

Cuando la historia del ordenador condiciona el significado de vocablos de la vida cotidiana y de las Matemáticas

A. Taiana, C. Alarcón, G. Gagliano, R. Mainieri y M. A. Morelli

Resumen

Nos encontramos dedicadas a la tarea de construir una Didáctica Específica de la Algoritmia para la Programación, en pos de seleccionar trayectorias didácticas que mejoren el proceso instruccional dentro de esa área de la Informática. Presentamos los primeros resultados que nos han permitido descubrir las diferencias esenciales de significado que tienen, dentro del mundo de la programación, conceptos tan básicos como **algoritmo**, **dato** y **variable** y qué obstáculos encuentran los alumnos durante el proceso de aprendizaje.

Abstract

We are devoted to the task of building a specific Didactics of Algorithm for Programming with the objective of selecting more effective didactic paths to improve the teaching process within the area of data processing. We offer the first results obtained that have led us to discover essential conceptual differences within the scope of programming; for example, such basic concepts as **algorithm**, **data** and **variable**. More over, we describe some of the difficulties students face during the learning process.

Introducción

Constituimos un grupo de investigación dedicado a la tarea de mejorar los procesos de enseñanza/aprendizaje de la Algoritmia para la Programación y nuestro objetivo es encontrar los recursos necesarios para lograrlo.

Nuestros objetos de estudio son los algoritmos y queremos favorecer la construcción de los mismos por parte de los alumnos.

Se trata de algoritmos que, traducidos a un lenguaje de Programación, puedan ser luego ejecutados por un procesador. Nos referimos a estos algoritmos como objetos sistémicos pues están constituidos por objetos simples que a su vez nos dedicamos a analizar.

Nuestra idea de investigar los procesos instruccionales de la Algoritmia para la Programación nace como consecuencia de los errores que detectamos, en nuestros alumnos de primer año de carreras de ingeniería, en la Universidad Nacional de

Rosario, en Argentina, durante nuestras experiencias áulicas, en los procesos de enseñanza aprendizaje de esa disciplina.

Consideramos que nuestro aporte de investigación también es de utilidad para alumnos de los últimos años de la Escuela Media ya que nuestro trabajo atañe a las nociones básicas de la algoritmia y, por lo tanto es apto para estos niveles.

Como bien sabemos las Ciencias de la Computación derivan de las Matemáticas y fueron creadas por matemáticos; además, construir un algoritmo es determinar los pasos a seguir para resolver un problema.

Siendo la Resolución de Problemas uno de los ejes esenciales de la Didáctica de las Matemáticas, elegimos para nuestra investigación, un marco teórico dentro de la Didáctica de las Matemáticas. Tal marco teórico es el Enfoque Ontosemiótico (EOS) de Juan Díaz Godino de la Universidad de Granada, España. La decisión de esta elección tiene que ver con la riqueza, completitud y flexibilidad de esta teoría que brinda potentes herramientas de análisis de registros, tanto de textos, trayectorias didácticas, protocolos de alumnos y otros. Estas herramientas nos permitieron determinar y profundizar las coincidencias y divergencias del significado de los vocablos dato y variable en Programación parangonándolas con sus significados en las Matemáticas y en la vida cotidiana. Además detectamos algunos conflictos didácticos de los textos específicos de la disciplina.

Volviendo al concepto de algoritmo, ya en el siglo IV AC. Euclides, matemático persa, describió un mecanismo para calcular el máximo común divisor entre dos números naturales. En el transcurso de la Edad Media los árabes transmitieron, desde España, sus conocimientos matemáticos y de ellos proviene el término algoritmo, de la palabra *algorismus*, que es traducción al latín del apellido del matemático árabe Mohammed al-Khowarizmi, quien vivió en el siglo IX. Al-Khowarizmi alcanzó gran reputación por el enunciado de sus reglas paso a paso para sumar, restar, multiplicar y dividir números decimales.

Así es que, en su sentido más antiguo y original, el término algoritmo (en realidad algorismo) hace referencia al “proceso de hacer cálculos aritméticos usando números arábigos”. Paulatinamente el significado de la palabra fue modificándose y se generalizó su uso para referenciar a “una manera de hacer cálculo” (algoritmo infinitesimal) o a “un proceso para hallar un determinado resultado” (algoritmo algebraico).

Los especialistas en computación tomaron el término algoritmo concretando su significado a: “un método preciso para resolver automáticamente un problema”.

En la enseñanza de las Matemáticas se presentan algoritmos, ya tradicionales, para realizar determinadas acciones operativas, por ejemplo, la secuencia de operaciones que nos permiten calcular la raíz cuadrada de un número.

Consideramos de interés resaltar la diferencia que existe entre los algoritmos clásicos usados en las Matemáticas, más precisamente en la aritmética, y estos otros algoritmos, los que construimos para programar.

Según Fernández Bravo (2006), *“El hacer matemático no está en la aplicación del algoritmo sino en los mecanismos intelectuales que nos han permitido llegar a él”*.

Es en la Algoritmia para la Programación donde se deben poner en funcionamiento estos mecanismos intelectuales. El ordenador sólo ejecuta las instrucciones (órdenes) que aparecen en el algoritmo pero, el programador que diseñó el algoritmo, ha aunado todos sus recursos de creatividad y capacidad de abstracción para lograrlo.

En nuestra disciplina el ejecutor del algoritmo deja de ser el hombre, para pasar a serlo el ordenador. Debido a esto, con la aparición de los ordenadores a partir del siglo XX, el término algoritmo cobró una nueva dimensión.

Cuando hablamos de Informática y en particular de programar, que es la parte de la disciplina que estamos investigando, surgen dos elementos bien precisos a tener en cuenta:

- el algoritmo que nuestros alumnos deben aprender a diseñar y que resuelve el problema planteado, la situación-problema.
- el aporte y restricciones que genera la presencia del ordenador, al momento de la ejecución del mismo. Esto se ve reflejado en determinadas exigencias de sintaxis y, en algunos lenguajes de programación, en declaraciones específicas necesarias para la buena ejecución del programa. Estas especificidades deben necesariamente figurar previo al inicio del algoritmo y hacen a la buena ejecución del mismo.

Intentamos que nuestros alumnos logren diseñar con éxito la secuencia de pasos que constituye un algoritmo en Programación.

Esta tarea no es nada mecánica, en ella están fuertemente involucradas, como acabamos de mencionar, las exigencias de resolver un problema junto con las propias de hacerlo usando un ordenador como herramienta ejecutora.

Desde la Programación, desarrollar un algoritmo implica seleccionar la secuencia ordenada de pasos que llevan, si son correctos, una vez que el algoritmo ha sido ejecutado por el ordenador, a la determinación del resultado del problema.

Recordemos que el algoritmo, para ser ejecutado por el ordenador, debe haber sido codificado en lenguaje de programación (el programa).

La secuencia de pasos que constituye el algoritmo es creada por el alumno (incipiente programador o programador en formación) y la logrará a partir de:

- el conocimiento de los conceptos involucrados en el problema, al decir de Polya (1972) “*tenemos un plan cuando sabemos, al menos grosso a modo, qué cálculos, qué razonamientos o construcciones habremos de efectuar para determinar la incógnita*”,
- las exigencias propias de resolverlo usando un ordenador como herramienta ejecutora para la determinación de esa solución.

Siguiendo los lineamientos establecidos en el enfoque de investigación semiótico-antropológico, en la primera etapa de nuestro trabajo mostramos los resultados de un análisis semiótico aplicado a textos de la disciplina; a dos conceptos básicos: dato y variable.

Al igual que el vocablo algoritmo, las dos entidades fundamentales para la formalización de un algoritmo, dato y variable, ya conceptualizados por nosotros, Taiana & Otros (2005), son nociones primitivas y, muchas veces, de carácter intuitivo en su aplicación.

A continuación precisamos algunos detalles de los lineamientos del EOS que hemos utilizado para nuestro análisis.

Marco teórico de referencia

Asumimos una perspectiva antropológica, que interpreta a los objetos de la algoritmia como entidades compuestas (sistémicas) y relativas a un contexto particular, es decir a una institución y a un tiempo. La comprensión y el conocimiento no se conciben solamente en su dimensión mental, sino también, entre otras, en sus dimensiones personales e institucionales a las cuales nos referiremos, involucrando los sistemas de prácticas operativas y discursivas, es decir, lo que el sujeto hace y lo que el sujeto expresa, ante cierto tipo de tareas problemáticas. En su modelo teórico de la cognición matemática Díaz Godino establece seis dimensiones a tener en cuenta: el lenguaje, las definiciones, los argumentos, las proposiciones, las situaciones y los procedimientos.

Según Díaz Godino podemos considerar dos tipos de significados: los personales y los institucionales. Los significados personales incluyen el significado global, el declarado, el logrado, el final y el inicial, mientras que los significados institucionales están constituidos por los referenciales, los pretendidos, los implementados y los evaluados.

El análisis semiótico es entonces, la indagación sistemática de los significados puestos en juego en un proceso de enseñanza-aprendizaje (significados pretendidos-implementados), pudiendo coincidir o no con los significados personales del alumno. En la medida que se logre una adaptación entre ambos, habremos logrado lo que Godino define como Idoneidad Didáctica.

Asumiendo por Significados de referencia, dentro de los Significados Institucionales, a los libros de texto, nos preguntamos si los Significados Pretendidos, en pos de una Idoneidad Didáctica, son satisfechos por los Significados de Referencia.

Si bien el actual marco teórico de Díaz Godino permite, como mencionáramos anteriormente, analizar las configuraciones didácticas hasta en seis dimensiones, restringimos nuestro estudio del significado de los objetos o conceptos a un triplete epistémico constituido por *praxis*, *logo* y *lenguaje*. Estas tres componentes constitutivas de un objeto, a semejanza del triplete establecido por Vergnaud, son suficientes para el análisis, por las características de los objetos puestos en juego en este artículo, dado que no poseen una concepción excesivamente compleja.

- La praxis, incluye las situaciones-problema vinculadas al objeto y las acciones relacionadas.
- El logos está integrado por conceptos, proposiciones y argumentaciones.
- El lenguaje describe todas las posibles formas de representar al concepto, tales como expresiones, notaciones y representaciones.

Sin duda, el grado de completitud y complejidad de estas tres componentes dependen del momento, espacio y circunstancia en donde esté situado el proceso instruccional.

Hemos investigado estos dos objetos esenciales, dato y variable, recurriendo también a su epistemología e historia dentro de la Programación, desde sus primeros paradigmas hasta los actuales.

Hemos realizado análisis semiótico de textos de enseñanza de la Programación y de descripciones de experiencias áulicas específicas que nos permitieron corroborar la utilidad de nuestras investigaciones.

Como resultado del análisis, a continuación mencionamos algunas diferencias de los dos conceptos, desde la Programación y desde las Matemáticas, trabajamos sobre un ejemplo sencillo y presentamos en forma tabulada los resultados del análisis semiótico, describiendo los elementos constitutivos del triplete de cada uno de los dos objetos.

Significado de dato y de variable según los contextos

Los vocablos dato y variable son dos palabras de la vida real en general y en particular de las Matemáticas, de uso cotidiano.

Al preguntarles a alumnos novatos en Informática, qué significado tiene para ellos la palabra dato, algunos responden con concepciones surgidas del uso del vocablo en la vida cotidiana, (*“me dieron el dato de un buen restaurante”, “los datos del tiempo son...”*, *“un dato relevante de la encuesta es la gran cantidad de*

indecisos”....) pero, estas mismas preguntas realizadas a alumnos de niveles escolares superiores, producen respuestas muy variadas. Algunos no responden, otros dan respuestas más próximas a las “esperadas” por nosotros. Otros logran una mayor precisión en su expresión.

Frente a un problema matemático el alumno emplea esos vocablos en forma intuitiva.

A continuación presentamos un problema sencillo que analizaremos desde ambas disciplinas.

Veamos el siguiente problema: *Hallar el área de un círculo conociendo el radio del mismo.*

- Desde la Geometría identificamos el área del círculo con la fórmula:

$$\text{Área} = \pi \cdot \text{radio}^2$$

- Desde el Análisis Matemático la expresión:

$$\text{Área (radio)} = \pi \cdot \text{radio}^2, \text{ con radio} > 0$$

representa la ley de una función cuya expresión clásica es: $f(x) = \pi \cdot x^2$, su dominio son todos los números reales. En el contexto geométrico restringimos el dominio a los reales positivos. La función Área es una variable que llamamos *dependiente* de la variable *independiente* radio.

En Matemáticas no nos preocupa si el número con el que estamos operando es un número entero (real con parte decimal nula) o un número real con parte decimal no nula.

En Programación el tipo de dato con que trabajemos nos interesa pues la presencia del ordenador nos obliga en algunos lenguajes a declarar al comienzo, previo al inicio del algoritmo, el tipo de las distintas variables que usaremos durante la ejecución el programa.

Es así que trabajando con datos numéricos debemos diferenciar entre tipo de dato entero y tipo de dato real o lo que en Programación llamamos aritmética de punto fijo y aritmética de punto flotante. Esos valores quedan así almacenados en variables que se caracterizan por satisfacer las necesidades de capacidad de almacenamiento de estos números, según el tipo de dato que les corresponda almacenar a cada una de ellas.

Analicemos qué información nos brinda este sencillo enunciado de problema desde las Matemáticas.

- Datos intervinientes en el problema:

El único dato presente explícitamente en el problema es el radio.

Existe otro dato implícito en el enunciado del problema que proviene de la fórmula del área del círculo. Se trata del número π .

- Datos del problema vistos desde el algoritmo para la Programación que construiremos.

El radio es un dato cuyo valor será información de entrada y como π elegimos, por ejemplo, el valor aproximado constante 3.14.

Pero..., ¿significan algo más en nuestro algoritmo?

La respuesta es afirmativa, el dato radio deberá corresponderse con un lugar de memoria, donde será almacenado, que puede asumir distintos valores numéricos, (todos los posibles valores de los radios de los círculos a los que queramos calcularles el área), se trata, para nuestro algoritmo, de una variable.

El número π , en cambio, en el algoritmo representa una constante, más precisamente el valor aproximado y bien determinado que hayamos elegido para aproximarlos en la operación.

Si contextualizamos el concepto de Variable en Programación, nos preguntamos:

¿Se trata de nuestras conocidas variables usadas en las Matemáticas?

Se hace difícil, en Programación, distinguir entre los contenidos específicos: dato y variable pues a ambos, dentro del algoritmo, cuando se refieren a una misma medida del mundo físico se los nota con un mismo nombre que los identifica.

Sin embargo, el primero individualiza la representación de un elemento con que opera el procesador, mientras que el segundo identifica el espacio de memoria donde va a almacenarse el valor identificado con dicho nombre.

Las variables en Programación tienen una diferencia fundamental de significado con las variables de las Matemáticas. Ese significado es inherente a su rol en la propia Programación. El significado está vinculado con el soporte físico donde es almacenado su valor, identifica la dirección de la memoria donde el dato se almacena.

Si nos remontamos a la epistemología del término variable en Informática, éste tiene que ver, históricamente, con un cambio de paradigma en la Programación que, en pos de facilitar la construcción de algoritmos, buscando que su redacción estuviera más próxima al lenguaje humano, individualizó a las direcciones de los

lugares de memoria donde se almacena la información, con los nombres identificadores de los valores que dichos lugares almacenan.

Volviendo al problema presentado, si tenemos que calcular el área del círculo:

$$\text{Area} = \pi \cdot \text{radio}^2$$

Planteamos algunas preguntas:

- ¿Qué rol juega el radio en este problema matemático? El radio es un dato del problema, pero, ¿para qué se usan los datos de un problema en Matemáticas?, es la información necesaria para poder resolver el problema.
- ¿Qué es ese mismo radio en Programación? Es el dato de entrada necesario para el cálculo del área y esta última: Área, será la respuesta al problema.

Área es entonces el dato resultado de la operación.

- ¿Qué es el Área en este problema matemático? El Área es el resultado del problema.
- ¿Qué es esa misma Área en Programación? En Programación el Área también es el resultado buscado y por lo tanto es un dato de salida.

¡Radio y Área son datos en Programación!

Si pensamos nuevamente en el significado matemático de variable como:

$$\text{Área}(\text{radio}) = \pi \cdot \text{radio}^2, \quad \text{radio} > 0$$

El radio es la variable independiente y el Área la variable dependiente, para la función Área (radio).

¿Son Radio y Área variables en Programación?

Dado que ambos identifican dos direcciones de memoria que almacenan los respectivos valores podemos afirmar que, en Programación, se trata de dos variables.

En la Fig. 1 presentamos un trozo de algoritmo que realiza las acciones ejecutables necesarias para este sencillo cálculo:

Pi ← 3.14
Leer (radio)
Area ← Pi * radio * radio
Mostrar (' El área es: ', Area)

Figura 1: Cálculo del área a partir del radio

Dato y variable en Programación se representan con un mismo nombre, se opera con ellos de igual forma, pero uno está individualizando la notación de un elemento con que opera el procesador, mientras que el otro identifica la dirección de memoria donde va a almacenarse el valor representado por dicho nombre.

A continuación retomamos la idea del triplete de Godino-Batanero.

Elementos constitutivos de un concepto: logos, praxis y lenguaje

- *En el caso del dato:*

Logos: consideramos que un elemento constitutivo del logos del dato es su definición, elegimos una definición de uno de los textos analizados: “El dato es la expresión general que describe los objetos con los cuales opera un computador” Braunstein y Gioia (1995)

Praxis: nos referimos a cómo manipulamos con ellos.

- los leemos (datos de entrada): Leer (**radio**);
- los mostramos (datos de salida), Mostrar (**Area**);
- operamos con ellos: $Pi * \mathbf{radio}^{**2}$

Lenguaje: **radio**, **Area** son ejemplos de nombres o identificadores de cómo podemos representarlos.

- *En el caso de la variable:*

Logos: Al igual que para el dato el logos de la variable es su definición, mencionamos una: “La variable es una posición con nombre en memoria donde se almacena un valor de un cierto tipo de dato”, (Joyanes Aguilar & otros 2001)

Praxis:

- Le asignamos valores, por ejemplo: **Area** \leftarrow $Pi * \mathbf{radio}^{**2}$,
- Las usamos para construir expresiones.

Lenguaje: Al igual que en el caso del dato se lo identifica con un nombre. Ese nombre, además de estar representando un valor está determinando la dirección del lugar de memoria donde se va a almacenar el valor (la variable).

Las Fig.2 y Fig.3, muestran tablas, donde presentamos los elementos detectados como componentes de los respectivos tripletes, Taiana y cols (2005)



Fig. 2 Algunos elementos de las componentes del triplete del concepto Dato en Programación

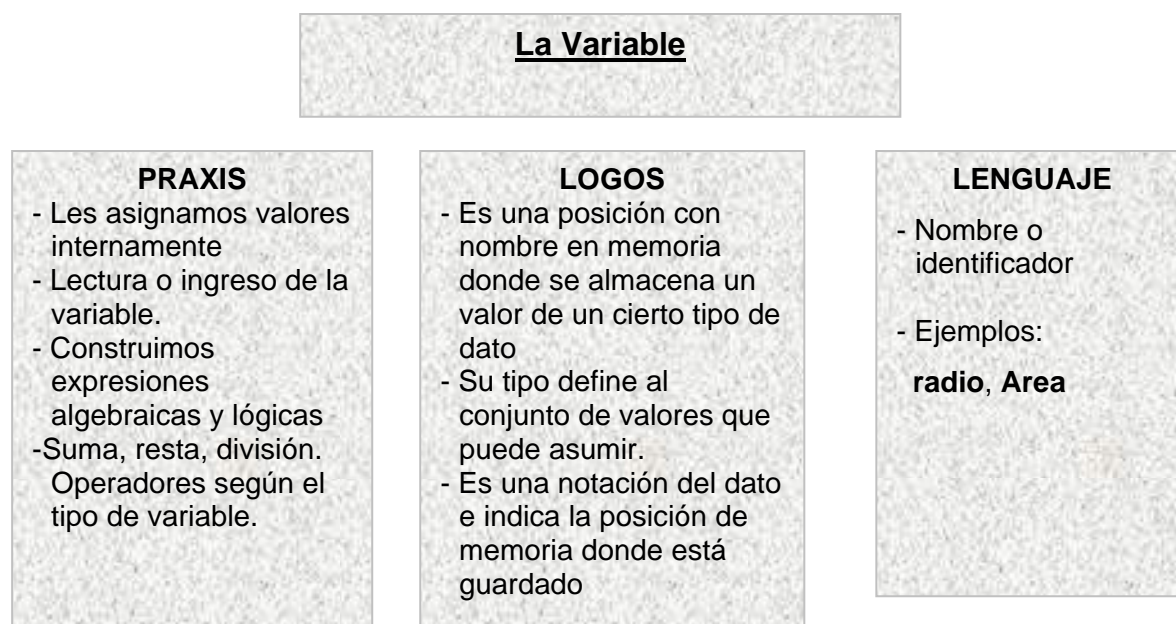


Fig. 3 Algunos elementos de las componentes del triplete del concepto Variable en Programación

Volviendo al tema de los errores, en este específico proceso instruccional, en el transcurrir de nuestros años de docencia, hemos detectado diversos tipos de errores.

Tipos de errores detectados en nuestros alumnos pero, ¡también en los textos!

En la mayoría de los casos, los errores detectados, más allá de la existencia de obstáculos ontogenéticos¹, se trata de obstáculos epistemológicos² y didácticos³.

Damos algunos ejemplos de errores de estos dos últimos tipos:

- ausencia de definición precisa de dato y de variable, que conlleva a que los alumnos permanezcan con la idea de que lo que identificamos como un dato o como una variable en Informática tiene el mismo significado que para la Matemática tradicional.
- confundir el concepto de constante numérica con el concepto de dato, por ejemplo:

En nuestro problema el valor elegido de

Los alumnos consideran que en Programación 3.14 o π es un dato, cuando en realidad se trata de una constante numérica dentro del algoritmo.

No hemos encontrado dos libros de programación que definan dato o variable en igual forma, es más, la mayoría de ellos no definen dato y pocos definen variable.

Conclusiones y reflexiones finales

En Programación dato y variable son dos vocablos que tienen características particulares que los identifican y esas particularidades están íntimamente involucradas, como ya lo expresáramos con la presencia del ordenador.

Los autores de libros de Programación no reparan y por lo tanto no precisan las diferencias de significado que hemos mencionado y que hacen que uno y otro concepto sean únicos. Esto produce errores de interpretación en nuestros alumnos, que podrían caracterizarse como obstáculos didácticos y epistemológicos, pues estas diferencias intrínsecas de los vocablos en uno u otro contexto, no están resueltas por los propios autores. A esto se suma la ausencia del proceso histórico epistemológico de los distintos paradigmas de la Algoritmia para la Programación.

Como mencionamos en la introducción del trabajo, el análisis presentado forma parte de una etapa preliminar en nuestra investigación, brindando elementos para el análisis epistémico a priori que nos proponemos desarrollar. Es evidente la necesidad de transitar diferentes focos de investigación. Si las cuestiones epistémicas resultan esenciales para caracterizar el objeto de enseñanza, los aspectos cognitivos y didácticos deberán ser abordados para trabajar el diseño de trayectorias didácticas y analizar en qué medida las mismas pueden favorecer el

¹ Los referidos a los procesos de maduración y de las estructuras de conocimiento que posee y pueda desarrollar el alumno

² Los inherentes al conocimiento propio en estudio y que se apropian al igual que los conceptos correctos

³ Los que introducen los maestros y que no derivan de propiedades del objeto de estudio

aprendizaje significativo y la negociación de significados, sin que esto vaya en desmedro de la veracidad de los mismos.

Cabe destacar, sin embargo, que no hemos presentado de manera extensiva la totalidad del análisis. No se han incluido, por razones de espacio, aspectos que son relevantes y fundamentales para completar la caracterización de los significados de los otros conceptos ya analizados, entre ellos, los elementos constitutivos de las estructuras de control de la Programación, Taiana y otras(2006).

Queremos resaltar la potencialidad del enfoque teórico-metodológico desarrollado por Godino-Batanero que nos permite reconocer y caracterizar los elementos de significado de los objetos de la Algoritmia para la Programación desde una perspectiva antropológica y semiótica. Dicho enfoque nos proporciona un lenguaje de referencia que posibilita la clarificación del campo de la Didáctica de la Algoritmia para la Programación orientándonos a futuras investigaciones y que favorece nuestro propio esclarecimiento del tema.

Continuamos nuestro trabajo con el análisis de cada uno de los contenidos esenciales de esta disciplina que convoca a todas las integrantes del grupo como docentes de la misma desde hace muchos años.

Bibliografía

- S. Braunstein, A. Gioia (1995): Introducción a la programación y a las estructuras de datos. Eudeba, Buenos Aires.
- G. Brousseau (1983): Les obstacles épistémologiques et les problèmes en mathématiques recherches en didactique des mathématiques Vol 7 Nro. 2, pp 33-115.
- J.A. Fernandez Bravo (2006): “Avatares y estereotipos sobre la enseñanza de los algoritmos en matemáticas” - Revista UNION nº 4).
- J. D. Godino, C. Batanero (1994): “Significado institucional y personal de los objetos matemáticos” Recherches en Didactique des Mathématiques, 14 (3): 325-355.
- J. D. Godino (2002): “Un enfoque ontológico y semiótico de la cognición matemática” Recherches en Didactique des Mathématiques, 22 (2/3): 237-284.
- J. D. Godino (2003): “Teoría de las Funciones Semióticas”. Trabajo de investigación presentado para optar a la Cátedra de Universidad de Didáctica de la Matemática de la Universidad de Granada. <http://www.ugr.es/local/jgodino>.
- J. D. Godino, D. Bencomo, V. Font, M.R. Wilhemi (2006): Análisis y Valoración de la Idoneidad Didáctica de Procesos de estudio de las Matemáticas. Paradigma XXVII (2) 221-252.
- L. Joyanes Aguilar, I. Zahonero Martinez (2001): Programacion en C, Metodologia, Algoritmos y estructuras de Datos McGraw-Hill Madrid.
- G. Polya (1972): Cómo plantear y resolver problemas Ed. Trillas. México.
- A. Taiana, A. Alarcón, G. Gagliano, R. Mainieri, M.A. Morelli (2005) GIDMyA “Conceptualización de elementos esenciales de la algoritmia en la programación” Memorias VCAREM Buenos Aires.

- G. Vergnaud (1990): La théorie des champs conceptuels. Recherches en Didactiques des Mathématiques. 10 (2-3).133-170.

- **Aída Taiana**, Licenciada en Matemática - Master en Didáctica de las Matemáticas, Directora GIDMyA (Grupo de Investigación en Didáctica de las Matemáticas y la Algoritmia)

Proyectos de investigación del grupo dependientes del Rectorado UNR

- 2005/2007 - Fundamentos de la Didáctica de la Matemática en el Proceso de Enseñanza / Aprendizaje de la Algoritmia
- 2007 continua - Hacia una Didáctica Especifica para la Algoritmia en Programación Integrantes GIDMyA:

- **Cristina Alarcón**, Ingeniera - Especialista en Docencia Universitaria

- **Gracia Gagliano**, Ingeniera - Especialista en Docencia Universitaria

- **Rosanna Mainieri**, Ingeniera - Especialista en Docencia Universitaria

- **María Alicia Morelli**, Ingeniera.

Todas las autoras son integrantes del GIDMyA y de los Proyectos antes mencionados y Profesoras de Informática I del Departamento de Matemática de la Escuela de Formación Básica de la FCEIA-UNR

A. Taiana, A. Alarcón, G. Gagliano, M.A. Morelli (2006): "Los Procesos Instruccionales en la Algoritmia para la Programación: Entidades simples y mixtas resultantes del Análisis Ontosemiótico-Epistémico de un ejemplo sencillo". CACIC2006 XII Congreso Argentino de Ciencias de la Computación San Luis-Argentina

A. Taiana, A. Alarcón, G. Gagliano, R. Mainieri, M.A. Morelli (2006): "Un Análisis Ontosemiótico de las Estructuras de Control de Repetición en la Modelización del Conocimiento de la Algoritmia", (8vo. SEM) Seminario de Educación Matemática CD: Registro ISBN-10: 987-20239-4-8; ISBN-13:978-987-20239-4-2.C. C. Borges. Buenos Aires- Argentina

A. Taiana, R. Katz, N. Sgreccia (2008): "Una secuencia didáctica para vincular la probabilidad experimental y la teórica" En: El aula de Matemática: conocimiento +entretenimiento Material didáctico para la escuela media 17-24

UNR Editora Colección académica ISBN 978-950-673-626-2

A. Taiana, R. Katz, N. Sgreccia (2008): "Presentación, análisis y fundamentación de una propuesta didáctica de matemática para distintos niveles de escolaridad"

En: El aula de Matemática: conocimiento +entretenimiento Material didáctico para la escuela media 81-90 UNR Editora Colec acad ISBN 978-950-673-626-2

A. Taiana, R. Katz, N. Sgreccia (2008): "Actividades que propician aprendizajes significativos desde la diversidad de registros".

En: El aula de Matemática: conocimiento +entretenimiento Material didáctico para la escuela media 91-116 UNR Editora Colec. Acad. ISBN 978-950-673-626-2