Comparative Analysis of Algebraic Generalization and Graph Paper Programming as Strategies that Favor the Development of Computational Thinking

Francisco Bavera[1,2] [0000-0001-7795-2139], Flavia Buffarini[1] [0000-0001-8340-0998], Fabiana Rosso[0009-0007-1363-786X], María Marta Novaira[1] [0009-0001-3657-3631]

[1] Universidad, Nacional de Río Cuarto, Argentina
[2] Universidad Nacional de Villa Mercedes, Argentina
pancho@dc.exa.unrc.edu.ar, fbuffarini@exa.unrc.edu.ar, fabianarosso67@gmail.com, mnovaira@dc.exa.unrc.edu.ar

Abstract. This paper examines the strategic convergence between mathematics education and computer science education, establishing computational thinking as an essential cognitive process in problem-solving. Two fundamental models are compared: Algebraic Generalization and Graph Paper Programming. The research analyzes how the transition from the concrete to the abstract, mediated by unplugged activities, constitutes the pedagogical core of both methodologies. It concludes that a profound synergy exists, where Algebraic Generalization contributes to the declarative conceptual design and the Graph Paper Programming the dynamic procedural implementation. This integration is vital for developing and/or strengthening higher-order skills (analyzing, creating, and evaluating) necessary for problem-solving and computational thinking.

Keywords: teaching methodologies, problem-solving, programming, thinking skills.

Análisis Comparativo de la Generalización Algebraica y la Programación en la Cuadrícula como Estrategias que Favorecen el Desarrollo del Pensamiento Computacional

Resumen. El presente trabajo examina la convergencia estratégica entre la didáctica de la matemática y la didáctica de las ciencias de la computación, fundamentando el pensamiento computacional como un proceso cognitivo esencial en la resolución de problemas. Se comparan dos modelos fundamentales: la Generalización Algebraica y la Máquina de Dibujar en la Cuadrícula. La investigación analiza cómo la transición de lo concreto a lo abstracto, mediada por actividades desenchufadas (unplugged), constituye el núcleo pedagógico de ambas metodologías. Se concluye que existe una sinergia profunda donde la Generalización Algebraica aporta el diseño conceptual declarativo y la Máquina de Dibujar en la Cuadrícula la implementación procedimental dinámica. Esta integración resulta vital para desarrollar y/o fortalecer habilidades de orden superior (analizar, crear y evaluar) necesarias en la resolución de problemas y el pensamiento computacional.

Palabras clave: metodologías didácticas, resolución de problemas, programación, habilidades del pensamiento.

1 Introducción

La didáctica de una determinada ciencia se preocupa por cómo se enseña y se aprende esta disciplina. Uno de los objetivos de la didáctica de las ciencias de la computación es desarrollar estrategias y enfoques efectivos para desarrollar el pensamiento computacional (PC) (Aho, 2012; Wing, 2014) de manera que sea accesible y significativo para los estudiantes. Desde la perspectiva de la didáctica de las ciencias de la computación, el pensamiento computacional se define como un proceso dialéctico de resolución de problemas que trasciende la interacción con dispositivos. Este enfoque implica la formulación de estrategias, la representación de datos mediante abstracciones y la automatización de soluciones a través de un pensamiento algorítmico crítico.

En el contexto educativo actual, la integración del pensamiento computacional en la educación primaria y secundaria se ha consolidado como una necesidad estratégica para preparar a los estudiantes para los desafíos del siglo XXI. Para lograrlo es fundamental contar con metodologías accesibles que no dependan de una infraestructura tecnológica costosa o compleja. En este marco, el enfoque de *actividades desenchufadas* (unplugged), propuesto a finales de los años 90 por Bell, Witten, y Fellows (1998), surge como una alternativa pedagógica relevante, ya que permite la enseñanza de conceptos de ciencias de la computación, la programación y la lógica algorítmica a través de actividades concretas, colaborativas y sin necesidad de utilizar computadoras. Al prescindir de infraestructura tecnológica, las actividades se centran en la construcción de bases conceptuales robustas sin limitaciones de disponibilidad de equipamiento.

En particular, dentro del conjunto de propuestas de programación desenchufada, la programación en papel cuadriculado (Code.org, 2014; Areces, 2018) se presenta como una metodología especialmente adecuada para estudiantes que se inician en la programación. Esta propuesta prioriza el trabajo con situaciones problemáticas significativas que promueven la explicitación de procedimientos, la construcción de algoritmos y el reconocimiento de regularidades. A través de actividades gráficas y simbólicas, la programación desenchufada favorece el desarrollo de habilidades propias del PC, tales como la secuenciación, la descomposición y la generalización, constituyéndose en una estrategia potente para el abordaje progresivo y reflexivo de la programación en contextos escolares.

Considerando que la didáctica de la Ciencias de la Computación es incipiente y se está desarrollando, se toman, para este estudio, aportes de la didáctica de la matemática para enmarcar, comprender y analizar la construcción de habilidades lógicas, algorítmicas y de resolución de problemas, como así también, sobre cómo abordar problemas mediante la identificación de patrones, la abstracción de soluciones y la formulación de estrategias formales.

El propósito de este trabajo es analizar y comparar dos metodologías didáctico-pedagógicas que favorecen el desarrollo de habilidades del PC: la *Generalización Algebraic*a (Sadovsky, 2004; Buffarini, 2005, 2018; Sessa, 2005) y la *Máquina de Dibujar en la Cuadrícula*, teniendo en cuenta que las capacidades/habilidades que se pretenden desarrollar son objeto de estudio en ambas metodologías.

La metodología de Generalización Algebraica utiliza problemas de conteo para guiar a los estudiantes desde estrategias de cálculo concretas hacia la derivación de fórmulas algebraicas generales. Por otro lado, la Máquina de Dibujar en la Cuadrícula es un enfoque que materializa los algoritmos en programas, introduciendo la necesidad de un lenguaje formal, preciso e inequívoco para la comunicación con una máquina.

A continuación se detalla cómo cada enfoque aborda conceptos clave del pensamiento computacional, desde la descomposición de problemas hasta la abstracción y la validación/depuración, ilustrando sus fortalezas individuales y su potencial sinergia en una progresión didáctico-pedagógica coherente.

2 Análisis de la Metodología de Generalización Algebraica

La metodología de Generalización Algebraica (GA) se presenta como una puerta de entrada estratégica al pensamiento abstracto y algorítmico. Su valor reside en conectar procedimientos concretos y tangibles, como el conteo manual, con la formalización simbólica del álgebra, sentando las bases para la construcción de soluciones algorítmicas con sentido. Este enfoque permite a los estudiantes experimentar el proceso de generalización de manera natural y significativa.

Esta metodología se desarrolla a partir de problemas *ejemplares*, como por ejemplo "*El Borde de un Cuadrado*". Los problemas de conteo, como el de determinar la cantidad de cuadraditos que hay en el borde de un cuadrado de mayor tamaño se proponen para favorecer el inicio de la generalización algebraica. Estas situaciones, que parten de escenarios donde pueden obtener la totalidad contando sobre el dibujo se va complejizando con el propósito de abandonar lo concreto y se avance hacia la abstracción mediada por el empleo de fórmulas algebraicas que permitan conocer la respuesta para cualquier situación de igual característica. Además, la fórmula tiene que tener implícita la manera de contar, lo que la vuelve segura y válida, pasando así a un lenguaje formal.

El proceso pedagógico se desarrolla a través de una secuencia de etapas diseñadas para guiar al estudiante desde lo particular a lo general:

- *Caso Específico*: Inicialmente, se pide a los estudiantes calcular la cantidad de cuadraditos en el borde de una figura concreta, como un cuadrado de 6x6 (ver Fig. 1). Esta tarea permite una aproximación directa, como el conteo uno a uno.
- *Necesidad de Generalización*: A continuación, se plantea el mismo problema para un cuadrado de mayor tamaño, como uno de 37x37. El tamaño elegido desalienta el conteo directo y obliga a los estudiantes a buscar simplificaciones, patrones y relaciones para encontrar una solución de manera más eficiente.
- *Explicitación de la Estrategia*: Se solicita a los estudiantes que expliquen por escrito el método o procedimiento de conteo que utilizaron. Este paso es crucial, ya que convierte una acción intuitiva en un algoritmo explícito.
- *Producción de la Fórmula*: Finalmente, se pide generalizar el método de conteo descrito en una fórmula que funcione para un cuadrado de cualquier

tamaño, representado por el lado n (cantidad de cuadraditos por lado). Este paso formaliza la estrategia en lenguaje matemático.

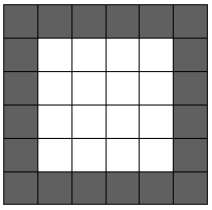

Fig. 1. Cuadrado de 6x6 para el caso específico del Borde de un Cuadrado.

La GA se sitúa como una puerta de entrada estratégica al pensamiento abstracto. Bajo el marco de la Teoría de las Situaciones Didácticas de Brousseau (1986, 1997), se plantea el problema de "El Borde del Cuadrado" como una Situación Fundamental que obliga a transitar desde la aritmética hacia el álgebra. El proceso inicia con una fase de Acción (conteo en un cuadrado de 6x6), pero evoluciona hacia una Situación Adidáctica al proponer un cuadrado de 37x37 (Buffarini, 2005). El tamaño de esta nueva figura actúa como una variable didáctica que desalienta el conteo uno a uno, forzando al estudiante a buscar regularidades. En la fase de Formulación, los estudiantes deben explicitar sus estrategias, lo que deriva en diversas expresiones algebraicas. Esta fase implica el desarrollo de procesos de validación de las estratégias y fórmulas planteadas.

Una de las mayores riquezas de este enfoque es la diversidad de estrategias de conteo que surgen de los estudiantes, como se puede observar en la Tabla 1, cada una de las cuales se traduce en una fórmula algebraica distinta pero equivalente. Esta equivalencia algebraica aporta riqueza pedagógica al modelo, donde se valida que múltiples algoritmos (fórmulas) pueden resolver el mismo problema.

El análisis de la equivalencia algebraica entre estas fórmulas es un pilar fundamental de la metodología. La discusión sobre por qué procedimientos tan diferentes como 4n–4 y 2n+2(n-2) producen siempre el mismo resultado permite a los estudiantes comprender un concepto profundo: puede haber múltiples algoritmos correctos para resolver un mismo problema. Esta comprensión se construye en tres niveles: la validación empírica mediante la evaluación en números particulares, la justificación contextual de que todos los métodos cuentan la misma cantidad, y la demostración formal apoyada en las propiedades de las operaciones.

Por otro lado, el análisis comparativo de las diferentes fórmulas permite reflexionar sobre cuál es la más conveniente y por qué en cada caso, así por ejemplo puede seleccionarse a 4n-4 como la más económica para calcular o se puede seleccionar porque en dicha fórmula se observa que el número total debe ser un múltiplo de 4. Las fórmulas muestran propiedades que los números ocultan, por ello algunas fórmulas son mejores que otras.

Tabla 1. Ejemplos de Estrategias de Conteo.

Estrategia de Conteo Descrita	Fórmula Algebraica	Justificación Lógica
"Contar 4 veces el lado del cuadrado de 37 y luego sacarle uno por cada esquina donde se superponen dos lados."	4n – 4	Evita la doble contabilidad de los vértices en el perímetro.
"Contar dos lados enteros de 37 y luego 35 por cada uno de los otros dos lados."	2n + 2(n-2)	Divide el borde en dos barras completas y dos segmentos internos.
"Contar un cuadradito menos por lado de 36 y luego multiplicar por 4 o sumar 4 veces lo mismo."	(n – 1) x 4	Visualiza el borde como 4 segmentos rotativos de igual longitud.
"Contar todos los cuadraditos del cuadrado de 37 de lado y restarle los del cuadrado de 35 de lado"	$n^2 – (n-2)^2$	Diferencia entre el cuadrado total y el espacio vacío central.

3 Análisis de la Metodología de Máquina de Dibujar en la Cuadrícula

La Máquina de Dibujar en la Cuadrícula (MDC) es un método que introduce a los estudiantes en la traducción de algoritmos a programas concretos y ejecutables. Su importancia radica en hacer explícita la necesidad de un lenguaje formal, preciso y sin la ambigüedad inherente al lenguaje natural, para comunicarse con un autómata. Este autómata puede ser un compañero ejecutando instrucciones en papel, un robot o una computadora, para reproducir un dibujo en la cuadrícula. En programación, la precisión es indispensable, ya que un autómata ejecuta instrucciones "*al pie de la letra*" (Martínez Lopez, 2017). Este enfoque materializa conceptos de programación que de otro modo podrían permanecer abstractos.

El proceso de construcción de un programa en la MDC sigue una progresión conceptual clara, que va desde lo simple a lo complejo y de lo concreto a lo abstracto:

- *Definición de Instrucciones Primitivas*: Se establece un conjunto de comandos limitado y sin ambigüedad, como ← (mover a la izquierda), → (mover a la derecha), ↑ (mover arriba), ↓ (mover abajo) y ⬛ (pintar el cuadrado). La máquina solo puede entender estas instrucciones.
- *Secuencia de Instrucciones Primitivas*: Se definen secuencia de instrucciones primitivas para obtener la solución al problema. Por ejemplo, dibujar una línea, donde una posible solución es ⬛ → ⬛ → ⬛ (ver Fig. 2).
- *Identificación de Patrones y Repetición*: Al escribir programas para tareas como dibujar un cuadrado, los estudiantes observan que ciertas secuencias de instrucciones se repiten. Esta observación conduce naturalmente a la

introducción de una notación formal para la repetición, como (⬛ →)3, que representa la ejecución de la secuencia ⬛ → tres veces.

- *Abstracción mediante Procedimientos*: Cuando los problemas se vuelven más complejos la división en subproblemas es una herramienta de diseño que permite gestionar la complejidad y fomentar la reutilización. Para esto se introduce el concepto de procedimientos. Estas construcciones son una herramienta fundamental para la abstracción y para expresar la estrategia, encapsulando secuencias de código bajo un nombre significativo como [DibujarUnCuadrado].
- *Generalización con Parámetros*: Para que los procedimientos sean generales, se introducen parámetros. En lugar de tener procedimientos específicos como [DibujarCuadrado3], se crea uno general como "[Dibujar cuadrado (nro_cuadraditos)]", que puede resolver toda una clase de problemas (dibujar un cuadrado de cualquier tamaño).

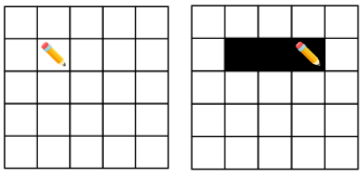

Fig. 2. Cuadrícula original y el resultado de ejecutar ⬛ → ⬛ → ⬛.

La descomposición en subproblemas y la legibilidad son dos principios fundamentales en esta metodología. Problemas complejos, como "*dibujar una casa*", se abordan de manera sistemática al descomponerlos en subproblemas más sencillos y manejables. El uso de procedimientos con nombres claros y descriptivos mejora drásticamente la legibilidad del programa. Esta diferencia se evidencia al comparar una larga e ininteligible cadena de símbolos con un programa estructurado que utiliza procedimientos claros como [Dibujar techo] y [Dibujar ventana], donde la legibilidad facilita la comprensión, depuración y modificación.

4 Habilidades del Pensamiento Computacional en GA y MDC

Habiendo analizado ambas metodologías por separado, el siguiente paso es compararlas directamente para evaluar cómo cada una desarrolla habilidades específicas del pensamiento computacional. En esta sección se realiza un análisis comparativo punto por punto para evaluar cómo las metodologías de Generalización Algebraica y Máquina de Dibujar en la Cuadrícula fomentan las habilidades clave del PC.

4.1 Descomposición del Problema y Formulación de Estrategias

- En GA: La descomposición es un proceso implícito. La búsqueda de diferentes maneras de contar los cuadraditos del borde (por ejemplo, "*contar dos lados enteros y dos lados menos dos*") representa una forma de

descomponer la tarea de conteo total en subcálculos más simples. La estrategia es el método de conteo en sí mismo, que luego se formaliza en una expresión algebraica.

- En MDC: La descomposición es un principio explícito y fundamental. Se instruye activamente a los estudiantes a dividir problemas complejos como "*dibujar una casa*" en subproblemas como "*dibujar pared*" y "*dibujar puerta*". La estrategia se articula a través de la secuencia de procedimientos que, en conjunto, resuelven el problema mayor.

4.2 Reconocimiento de Patrones y Repetición

- En GA: El reconocimiento de patrones es central para pasar de un caso particular ("*contar el borde de un cuadrado de lado 37*") a una regla general. El patrón es la regularidad numérica que subyace en la relación entre el lado del cuadrado y el número de cuadraditos del borde, dando origen a la fórmula.
- En MDC: El reconocimiento de patrones se manifiesta de forma visual y secuencial al observar que una misma secuencia de instrucciones primitivas se repite. Este reconocimiento conduce directamente a la formalización del concepto de repetición (ciclos), utilizando una notación específica, (instrucciones)numero, para hacer el código más conciso y claro.

4.3 Abstracción y Generalización

- En GA: La abstracción se logra al crear una fórmula (por ejemplo, (n-1)x4) que encapsula un procedimiento completo de conteo. La generalización se consigue mediante el uso de la variable n, que permite que la fórmula resuelva el problema para cualquier tamaño de cuadrado, representando una solución general a una clase de problemas.
- En MDC: La abstracción se consigue al definir un procedimiento (por ejemplo, [DibujarUnCuadrado]) que encapsula una secuencia de comandos bajo un único nombre. La generalización se alcanza a través del uso de parámetros (por ejemplo, nro_cuadraditos). Esta conexión es directa y análoga al rol de la variable n. Los parámetros permiten resolver problemas conocidos con datos desconocidos (valores concretos desconocidos). Por ejemplo, se debe dibujar un cuadrado pero no se conoce de qué tamaño entonces se generaliza la solución con el tamaño como parámetro.

4.4 Validación, Verificación y Depuración

- En GA: La validación de una fórmula se realiza mediante la evaluación en números particulares para constatar que produce el resultado esperado (similar a realizar prueba/testing sobre un programa). Cabe mencionar que la validación/justificación de que la fórmula es correcta también está dada porque se construye a partir de observar cómo se cuenta. En la Tabla 1 se puede ver claramente que la descripción de la estrategia de conteo y su justificación lógica son argumentos válidos que justifican absolutamente a la fórmula.

Además, se utiliza el razonamiento sobre las propiedades algebraicas para validar la viabilidad de un resultado; por ejemplo, al determinar que el resultado de 4n-4 siempre debe ser un múltiplo de 4, se puede descartar un conteo erróneo como 587. Este tipo de razonamiento promueve una comprensión más profunda de la estructura de la fórmula, alineándose con el objetivo de "*leer*" la información que una expresión algebraica proporciona. La verificación de las fórmulas se realiza por medio de la equivalencia algebraica entre ellas.

- En MDC: La prueba/testing (validación) y la depuración (debugging) son actividades explícitas y centrales. La prueba implica ejecutar un programa para corroborar que el resultado visual coincide con el objetivo. Si no lo hace, se inicia el proceso de depuración: encontrar y corregir los errores (bugs) en la secuencia de instrucciones hasta que el programa funcione correctamente. Si bien en las propuestas de MDC (Code.org, 2014; Areces, 2018) no se trabaja sobre la verificación de los programas podría introducirse en MDC, pero es un aspecto que necesita ser estudiado.

Las diferencias y similitudes observadas en este análisis señalan que estas metodologías se complementan y se enriquecen mutuamente.

5 Sinergia y Progresión Pedagógica

El análisis comparativo revela que, lejos de ser enfoques excluyentes, la Generalización Algebraica y la Máquina de Dibujar en la Cuadrícula forman una progresión pedagógica lógica y efectiva. La GA actúa como un "fundamento sólido y una introducción natural" a los conceptos que luego se materializan y expanden en la MDC y se aplican a introducir la programación.

La GA sienta las bases conceptuales para la MDC al establecer conexiones clave:

- El desarrollo de una estrategia de conteo en GA es conceptualmente análogo al diseño de un algoritmo en MDC. En ambos casos, se trata de definir una secuencia de pasos para resolver un problema.
- La generalización a una fórmula con la variable n en GA es el análogo conceptual directo de la generalización de un programa mediante parámetros en MDC. Ambas herramientas sirven para crear soluciones generales a partir de casos específicos.
- La discusión sobre fórmulas equivalentes en GA introduce la idea fundamental de que puede haber múltiples soluciones correctas para un mismo problema. Este concepto evoluciona en MDC hacia la evaluación de diferentes programas no solo por su corrección, sino también por criterios de calidad como la legibilidad, la eficiencia y la modularidad.

Se podría describir esta sinergia, pedagógicamente potente, de la siguiente manera: la GA se enfoca en la solución conceptual y universal (el "qué", un enfoque denotacional en la resolución de problemas), encapsulada en una fórmula que es

independiente de la ejecución. La MDC se enfoca en el proceso de implementación (el "cómo", un enfoque operacional en la resolución de problemas), utilizando herramientas (procedimientos) y especificaciones parametrizadas para construir una instancia concreta y ejecutable de la solución.

Esta progresión permite a los estudiantes construir una comprensión robusta del pensamiento algorítmico antes de enfrentarse a la sintaxis rígida de un lenguaje de programación.

6 Conclusión

Se han analizado dos metodologías didáctico-pedagógicas, la *Generalización Algebraica* y la *Máquina de Dibujar en la Cuadrícula*, concluyendo que ambas son enfoques robustos y eficaces para fomentar el desarrollo de habilidades del Pensamiento Computacional y enseñar los fundamentos de la programación a principiantes.

Cada metodología posee fortalezas distintivas que se complementan mutuamente. La Generalización Algebraica sobresale en el desarrollo del pensamiento abstracto, la habilidad de identificar patrones numéricos para formular reglas generales, la explicitación de estrategias de solución y la comprensión profunda del concepto de equivalencia entre diferentes soluciones algorítmicas. Por otro lado, la Máquina de Dibujar en la Cuadrícula es excepcional para enseñar conceptos de programación fundamentales de manera tangible, como la secuenciación de instrucciones, la descomposición de problemas, la abstracción a través de procedimientos y la importancia de un lenguaje formal. Además, ambas introducen de forma explícita las prácticas esenciales de prueba y depuración.

En consecuencia, la implementación secuencial de la Generalización Algebraica seguida de la Máquina de Dibujar en la Cuadrícula no solo enseña conceptos aislados, sino que modela el proceso auténtico del desarrollo de software (desde la concepción abstracta de una solución hasta su implementación concreta y verificable), ofreciendo así una secuencia de aprendizaje integral y de gran impacto.

La metodología y las actividades utilizadas en la generalización algebraica se pueden plantear como un fundamento sólido y una introducción natural para la programación con dibujos en la cuadrícula porque ambas requieren el desarrollo del pensamiento computacional, la abstracción de soluciones y la definición de estrategias formales. Se puede afirmar que el proceso de Generalización Algebraica introduce de manera temprana las habilidades clave del pensamiento computacional, como el reconocimiento de patrones, la división en subproblemas, la abstracción/generalización y la validación de soluciones, preparando conceptualmente al estudiante para la traducción de algoritmos a un lenguaje formal de programación mediante el uso de procedimientos y parámetros en una cuadrícula.

Durante la preparación de este trabajo, los autores utilizaron NotebookLM con el fin de mejorar el lenguaje y la legibilidad del manuscrito. Tras utilizar esta herramienta, los autores revisaron y editaron el contenido según fuera necesario y asumen plena responsabilidad por el contenido de la publicación.

Referencias

Aho, A. V. (2012). Computation and computational thinking. The Computer Journal, 55(7), 832-835.

Areces, C., et al. (2018) Secuencia Didáctica 2: Programas. Capítulo 2. Ciencias de la computación para el aula : 2do. ciclo de primaria : libro para docentes. 1ra edición. Fundación Sadosky.

Bell, T. C., Witten, I. H., et al. (1998). Computer Science Unplugged: Off-line activities and games for all ages. https://jmvidal.cse.sc.edu/library/bell98a.pdf

Brousseau G. (1986): Fundamentos y métodos de la Didáctica de la Matemática, Universidad Nacional de Córdoba, Facultad de Matemática Astronomía y Física, Serie B, Trabajos de Matemática, No. 19 (versión castellana 1993).

Brousseau, G. (1997). Theory of Didactical Situations in Mathematics. Kluwer Academic Publishers.

Code.org (2014) Lección 2: Programación con Papel Cuadriculado. Curso D. Code.org. https://studio.code.org/courses/coursed-2022/units/1/lessons/2

Buffarini, F. (2005): El álgebra como herramienta de modelización y validación. Las interacciones entre pares como medio de evolución. Tesis de Maestría. UNRC.

Buffarini, F., Bavera, et al. (2018) La construcción del pensamiento computacional: una propuesta desde la Didáctica de la Matemática. XLI Reunión de Educación Matemática. Unión Matemática Argentina. Universidad Nacional La Plata. La Plata, Argentina.

Martínez López, P. (2017): Sugerencias para el dictado del curso La programación y su didáctica. Método Program.AR Complemento al cuaderno Actividades para aprender a Program.AR de la Fundación Sadosky. Universidad Nacional de Quilmes. Versión del 3.

Sadovsky, P. (2005): Teoría de las Situaciones didácticas: un marco para pensar y actuar la enseñanza de la matemática. Reflexiones teóricas para la educación matemática, Libros del Zorzal.

Sadovsky, P. y Sessa, C. (2004): The adidactic interaction with the procedures of peers in the transition from arithmetic to algebra: a milieu for the emergence of new question. Revista Educational Studies in Mathematics, dedicado a la Escuela Francesa de Didáctica de la Matemática. Versión en castellano en http://www.fcen.uba.ar/carreras/cefiec/cefiec.htm

Sessa, C. (2005): Iniciación al estudio didáctico del algebra. Orígenes y Perspectivas. Libros El Zorzal.

Wing, J. (2014) Computational Thinking Benefits Society. Social Issues in Computing. Disponible en:http://socialissues.cs.toronto.edu/2014/01/computational-thinking/