

10048 SITUACIONES ESCOLARES DE JÓVENES QUE APRENDEN PROGRAMACIÓN. UNA POSICIÓN ACTIVA DEL SUJETO DEL APRENDIZAJE

María Emilia Echeveste

Facultad de Matemática, Astronomía,
Física y Computación- CONICET
meecheveste@gmail.com

Resumen: En este artículo se presentan dos situaciones de clases de dos escuelas técnicas con orientación en programación de la ciudad de Córdoba Capital, analizadas desde la relación de los estudiantes con sus docentes y con este conocimiento en particular. Se hará foco en algunas actividades que los jóvenes estudiantes realizan a la hora de acercarse a los conocimientos de computación y se tomarán principalmente aportes del pensamiento computacional de Brennan y Resnik (2012) y del construccionismo de Papert (1987). En el análisis de las situaciones presentadas se referenciarán acciones que realizan los estudiantes donde considero que se ponen en juego estrategias de aprendizaje y apropiación de los conocimientos. Se habla de conocimientos y no de saberes al realizarse desde un sentido operacional y relacional ligado a lo escolar.

Palabras clave: ESTUDIANTES SECUNDARIOS, APRENDIZAJE DE LA PROGRAMACIÓN, SUJETOS ACTIVOS, ESTRATEGIAS DE APRENDIZAJE.

Introducción

Las prácticas educativas adquieren en el aula con una materialidad y temporalidad específica. Aluden a procesos de transmisión y apropiación de contenidos curriculares seleccionados dentro de un universo amplio de conocimientos posibles de ser transmitidos, contextuales más amplias que marcan la tarea de enseñar y la tarea de aprender.

Particularmente nos interesa conocer cómo aprenden a programar los jóvenes en la escuela secundaria; cómo es esa relación que construyen los jóvenes con ese conocimiento. Un tipo de conocimiento específico, un conocimiento escolar, en el que no se puede dejar de lado ni sus inferencias institucionales ni del Sistema Educativo en general.

Para empezar a pensar en este tema recupero la noción de *educación en programación* que menciona Federico Aloí y su equipo de la Universidad de Quilmes (2016), donde consideran que *“La educación en programación requiere del dominio de diferentes herramientas informáticas tales como lenguajes de programación, bibliotecas, entornos de desarrollo, compiladores, etc, que se suman a lo más importante que es la comprensión y utilización de herramientas conceptuales, de conocimientos y de estrategias que conforman una manera de hacer y sobre todo una manera de pensar.”*¹⁴ En esta oportunidad me detendré en analizar dos clases en las cuales se pueden observar distintas estrategias que despliegan los estudiantes

¹⁴ La negrita es mía.

en relación a la programación articuladas con el pensamiento computacional y el conocimiento práctico.

¿Qué sabemos del aprendizaje en programación?

Tomando como base la definición de educación en programación mencionada anteriormente, retomo la idea de “**maneras de hacer**” pretendiendo hacer foco a lo que denominaré “conocimiento práctico” o “conocimiento en uso” según Papert (1995) y “**maneras de pensar**” haciendo foco en el pensamiento computacional, como actividad cognitiva que se aplica en la relación con los conocimientos de programación. Esta distinción es puramente explicativa ya que considero que estas dos acciones se relacionan dialécticamente.

Papert (1995) menciona la noción “Conocimiento en uso” como un tipo diferente de conocimiento que se introduce en su contexto de uso, donde uno puede jugar con él. Un conocimiento distinto al fragmentado, que solo puede memorizarse y reproducirse. *“El aprendizaje en uso libera a los estudiantes y les permite aprender de manera personalizada, lo que a su vez libera a sus profesores y les permite ofrecer a los estudiantes algo más personalizado y provechoso para ambos.(p:79)”*

Dentro del aprendizaje de la programación, tomamos la perspectiva construccionista de Turkle y Papert (1990) quienes consideran que existen distintas formas de aproximarse a la programación lo que permite reconocer diversos caminos de conocimientos. Estos autores denominan “Pluralismo Epistemológico”, al igual acceso de los elementos básicos de la computación, aceptando la validez de múltiples formas de saber y de pensar y lo consideran una condición necesaria para una cultura informática más inclusiva. Desde esta perspectiva buscan colaborar en la deconstrucción del pensamiento canónico como única forma de pensar desafiando al pensamiento formal a través de comprender otros estilos en donde la lógica es un instrumento poderoso pero no una ley de pensamiento (Turkle y Papert, 1990 en Losano, 2011).

Articuladamente y en relación al pensamiento computacional, su principal referente Jeannette Wing (2006) lo describe como un pensamiento que proporciona habilidades y competencias intelectuales que constituyen una forma de pensar que tiene características propias y diferentes a las de otras ciencias, como ser: la descomposición en sub-problemas, abstracción de casos particulares, procesos de diseño, implementación y prueba de lógicas algorítmicas, para nombrar las más significativas. En relación a esto, Brennan y Resnick (2012), consideran que el pensamiento computacional incluye tres dimensiones claves: conceptos computacionales; refiriéndose a los conceptos que emplean los sujetos cuando programan; prácticas computacionales, relacionado a las prácticas que se desarrollan a medida que programan y las perspectivas computacionales, como perspectivas que los programadores construyen sobre el mundo a su alrededor y sobre ellos mismos. Puntualmente profundizo en la idea de prácticas computacionales donde se pueden observar a su vez, cuatro conjuntos de prácticas que hacen a este tipo de pensamiento: (1) ser incremental e iterativo, (2) ensayar y depurar, (3) reusar y remezclar y, (4) abstraer y modularizar.

Estrategia y Apropiación.

Considero que los jóvenes despliegan estrategias para acercarse al conocimiento, en esta ocasión acercarse a la programación y apropiarse de él.

Monereo C, Castelló M, Clariana M, Palma M, Perez ML(1999) consideran que la utilización de estrategias de aprendizaje supone un uso reflexivo de los procedimientos que se utilizan para realizar una determinada tarea, que exceden la mera comprensión y utilización de los procedimientos. La estrategia se considera como una guía de las acciones que hay que seguir, y que es anterior a la elección de cualquier otro procedimiento para actuar. Desde este punto de vista, se considera que la calidad del aprendizaje no depende tanto de un supuesto coeficiente intelectual, ni del dominio de un buen conjunto de técnicas y métodos para estudiar con provecho, sino de la posibilidad de captar las exigencias de las tareas en una situación de aprendizaje determinada y controlar con los medios adecuados dicha situación.

Me posiciono desde una perspectiva más amplia de la noción de saber, no solo epistemológica sino también considerando el entramado social y complejo de las relaciones que un sujeto mantiene con el mundo, consigo mismo, con otros: familia, escuela, barrio, etc (Charlot, 2006). Este autor considera que la relación con el saber es una pregunta que debe interrogarnos sobre el sentido y la actividad, considerada esta última como movilización, que pone recursos en movimiento y sentidos como relaciones con el mundo y con otros. Opto por hablar de conocimientos en lugar de saber, ya que lo realizo desde un sentido operacional y relacional ligado a lo escolar y lo académico (Beillerot 1998; Ortega 2001).

Charlot (2006) considera que *“si el saber es relación, el valor y el sentido le vienen de las relaciones que implica e induce su apropiación”* (p: 105) y denomina a la Apropiación como una figura de la relación con el saber, la cual puede ubicar al sujeto como *imitador*, tendiente a determinaciones y conductas heredadas, más cercano a la conformidad, a un saber cerrado, construido y determinado también llamado por el autor como Aprendiz, a diferencia de un sujeto *Autodidacta*, el cual aprende al margen de las normas, donde el saber se presenta como interrogación, considerándolo como un transgresor.

Este conocimiento escolar que pretendo analizar no puede pensarse por fuera de la enseñanza, que si bien no será mi foco, tiene relación directa y articulada en el aprendizaje. Distintas investigaciones, tanto argentinas como internacionales (Ferreira Szpiniak, A; Rojo, G, 2006; Compañ-Rosique, et al, 2015) mencionan la existencia de varios enfoques para enseñar programación: algunos enseñan a programar en un lenguaje de programación particular (Payton, JAVA, C#, entre tantos), utilizando su sintaxis y su semántica; y otros emplean un lenguaje algorítmico lo bastante general cómo para permitir su traducción posterior a cualquier lenguaje de programación, alegando que esta última tendría mayor incidencia en el desarrollo del pensamiento computacional. Particularmente no me interesa centrarme en el aprendizaje de un lenguaje específico sino más bien un aprendizaje de tipo genérico. Sin embargo, no considero que estos dos enfoques sean excluyente, ya que en esta oportunidad analizaré situaciones donde se enseña desde un lenguaje específico pero mi análisis tendrá en cuenta generalidades del aprendizaje en programación.

Situación de aprendizaje en el aula.

No pretendo prescribir sino más bien compartir situaciones de clases que nos permitan pensar y re-pensar la relación con el conocimiento en programación.

Para la primer clase que analizo, tomo un fragmento del registro de la exposición de proyectos de la materia programación de 6to año donde los estudiantes, en grupos, presentaban al final de su trimestre el proyecto social en el que habían pensado y trabajado¹⁵ (Situación 1). En la segunda, tomo fragmentos de la materia “Base de datos” también de 6to año, donde se registra una situación de clase en la que estaban programando su base de datos de manera individual (Situación 2).

Pretendo tomar estos fragmentos como disparadores para pensar el aprendizaje en programación presentando particularidades en cada uno de los registros sin desconocer que se comparten referencias entre las dos situaciones analizadas.

Situación 1:

Estudiante 1 (E1): *Sí, porque teníamos la problemática que por ahí cuando vos querías buscar una casa, y decías una casa por esta zona y no sabías cómo se llamaba el barrio tenías que ir y buscar el barrio. Al tener un mapa te facilita mucho porque vos tenes el mapa de Córdoba y decías, bueno, yo quiero más o menos por acá y podés elegir tu casa, departamento u oficina. Tenemos un lugar donde se ven las 5 casas destacadas. Acá tenes un registro, un botón de contacto. Aca tenes ingresar, tenes un ingreso y un registro. Tenes toda la información, podés tener tus casas favoritas y acá si querés cambiar de data. Acá tenes las casas que se agregaron recientemente, funciona así, si la hora cambió le decis que te muestre las casas...*

Profesor (P1): *Eso lo hiciste con Java Script?*

E1: *Con Django¹⁶ lo hice.*

Estudiante 2: *Agregamos al models una variable que es un date time y después le pones if change y si cambió quiere decir que hay una casa nueva.*

P1: *No sabía eso.*

(...)

¹⁵ Este proyecto era una página de administración de casas, oficinas, departamentos. Trabajaban en grupo de a 4 integrantes donde tenían distintos roles y división de tareas utilizando Trello para esa organización. Esta escuela cuenta con dos profesores por clase, uno como titular y el otro como MET (Maestro de enseñanza técnica)

¹⁶ Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSDen julio de 2005.

P1: Mostrame el código ese que hiciste para el if change. Quiero ver que hace el if change. Profesor 2: Qué es, como un escuchador?

P1: No, eso solo es cuando renderizas el template

E1: Yo no me fijé mucho, leí change= cambio , lo tiré, funcionó y chau.

PJ: En realidad lo que hace eso es...

Estudiante 2: A ver como se usa propiamente.

Risas

P1: cuando tienen una iteración de algún tipo, por ejemplo tienen una iteración de una persona ordenada por edad, cada vez que la edad de las personas cambian, entra ahí adentro. Entonces cuando cambia de 6 años a 7 cambia, hago que todo lo de 7 entre ahí y cuando vuelve a pasar a 8 entra ahí. Entonces yo podría poner un if change arriba y decir que si cambia la edad yo pongo un título: Los de 7 años y entran todos los de 7 años, cambié, sí y así los de 8 años. No saben lo importante que es eso cuando yo hago un informe y tengo que mostrar por ejemplo, lo agregan por fecha y yo quiero mostrar la sumatoria total por día, si?, tengo que sumar ventas y las quiero hacer por día, entonces cada vez que me cambie el día yo tengo algo que me está diciendo cambio el día, sumá y poné el total. Eso tuve que hacerlo muchas veces a mano cuando tuve que hacerlo para la mutual. Tenemos un sistema bastante grande con el profe que todos esos form los tuve que hacer a mano y estar comparando si se guardó la anterior. Ay! Cada vez estoy más contento con la decisión de usar Django.

En este fragmento podemos identificar varias cuestiones que hacen a la relación de los jóvenes con el aprendizaje de la programación. Este proyecto surge de una necesidad que los estudiantes registraron en su entorno y a partir de ello, comenzaron a crear y programar una plataforma como proyecto de clase. Esta situación nos remite a la idea de aprendizaje basado en problemas (ABP) el cual se centra en cuestiones significativas para los alumnos donde investigan y proponen soluciones. Barrows (1986) lo define al ABP como *un método de aprendizaje basado en el principio de usar problemas como punto de partida para la adquisición e integración de los nuevos conocimientos*. Esta perspectiva ubica al estudiante en una posición activa, similar al aprendizaje por descubrimiento que propone Bruner (1960), el cual sugiere que los maestros fomenten un tipo de pensamiento inductivo, alentando a los estudiantes a hacer especulaciones basadas en evidencias incompletas y luego confirmarlas o desecharlas con una investigación sistemática.

En el ejemplo mencionado, los estudiantes exploraron el entorno en el que estaban trabajando sumado a razonamientos e intentos que posibilitaron acceder a la función que estaban buscando. Brennan y Resnik (2012) cuando mencionan los subconjuntos dentro de las prácticas computacionales plantean la idea de ensayar y *depurar* como parte del pensamiento computacional y lo consideran como la posibilidad de desarrollar estrategias para manejar y anticipar problemas; realizar diversas prácticas por ensayo y error, transfiriéndolas de otras actividades o apoyándose en otras personas más conocedoras.

Al respecto, Papert (1987) retoma las ideas del constructivismo de Piaget y propone el *construccionismo* como una manera de leer los aprendizajes donde este proceso de ensayar, errar y corregir el error conduce a los aprendices a crear y aprender, lo que él

llama un *proceso de depuración* y menciona que los errores benefician la comprensión de lo que anduvo mal y, a través de comprenderlo, poder corregirlo.

Estas perspectivas centran el foco en un sujeto activo, en términos de Charlot podríamos decir que estos estudiantes se ubicarían más en lugar de autodidactas.

En esta oportunidad considero importante reconocer la postura del docente en el fragmento de clase elegido, ya que se corre del lugar de sujeto supuesto saber y poseedor absoluto del conocimiento y se ubica como habilitador de este conocimiento que es nuevo para alumno pero también para el docente. Un docente que profundiza en la explicación del mismo y que se sorprende por ese descubrimiento de sus estudiantes.

Papert (1995) al relatar y analizar las primeras experiencias de docentes frente a la programación menciona que la conciencia de ser profesores les impedía entregarse por completo a la experiencia de aprendizaje. Una experiencia pensada como una construcción colaborativa, donde los estudiantes puede recibir pero también pueden dar, un proceso de aprendizaje donde el docente no competía con los alumnos sino que contribuía al proceso que ellos seguían.

Beatriz Grecco (2013) tensiona la idea de pensar disociadamente que quien enseña no es pensado aprendiendo y viceversa. Y reconoce que la autoridad que poseen los docentes en el territorio educativo se vuelve autorización cuando se hacen habilitadora de aquello que aún no acontece y que sólo puede desplegarse en ese “entre” de enseñantes y aprendientes.

Situación 2:

El docente¹⁷ comienza la clase retomando la actividad iniciada en las clases anteriores que consistía en programar una base de datos. Puntualmente ese día, estaban intentando hacer las flechas a sus programas y su consigna consistía en hacer una base de datos sobre una biblioteca virtual, donde la propuesta era individual pero la misma para todos. De esta manera cada alumno empieza a trabajar en su computadora y si bien todos están tratando de agregar las flechas cada uno lo hace de manera diferente. Presentaré distintos fragmentos del registro en donde se muestran distintas actividades que realizan los estudiantes en relación a esta misma propuesta e intentaré tensionarlo con estrategias que hacen al aprendizaje de la programación.

En el caso de la alumna Teresa¹⁸, ella empieza a copiar y pegar pedazos de un código que ya está escrito. Le pregunto qué está haciendo, ya que estaba sentada al lado y me dice que está adaptando algo ya escrito a lo que tiene que hacer ahora. Prueba para ver si le sale pero no lo logra. Entonces vuelve al código anterior y sigue probando. Intenta, aparecen las flechas pero no de la forma que ella quería, entonces vuelve al código y continúa. Cada vez que prueba va mejorando. Luego de unos minutos me dice: lo hice y sonrío.

¹⁷ En esta escuela, los docentes no trabajan en las horas de clases con los maestros de enseñanza práctica, por lo tanto siempre están solos frente al aula.

¹⁸ Los nombres de los estudiantes han sido modificados para resguardar la confidencialidad de los datos, al mismo tiempo que no se nombrará la institución en cuestión.

Estos intentos que hace Teresa en la realización de su programa, forman parte de una actividad de prueba y error, que mencionamos en la situación anterior, donde el error permite acercarnos a nuevos conocimientos. En la programación muchas veces es la misma práctica la que marca los errores y los aciertos a la hora de hacer funcionar un programa sin necesidad de una supervisión constante, en este caso del docente. El feedback inmediato que nos posibilita la computadora favorece no solo la exploración y descubrimiento de los contenidos sino que les permite avanzar y otorgar autonomía y confianza a sus alumnos en la aprehensión de los conocimientos. En una experiencia de trabajo en clases Díaz (2006) referencia que la mayoría de los textos de programación prestan en general poca atención a principios didácticos, presentando los contenidos en un orden conceptual y desconsiderando los principios metodológicos de solución de problemas o cuestiones de depuración de los errores, que son instancias probadamente problemáticas para los novicios en la programación.

En los subconjuntos mencionados por Brennan y Resnik (2012) dentro de las prácticas computacionales se encuentra la actividad de *reusar y remezclar*. Y la reconocen como la posibilidad de construir, no solo sobre lo hecho por uno mismo sino también sobre lo que otros ya han realizado, posibilitándoles potenciarse para crear cosas más complejas. Consideran que reusar y remezclar apoya el desarrollo de las capacidades de lectura crítica de código.

Otra situación registrada en la misma clase, muestra que el docente deja que los alumnos avancen en sus programas y pasa por sus lugares a observar cómo lo están realizando, de esta manera cada alumno recibe sugerencias particulares de su trabajo. A su vez, considerando que cada alumno tiene un ritmo de trabajo singular, este docente propone un trabajo colaborativo entre los estudiantes:

Luis, uno de los alumnos, termina su programa y el profesor le pide que le muestre a Marcos para ayudarlo. (...) Este docente en un momento de la clase pregunta si saben algo de los dos chicos que faltaron. Ellos responden que no. Como son una chica y un chico los ausentes ese día, el profesor les dice que los varones expliquen al varón que faltó y las chicas a su compañera. Y el que haga que esos compañeros entiendan tiene un punto más en la prueba. El profesor le dice a Marcos que lo haga, que él necesita puntos y le pregunta por su nota del 3er cuatrimestre. Éste responde que necesita un 7 y luego el docente le pregunta si considera que se lo ganó porque él no regala notas. Marcos le dice que él está trabajando y que sí siente que se lo merece, a lo que el docente agrega que no le diga que se lo merece en telecomunicaciones y redes porque le ha entregado hojas en blanco.

Esta situación nos permite reconocer una forma particular de estructurar su clase que responde a un estilo docente, donde la intención aquí remite a un trabajo colaborativo proponiendo la estrategia de darle, al estudiante que ya finalizó, la tarea de ayudar a su compañero que presenta dificultades en el tema propuesto. Si bien en esta oportunidad la agrupación no surge de los mismos estudiantes es común observar en las demás clases que se consultan entre ellos cuando alguno no puede continuar con su programa. Meirieu (1998) al respecto, propone renunciar a querer formar grupos homogéneos debidamente preparados para seguir un tratamiento estandarizado y considera pertinente afrontar la heterogeneidad en el mismo grupo de trabajo, tal como se manifiesta ante una tarea y sobre todo ante una situación problema. En relación al trabajo en equipo, Perrenoud (2007) toma los aportes de Gather Thurler, quien distingue como una de las competencias necesarias para este tipo de trabajo, el saber discernir los problemas que requiere una cooperación. Trabajar en equipo es saber hacerlo en el momento oportuno, cuando resulta más eficaz. Es participar en una

cultura de cooperación, estar abierto a ella, saber encontrar y negociar las modalidades de trabajo óptimas, en función de los problemas que resolver. Este autor, expresa en este sentido, que la enseñanza mutua no es una idea nueva, sino que está inspirada en Lancaster, donde el profesor tenía cien o doscientos alumnos a su cargo, de todas las edades, y evidentemente no podía ocuparse de todos ni proponer una lección a un público tan amplio y heterogéneo. Por ello, en estas condiciones organizaba subconjuntos a cargos de alumnos mayores bajo la responsabilidad de «submaestros» sin formación pedagógica. Sin embargo el papel del maestro no quedaba relegado sino que consistía en hacer funcionar el conjunto, más que enseñar directamente a todos (Echeveste y Martínez, 2016). De esta manera se puede observar un “venir entre”, donde se articulan los alumnos y el conocimiento y en donde el docente se ubica como mediador y habilitador de esos contenidos. En esta postura se observa que hay un reconocimiento del otro como sujeto de aprendizaje y enseñante a la vez.

Verónica Edwards (s/d) plantea el concepto de “forma de conocimiento” para describir la existencia material y social del conocimiento en la escuela, donde considera las dimensiones de interrelación entre los docentes y los alumnos, cómo son presentados los conocimientos, el contenido de la clase, la forma de enseñanza, y más. Una dimensión que constituye esta forma de conocimientos es la *Lógica de la interacción*, refiriéndose a los modos de dirigirse alumnos y maestros, unos a otros tanto implícito como explícito. Esto evidencia una determinada lógica de la enseñanza y de la participación formal de los alumnos.

De esta manera, el conocimiento escolar es transmitido, construido y reconstruido por los sujetos en la práctica escolar, en donde se cruzan estilos docentes y lógicas de conocimientos que hacen que las prácticas escolares sean singulares, casuísticas y culturales.

Conclusión

De esta manera y presentando el análisis de estas dos situaciones de clases en las que jóvenes aprenden a programar pretendo sumar a las investigaciones sobre el aprendizaje de la programación. En este artículo aparecen nociones como la exploración mediante el ensayo y el error, el feedback que realiza la computadora, el trabajo colaborativo entre estudiantes, la habilitación de los docentes y la reutilización de código, que hacen que los estudiantes utilicen estrategias y construyan una relación particular con este tipo de conocimiento, que es la programación. A su vez, en ambos registros, los estudiantes se ubicaron como sujetos activos en el conocimiento y partícipes en la construcción del mismo.

Considero que las propuestas educativas donde los docentes autoricen los conocimientos de sus estudiantes, trabajen con proyectos que convoquen e interesen a los jóvenes y permitan un trabajo colaborativo, fomentarán un aprendizaje significativo y genuino de los contenidos de programación. Analizar situaciones escolares que presenten las estrategias que utilizan los jóvenes para acercarse estos conocimientos sumado a propuestas de trabajo que potencien sus aprendizajes, aportarán al debate sobre la didáctica en la Computación. Debates necesarios si tenemos en cuenta que la programación cada vez se introduce más en nuestra cotidianidad transformándose en un conocimiento necesario para desarrollar un ciudadanía plena.

Bibliografía

- Aloi, F., Bulgarelli, F., Palumbo, N., & Spigariol, L. (2016). Corrección automatizada de programas como recurso pedagógico. En XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).
 - Barrows, H.S. (1986). A Taxonomy of problem-based learning methods, en *Medical Education*, 20/6, 481–486.
 - Beillerot, J. (1998). Los saberes, sus concepciones y su naturaleza. En J. Beillerot,
- C. Blanchard Laville & N. Mosconi (Eds),
- Saber y relación con el saber (pp. 43-78). Ed. Paidós. Buenos Aires
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada (pp. 1-25).
 - Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia*, (46).
 - Charlot, B. (2007). La relación con el saber. Elementos para una teoría. Ed. Libros del Zorzal. Argentina.
 - Díaz, J. (2006). Enseñando programación con C++: una propuesta didáctica. *Revista de Informática Educativa y Medios Audiovisuales*, 3(7), 12-21.
 - Echeveste, M. E., & Martínez, M. C. (2016). Desafíos en la enseñanza de Ciencias de la Computación. *Virtualidad, Educación y Ciencia*, 7(12), 34-48.
 - Edwards, V. (s/d) “El conocimiento escolar como lógica particular de apropiación y alineación”. Mimeo.
 - Ferreira Szpiniak, A., & Rojo, G. A. (2006). Enseñanza de la programación. TE & ET.
 - Greco, B (2007) “La Autoridad (pedagógica) en cuestión. Una crítica al concepto de autoridad en tiempos de transformaciones.” Ed Homo Sapiens. Rosario. Argentina
 - Meirieu, P. (1998) *Frankenstein educador*. Barcelona. Laertes.
 - Monereo C, Castelló M, Clariana M, Palma M, Pérez M.L (1999) *Estrategias de enseñanza y aprendizaje: formación del profesorado y aplicación en la escuela* (Vol. 112). Graó.
 - Losano, A.L (2011) *Procesos Situados de Aprendizaje en cursos básicos de programación: volverse miembro de una comunidad* (Tesis Doctoral en Ciencias de la Educación) Facultad de Filosofía y Humanidades. UNC. Córdoba.

- Ortega, F. (2001) Atajos. Saberes escolares y estrategias de evasión. Ed. Miño y Dávila. Buenos Aires.
- Papert, S. (1995). La máquina de los niños: replantearse la educación en la era de los ordenadores. Paidós.
- Papert, S. (1987). Desafío a la mente. Computadoras y educación. Buenos Aires: Galápagos.
- Perrenoud, P. H. (2004). Diez nuevas competencias para enseñar. Barcelona, España: Graó
- Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. Signs: Journal of women in culture and society, 16(1), 128-157.
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.