (\beta^*)^{n-1} + \gamma_{n-2} (\beta^*)^{n-2} + \dots + \gamma_1 (\beta^*)^1 + \gamma_0 (\beta^*)^0)_{\beta'}$$

Este proceso se puede ver algorítmicamente estableciendo la conversión de cada uno de los dígitos, pudiendo contar para ello con una tabla de conversión y contando también con la conversión de la propia base, de esta forma, el proceso culmina con una sumatoria ponderada.

Los alumnos que ejercitan este proceso notan que, si bien el método es de aplicación general, resulta sumamente valioso para conversiones donde la base de llegada es la base 10, a fin de no tener que diagramar

procesos que realicen un álgebra desconocida por la computadora.

**Método de la División Reiterada:** la presentación de este método se basa en mostrar cómo en sistema decimal, la división entera de un número por 10 genera como resultado, en el cociente los dígitos más significativos y en el resto el dígito correspondiente a las unidades del número dado. Se justifica el método analizando cada resto de división donde el cociente es la base de llegada expresada en sistema de partida y comprobando que cada resto resulta menor que dicho cociente, por lo tanto, es el equivalente a un dígito en la base de llegada.

Nuevamente, se presenta a los alumnos este proceso en forma algorítmica y se discute respecto a la conveniencia de aplicar este método en los casos en que la aritmética sea la decimal, es decir, en los casos en que la base de partida sea  $\beta=10$ .

**Reutilización de Componentes en el caso práctico:** una vez que son presentados ambos métodos de conversión, se trabaja la conversión de sistemas desde el enfoque algorítmico guiado por la siguiente secuencia:

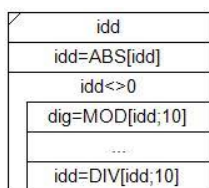
1. Identificación de los componentes asociados al proceso de conversión de números enteros: “*Separación\_Dígitos\_Entero*” y “*Composición\_Número\_Entero*”.
2. Adecuación de los mismos para su reutilización en el contexto de las situaciones problemáticas relacionadas con la conversión de sistemas.
3. Construcción de su propio componente de conversión de números enteros.
4. Validación del componente a través de la herramienta de prueba y depuración que ofrece *Diagramar*.
5. Socialización del componente entre pares y docentes a través del entorno virtual.
6. Reutilización de su componente en la transferencia propia de los problemas clásicos de la conversión de sistemas.

Se ilustra la metodología descrita con un par de figuras:

Para el paso 1, la Figura 1 muestra parcialmente el componente “*Separación\_Dígitos\_Entero*”.

Para los pasos 3 y 4, la Figura 2 muestra los componentes correspondientes al siguiente problema de conversión: “*elaborar un diagrama que permita convertir un número entero X expresado en sistema decimal, a su equivalente en base B (1 < B < 10) o viceversa, es decir, convertir a sistema decimal el número entero X*”<sub>B</sub>”.

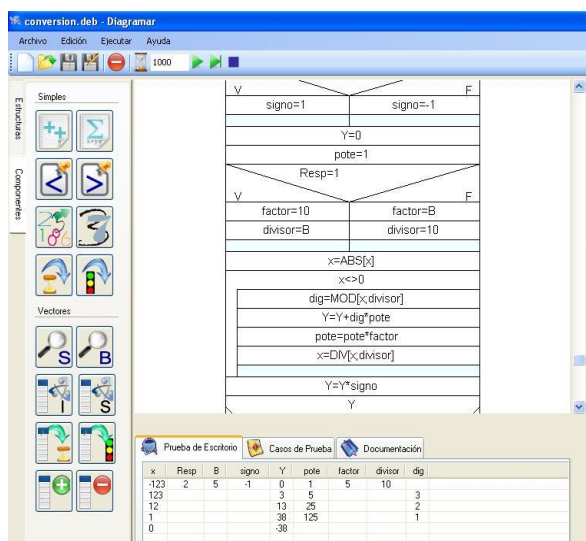




**Figura 1: Componente Separación\_Dígitos\_Entero**  
Donde:

**idd** es el identificador de la variable que contiene el número entero cuyos dígitos se desea separar.

**dig** es el identificador de la variable que contiene cada dígito del número entero **idd**, obtenido por aplicación de la función MOD.



**Figura 2: Reutilización de componentes**

## 5. Resultados

Se presenta un análisis de resultados, comparando la metodología actual, con la clásica aplicada en los planes anteriores a LAS 2010.

Los resultados corresponden a tres evaluaciones objetivas. En la primera (2007), se evalúan conversiones con aplicación de distintos métodos. En la segunda (2009), se propone un ejercicio que requiere conversiones y aplicación de cálculos sencillos. En la tercera (2012), aplicando la nueva estrategia que busca profundizar los conceptos algorítmicos de la Programación Estructurada, se evalúa la comprensión de los pasos involucrados en una conversión de sistemas. De esta forma, se intenta comparar si existe una mejora en las competencias que adquieren los alumnos en los primeros años de las carreras de computación.

Las consignas de las evaluaciones y sus alternativas de respuesta son las siguientes:

Evaluación año 2007

Consigna: Señalar la proposición correcta

Alternativas de respuesta:

- $1010_2 \equiv 3F2_{16} \equiv 1762_8$
- $1010_2 \equiv A_{16} \equiv 10_8$
- $1010_{10} \equiv 3F2_{16} \equiv 1762_8$
- Ninguna de las anteriores

Evaluación año 2009

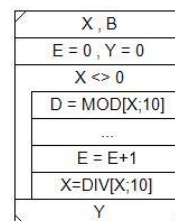
Consigna: Señalar el resultado de la siguiente operación:  $14_8 / 4_{16} + 200_5 - 401_5$

Alternativas de respuesta:

- $-48_{10}$
- $48_{10}$
- $-8_{10}$
- Ninguna de las anteriores

Evaluación año 2012

Consigna: El siguiente algoritmo describe el proceso de convertir un número natural  $X$  expresado en un sistema de numeración de base  $B$  menor que 10, a sistema decimal. Indicar la operación faltante en el rango del ciclo.



**Figura 3: Algoritmo aplicado en evaluación 2012**

Alternativas de respuesta:

- $Y = Y + D^E$
- $Y = Y + D * 10^E$
- $Y = Y + D * B^E$
- Ninguna de las anteriores

La tabla 1 muestra las cantidades de respuestas correctas e incorrectas observadas, en el conjunto de alumnos aprobados en las tres evaluaciones.

Plan LAS	Año	Bien	Mal	Total	% Bien
1997	2007	101	22	123	82%
1997	2009	138	25	163	85%
2010	2012	83	14	97	86%

**Tabla 1: Resultados estudiantes aprobados**

Los porcentajes no muestran una variación significativa en relación a las estrategias adoptadas.

La investigación se concentra en las posibles causas de error en la conversión de sistemas, entre los alumnos reprobados. Las siguientes figuras distinguen respuestas correctas de incorrectas de los ejercicios aplicados.

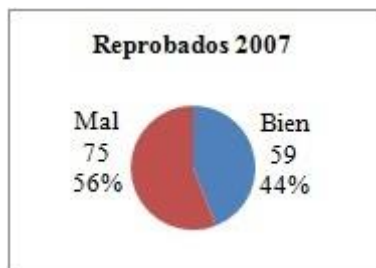


Figura 4: Resultados reprobados 2007



Figura 5: Resultados reprobados 2009



Figura 6: Resultados reprobados 2012

El ejercicio del año 2009 requiere la aplicación de métodos de conversión que pueden realizarse de forma mecánica, mientras que en el año 2007, además de las conversiones, se plantea la dificultad extra que supone mantener estables las representaciones escritas de números que parecen ser el mismo, pero que son de magnitud diferente, según sea el sistema numérico en el que están definidos. El 44% de estudiantes que responden bien, por debajo del 64% del año 2009, habla – al decir de Brousseau- de un posible obstáculo epistemológico, presente al momento de romper el equilibrio que el sujeto ha mantenido un tiempo suficientemente largo.

Profundizando la investigación, en el problema del año 2012, se propone aplicar las fases de la metodología descripta. Se espera que el alumno reconozca un Componente y su adaptación para construir la solución parcial del problema (Fase II), ensamble correctamente el paso correcto para brindar la salida que resuelva el

problema formulado (Fase III) y ejercite pruebas de caja blanca (Fase IV). El 51% de los alumnos que reprobaron el parcial realizaron mal este ejercicio, lo cual está por encima del 36% que se muestra en la Figura 5. Sin embargo, los 83 aprobados, más los 44 reprobados que lo resuelven correctamente, dan un total de 127 estudiantes, que es el 68% de los evaluados. Este porcentaje estaría señalando la conveniencia de aplicar una metodología basada en el enfoque algorítmico, para alcanzar mayores competencias de abstracción desde los cursos básicos de la carrera.

## Conclusiones y futuras líneas de acción

Como puede apreciarse, el enfoque algorítmico se encuentra transversalmente asociado a otros temas que aborda la asignatura. Ello en respuesta a un nuevo diseño curricular que sigue los lineamientos del proceso de acreditación, procurando potenciar las competencias de abstracción en diversas situaciones problemáticas propias de la Computación.

Los resultados alcanzados hasta hoy no permiten asegurar una correlación entre el enfoque algorítmico y el rendimiento académico de los estudiantes que abordan los Sistemas de Numeración. Sin embargo, esclarece la conjetura de la investigación, acerca del bajo nivel de conceptualización alcanzado al resolver sólo cálculos básicos de conversión, siguiendo la estrategia clásica de abordaje del tema, respecto de la estrategia actual, en la que deben manifestar el dominio de esas conceptualizaciones a través del diseño algorítmico.

La estrategia didáctica aplicada en los espacios presenciales y virtuales se traduce en evaluaciones que, paulatinamente, incorporan ejercicios y problemas en los que el estudiante debe comprometer un mayor esfuerzo de abstracción y comprensión algorítmica, por encima de las competencias básicas del cálculo mecánico.

Se promueve desde los cursos básicos, la reutilización y adecuación de componentes para un diseño algorítmico que gradualmente debe incorporar calidad en las producciones de los alumnos iniciales de Programación.

Este enfoque propicia el uso de las herramientas diseñadas a medida de nuestros estudiantes, tanto la aplicación *Diagramar* como la publicación de las producciones personales a través del entorno virtual de la cátedra. De esta forma, al finalizar el cursado en el que el contenido central es el diseño algorítmico, los estudiantes poseen, además de la galería de componentes suministrada por la aplicación, una enriquecida colección de algoritmos probados y documentados que se retoman en la materia correlativa

para su programación, usando un lenguaje de alto nivel.

Las asignaturas iniciales de Programación adoptan aulas extendidas, en las que se incorpora y se amplía el ámbito del aula mediante el uso de TICs. Considerando que las tecnologías móviles se insertan en la sociedad y particularmente en el colectivo de los jóvenes con gran aceptación, es que el grupo de investigación tiene proyectado desarrollar los software educativos *Diagramar 2.0* para dispositivos móviles y el módulo *Diagramar Colaborativo* para integrarse a la plataforma Moodle y que desde ésta sea posible interactuar con *Diagramar 2.0*.

En este sentido, siguiendo la experiencia enunciada por Susana Herrera en [10], el ecosistema de m-learning debe considerar ajustes a los contenidos y a los recursos, entre los que figura *Diagramar*, a fin de lograr su correcta adaptación y oportuna adopción en estos espacios extendidos de aprendizaje.

## Referencias

- [1] Brousseau G. "Obstacles épistémologiques, conflicts socio-cognitif set ingénierie didactique. En: Bodnarz N., Garnier C. (editores) (1989). Les obstacles épistémologiques et le conflit socio-cognitif. Construction des savoirs (obstacles et conflits)". Coloquio internacional CIRADE, Universidad de Québec, Montreal. 1989.
- [2] Cabero, J. La función tutorial en la teleformación, en Martínez, F. y Prendes, M.P. (coords): Nuevas Tecnologías y Educación, Madrid, Pearson Educación. (2004).
- [3] Duggleby, J. El Tutor Online. La enseñanza a través de Internet. Ediciones Deusto SA. (2001).
- Landeta Etxeberría, A. Capítulo 2: Modelos de Buenas Prácticas. Libro de Buenas Prácticas de e-learning. Artículo en línea <http://www.buenaspracticaselearning.com>. (2007)
- [4] González González, Carina. Tutorización, evaluación y aprendizaje colaborativo en el aula virtual: un enfoque práctico. Universidad de La Laguna. Artículo en línea <http://www.carlosruizbolivar.com/articulos/archivos/03> Tutorización y evaluación en elearning.pdf.
- [5] Fernández E., López, M., Mac Gaul, M. Reconstrucción del rol docente en los espacios virtuales. En actas de las Jornadas: Aula Virtual en la Universidad ¿Un espacio para todos? (2009).
- [6] López, M., Fernández, E., Mac Gaul, M. Metamorfosis de docente tradicional a docente tutor. En actas del congreso X Encuentro Internacional Virtual Educa Argentina (2009).
- [7] M. Mac Gaul, P. del Olmo, M. López, E. Fernández, M. Massé Palermo, C. Vargas, Repensando estrategias didácticas en nuevos escenarios educativos. En actas del Congreso Iberoamericano de Educación – METAS 2021 (2010).
- [8] M. Mac Gaul, P. Aballay, R. Zamora, M. Soria, Interpreter Chart Diagram N-S. En actas del XV Congreso Argentino de Ciencias de la Computación – CACIC (2009).
- [9] Juan José Cueto. Método para la solución de problemas utilizando POO. Ed. Universidad de San Martín de Porres, 2003.
- [10] Susana Herrera, Cecilia Sanz, Cristina Fennema MADE-mlearn: un marco para el análisis, diseño y evaluación de experiencias de m-learning". Revista TE&ET n°10. 2012.

### Dirección de Contacto del Autor/es:

Marcia Mac Gaul  
 Dirección: Av. Bolivia 5150  
 Localidad: Salta - Capital  
 País: Argentina  
 e-mail: mmacgaul@cidia.unsa.edu.ar

Marcela F. López  
 Dirección: Av. Bolivia 5150  
 Localidad: Salta - Capital  
 País: Argentina  
 e-mail: mfflopez@unsa.edu.ar

Eduardo F. Fernández  
 Dirección: Av. Bolivia 5150  
 Localidad: Salta - Capital  
 País: Argentina  
 e-mail: effer@cidia.unsa.edu.ar

---

Marcia Mac Gaul. Especialista en Didáctica. Profesor Asociado en Elementos de Programación. UNSa. Directora del Proyecto de Investigación N° 1865/3 de CIUNSA. Miembro titular del Instituto de Investigaciones en Educación a Distancia (IIEDi), UNSa.

---



---

Marcela F. López. Profesor Adjunto en Programación. UNSa. Co-Directora del Proyecto de Investigación N° 1865/3 de CIUNSA. Miembro adherente del Instituto de Investigaciones en Educación a Distancia (IIEDi), UNSa.

---



---

Eduardo F. Fernández. JTP en Elementos de Programación. UNSa. Miembro del Proyecto de Investigación N° 1865/3 de CIUNSA. Miembro adherente del Instituto de Investigaciones en Educación a Distancia (IIEDi), UNSa.

---