

Desafíos en la enseñanza de Ciencias de la Computación

Challenges of teaching Computer Science

María Emilia Echeveste y M. Cecilia Martínez

Consejo Nacional de Investigaciones Científicas y Tecnológicas.
Facultad de Matemática, Astronomía y Física, y Facultad de Filosofía y Humanidades,
Universidad Nacional de Córdoba, Argentina.
E-mail: meecheveste@gmail.com; cecimart@gmail.com

Resumen

En este artículo presentaremos un análisis de distintas situaciones de aprendizaje y enseñanza que se registraron en escuelas primarias y secundarias a partir del desarrollo de clases de programación desde el enfoque de la enseñanza por descubrimiento. Específicamente trabajaremos sobre los datos obtenidos durante los dos años en que nuestro equipo ofreció un curso de formación docente en didáctica de la programación. Estos datos incluyen pre y post encuestas a docentes que participaron del curso y trasladaron lo aprendido a sus aulas como así también, registros de campo y observaciones de clases realizadas por tutores -especialistas en Ciencias de la Computación- que asistieron a las clases de los docentes para acompañar sus prácticas en el aula. A partir de esta información, se analizaron los datos obtenidos y se observaron recurrencias que permitieron repensar las experiencias de enseñanza de los docentes a la hora de introducir las CC en las escuelas. Cuatro situaciones se reportan recurrentemente en ambos años, las cuales, en lo que respecta a la enseñanza y el aprendizaje de la computación, desafían la estructura tradicional de la escuela formal.

Palabras claves: educación en Ciencias de la Computación; programación; gramática escolar.

Abstract

In this article, we will analyze different teaching and learning situations identified in primary and secondary schools when delivering inquiry based computer programming classes. Specifically, we will work on data obtained during the two years that our team offered a teacher professional development course in teaching computer programming. These data includes pre and post teachers' survey who participated in the course and transferred what they learned in the professional development to their classrooms. Tutor's -Computer Science advance students- who attended the lessons and supported teachers with the implementation of teaching programming, observed and recorded the lessons. We triangulated classroom observations with pre and post teacher survey and identified different experiences when introducing the teaching of computer programming in schools. We identified four situations along these two years. These situations challenge the traditional structure of formal schooling.

Key words: Computer Science education; teaching programming; school grammar.

Fecha de recepción: Marzo 2016 • Aceptado: Mayo 2016

ECHAVESTE, M.E. y MARTÍNEZ, M.C. (2016). Desafíos en la enseñanza de Ciencias de la Computación. *Virtualidad, Educación y Ciencia*, 12 (7), pp. 34-48.

Introducción

La tecnología ha logrado tener un gran protagonismo en la vida de las personas. Casi sin darnos cuenta nos hemos rodeado de computadoras que poco a poco empezaron a formar parte de nuestra cotidianeidad: teléfonos inteligentes, cajeros automáticos, lavarropas, etc. En este escenario, la escuela, encargada de alfabetizar a la ciudadanía en saberes socialmente válidos, debe acercarse a los estudiantes a la comprensión del mundo digital que nos rodea (Busaniche, 2011).

Introducir conocimientos novedosos en las escuelas, tales como la alfabetización digital, nos remite a pensar en las condiciones organizativas y culturales de las mismas. Reconocidos investigadores nacionales (Alliaud, 2004, Jacinto y Terigi 2007 y Dussel 2011) han mostrado la necesidad de repensar la estructura rígida y tradicional de la escolarización. Esta estructura se caracteriza por la enseñanza de los conocimientos que se dividen en bloques cortos de tiempo, agrupando estudiantes de manera homogénea por la edad, considerando al aprendizaje dictado magistralmente por un docente y evaluando, la mayoría de las veces, el rendimiento individual a través de exámenes.

La enseñanza de la computación, más específicamente de la programación, presenta particularidades que desafían este régimen académico estándar: existe un feedback inmediato que ofrecen las computadoras, la colaboración necesaria entre los estudiantes, la posibilidad de trabajar entre distintas edades, tiempos prolongados de trabajo orientados a un objetivo concreto y el papel activo de los alumnos, serían elementos que se observan en las prácticas de enseñanza y tensionan la gramática escolar¹ tradicional.

Analizando los datos trabajados, se identifican cuatro situaciones que desafían la organización tradicional de trabajo en el aula a la hora de llevar conocimientos específicos de programación. Estos son: la inclusión de jóvenes considerados “problemas”, el saber práctico de la programación, el tiempo prolongado que lleva la programación y el trabajo heterogéneo y grupal. Sobre estas particularidades nos detendremos en este artículo presentando situaciones que ocurrieron en el aula y que permiten repensar la estructura tradicional de enseñanza y aprendizaje, en este caso sobre conocimientos específicos de la programación pero que sin duda podrán servir para analizar prácticas en otros conocimientos.

Consideraciones teóricas y aportes sobre el tema

La escuela tiene formas de ser y hacer que la caracterizan como tal, disposiciones que la constituyen como institución. En este caso tomaré la noción de Tyack y Cuban (2001) sobre gramática escolar para referir a un conjunto de tradiciones y regularidades institucionales sedimentadas a lo largo del tiempo, transmitidas de generación en generación por maestros y profesores; modos de hacer y de pensar compartidos, aprendidos a través de la experiencia. La gramática escolar tiene una importante participación en el diseño de la organización de la enseñanza y el aprendizaje, por lo tanto, incorporar nuevos conocimientos requiere tener que “negociar” entre lo instituido y lo instituyente.

Antes de profundizar y definir cada una de las dimensiones de análisis trabajadas es necesario

1 Término que desarrollaremos más adelante.

describir brevemente en qué consiste la enseñanza de la enseñanza de conceptos en Ciencias de la Computación, para comprender de esta manera las posibles derivaciones que tiene para abordar esta disciplina en la gramática escolar.

La enseñanza de la programación

En la década de los 90, se desarrolla la industria de las computadoras personales y se consolidan los grandes monopolios internacionales como: IBM, Intel, Microsoft, Apple. Estos procesos permearon las políticas educativas que igualaron la enseñanza de computación en las escuelas al adiestramiento en los productos que desarrollan estas empresas. Estas empresas privadas se encargarían de la formación de los docentes de enseñanza básica y media de Argentina con una fuerte concepción operativa e instrumental de la tecnología en detrimento de los conceptos propios de la disciplina Ciencias de la Computación, brindando así la ilusión de estar formando a los jóvenes y niños en esa área. Este movimiento fue denominado por Levis (2007) como “enfoque utilitario”.

De esta manera, no será lo mismo enseñar Ciencias de la Computación (CC) y más específicamente programación, que Enseñar Tecnología o introducir las TIC al aula. Nos detendremos en explicar con mayor detalle lo que corresponde a las CC.

En términos generales, introducir las TIC en el aula se relaciona a presentar dispositivos electrónicos, aplicaciones informáticas e Internet, entre otras cosas, como herramientas aplicadas en la escuela a cualquier área de conocimiento. Enseñar Tecnología excede la mera utilización y refiere a un conocimiento más profundo de los procesos y mecanismos del funcionamiento y desarrollo de cualquier tecnología, no sólo de aquellas relacionadas con la Computación. A diferencia de éstas, las Ciencias de la Computación profundizan aspectos propios de una disciplina. Esta ciencia, cuenta con fundamentos, principios, conceptos y métodos independientes de tecnologías concretas: incluye algoritmos (métodos para describir soluciones a problemas, realizando instrucciones bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos) y estructuras de datos (formas de organizar datos en la computadora) que se materializan en programas que ejecutan sobre determinadas arquitecturas (modelo y descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora) y que pueden requerir estar en red (formas de vincular las computadoras para que puedan establecer comunicaciones entre ellas). (Fundación Sadoksy, 2013).

El dominio de estas áreas conceptuales, promueve lo que algunos autores han denominado el pensamiento computacional (Wing, 2006) 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. Acciones que forman parte del hacer y del pensar de los que trabajan en CC.

Si sostenemos entonces que aprender CC es apropiarse de conceptos y competencias propias de la disciplina, no es suficiente dotar a las escuelas con computadoras o con acceso a Internet, también es necesario trabajar en la formación docente y en la formulación de nuevos repertorios de prácticas que permitan hacer una producción y construcción con la tecnología estableciendo usos

más complejos y significativos de los medios digitales.

Dussel y Quevedo (2010) consideran que las nuevas tecnologías tienen lógicas y modos de configurar el conocimiento muy diferente a los de la escuela tradicional. Las nuevas tecnologías funcionan en base a la personalización, el involucramiento personal y suelen ser muy veloces y con una interacción inmediata. La escuela, en cambio, es una institución basada en el conocimiento disciplinar, más estructurada, menos exploratoria, y con tiempos y espacios determinados previamente, más lentos y menos porosos. Por ello, éstos autores plantean que se debe dar un proceso de negociación y reacomodamiento de la institución escolar que no será automático ni inmediato, y que incluso no debería ser leído sólo como resistencia al cambio.

Para encontrar un camino que nos permita adaptar estos cambios, Perrenoud (2007) considera que una forma de acercarnos a un punto medio entre la enseñanza frontal² y la enseñanza individualizada sería organizar el trabajo en clase de forma distinta, quebrar con la estructuración en niveles anuales, facilitar la comunicación horizontal, crear nuevos espacios y tiempos de formación, entre otras cosas.

A continuación profundizaremos sobre dimensiones teóricas referidas a las cuatro recurrencias ya mencionadas, encontradas en las aulas y que desafían la gramática escolar a la hora de enseñar programación. Dichas recurrencias son: 1) la organización del tiempo en la escuela, 2) los modos de incluir a alumnos diferentes 3) el trabajo heterogéneo y grupal y 4) el saber práctico en la programación.

1) La organización del tiempo escolar.

Jacinto y Terigi (2007) plantean que el tiempo constituye una variable sustantiva para atender los esfuerzos de mejorar la educación. Estas autoras, sugieren como estrategia, mirar el régimen académico que regula la organización de las actividades de los alumnos y las exigencias a las que éstos deben responder. Específicamente en los conocimientos referidos a la computación, los programadores experimentan una particular relación con el tiempo, muchos de ellos necesitan permanecer trabajando de forma continua por horas, a diferencia de otros trabajos donde los horarios pueden ser más cortos y regulares. Parte de los programadores, incluso perciben una sensación de “fluir en el tiempo” cuando están trabajando frente a la computadora (William, 2007).

Si pensamos en la gramática escolar, el tiempo de trabajo en el aula es un punto importante para repensar la organización de la escuela y específicamente la incorporación de la programación. En Argentina, con la intención de mejorar la inclusión, se establecieron en la educación secundaria formatos escolares alternativos. Una de las características de estos formatos es una organización distinta a la tradicional en cuanto al espacio y al tiempo de trabajo, utilizando dispositivos curriculares y organizacionales que garanticen el trabajo pedagógico y los aprendizajes a través de talleres, tutorías, clases de apoyos, entre otros (Jacinto, 2009). De esta manera, creemos que esta organización del tiempo facilita que se establezca otra relación con el conocimiento, distinta a la tradicionalmente conocida. Según los datos que analizamos, muchos docentes que realizaron nuestros cursos de programación, organizaron instancias de tiempo prolongado.

2 Término que Perrenoud usa para hacer referencia a una pedagogía que utiliza la misma lección, los mismo ejercicios para todos. (2005 p:42). También se la suele llamar Pedagogía Tradicional.

2) Interés y acercamiento de los jóvenes a la programación.

Es característico asociar a los programadores o estudiantes de carreras de programación con estereotipos como “nerds”, inteligentes, varones, clase media, que usan lentes y demás. Distintos estudios (OEI³. Observatorio de la Ciencia, la Tecnología y la Innovación, 2009; Fundación Sadosky, 2013, Martínez y Echeveste, 2014) evidencian que las representaciones que los jóvenes tienen sobre los científicos y más específicamente sobre los que trabajan en CC muestran una recurrencia en considerarlo difícil o para inteligentes, como encualquier ciencia exacta. Los estudios de la OEI arrojaron que un 40% de estudiantes consideraban que un científico de las ciencias exactas es alguien distinto, con una inteligencia superior, y plantearon que la mayoría de los jóvenes finalmente no contempla la posibilidad de dedicarse a la investigación científica a la hora de imaginarse su futuro profesional. Problemática preocupante en lo que respecta a la necesidad de programadores para el desarrollo del país⁴

La escuela, como gran socializadora que es, juega un papel importante frente a los prejuicios sociales, los que muchas veces funcionan como veredictos condenatorios en relación al presente y al futuro de los estudiantes (Kaplan, 1994). Las expectativas que los docentes depositan en sus alumnos tienen un gran impacto en la subjetividad de los mismos y la mayoría de las veces, estas expectativas, condicionan, no sólo la relación docente-alumno, sino que pueden predecir y producir comportamientos. Por lo tanto, considerar que determinados alumnos pueden acceder o no a ciertos conocimientos sería desconocer, entre otras cosas, las distintas trayectorias escolares (Terigi, 2007) que se ponen en juego a la hora de transitar la escuela.

Si sostenemos la necesidad de repensar la gramática escolar y adecuar el curriculum del aula a la diversidad de los alumnos, se supone llevar a cabo modificaciones asumidas para un grupo de enseñanza y aprendizaje concreto. Gil y Sanchez (1999) consideran que adecuar las dinámicas del aula tienen una doble vertiente, por un lado las condiciones idóneas que permiten desarrollar programas de clases que pueden incluir a alumnos que precisen necesidades educativas especiales y por el otro garantizar que los sujetos que requieren adaptación no sean ajenos al grupo de aprendizaje sino que participen activamente en las actividades comunes para todo el grupo en la medida de sus posibilidades.

3) Trabajo en Grupo y Participación Heterogénea.

La escuela históricamente ha tenido la tendencia homogeneizadora de agrupar a los estudiantes por edad generando la fantasía de que todos aquellos alumnos aprenderían de la misma manera las mismas cosas, dejando por fuera cuestiones como los intereses, las motivaciones, la maduración, el desarrollo, etc. De esta manera, pensar en una pedagogía que atienda personalmente a cada alumno caería en una propuesta opuesta a la anterior pero igual de fantasiosa. Perrenoud (2007) menciona que la solución no es transformar la clase en una serie de relaciones duales, donde el profesor se encargue

3 Organización de Estados Iberoamericanos

4 Según los datos del observatorio de la CESSI (cámara que nuclea las empresas del sector, la industria del software y servicios informáticos) se requiere de un mínimo de siete mil profesionales al año, mientras que el sistema universitario y terciario del país gradúa apenas tres mil quinientos.

de cada alumno, uno detrás del otro, sino buscar estrategias de enseñanza y aprendizaje que puedan abordar las diferencias. Una de ellas tiene que ver con el trabajo en equipo. Sin que esto constituya una solución milagrosa, resulta interesante pensar una cooperación entre alumnos.

Mencionado autor, expresa en este sentido, que la enseñanza mutua no es una idea nueva, sino que está inspirada en Lancaster⁵, 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.

Complementario a esto, Meirieu (2001) 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.

De esta manera, se ha observado una gran disposición de los alumnos que aprenden Ciencias de la Computación por medio de un trabajo grupal e incluso heterogéneo en edad. Dos cuestiones, que como vimos anteriormente, que desafían el régimen académico rígido y tradicional.

4) El saber desde la práctica.

Jacinto (2009) considera que hay argumentos y evidencias de investigaciones que señalan que se genera una mayor motivación e interés en los jóvenes cuando los procesos de aprendizaje parten de saberes prácticos para luego desde allí plantear saberes teóricos, o lo que se conoce como el valor pedagógico de la formación orientada.

Una de las particularidades de los conocimientos en programación es su fuerte relación con el “saber hacer”. Investigadores de la Universidad de Quilmes (Lopez M, P; et al, 2014) observaron que es posible que los alumnos adquieran un buen nivel de entendimiento, manipulación en abstracto de los conceptos centrales de la programación, y comprensión analítica de la semántica del código que se escribe para programar, por medios derivados de la práctica.

La Universidad de Quilmes apuesta a un enfoque práctico para introducir la enseñanza de la programación a sus alumnos. Lopez y sus colegas, han basado su estudio en presentar un enfoque de trabajo que explicita un conjunto de conceptos pilares para la introducción en Ciencias de la Computación. Si bien su enfoque fue pensado para la educación inicial universitaria mencionan que sería muy provechoso que se aplique a la enseñanza de la programación a nivel secundario. Su propuesta de trabajo considera comenzar a trabajar en programación, con una formación de tipo más técnica que científica y plantea que los alumnos deben ser capaces de poder aplicar las ideas, reconociendo dónde y por qué son necesarias, utilizándolas de manera pertinente y crítica aunque su

5 Joshep Lancaster, fundó en 1798 una escuela primaria en Southwark (Londres) utilizando una variante del Sistema de Enseñanza Mutua, también conocida como lancasteriana, establecida primeramente en Madrás (India) en 1796 por Andrew Bell.

meta por el momento no sea teorizarlas.

En el desarrollo de este trabajo profundizaremos sobre estas cuatro dimensiones reflejadas en las dinámicas del aula entre alumnos y docentes que pasaron por la experiencia de acercarse a los conocimientos de Ciencias de la Computación.

Intervención y metodología

Propuesta de intervención

En esta oportunidad trabajamos con los datos obtenidos de dos intervenciones de formación docente realizadas en los años 2014 y 2015⁶. Ambas pertenecen a un programa de enseñanza de las Ciencias de la Computación que aborda tres ejes: formación docente, desarrollo de materiales didácticos y diseño y realización de pruebas piloto de experiencias de enseñanza de las CC en escuelas primarias y secundarias. El equipo de trabajo que coordina este programa se llama UNC++ y trabaja interdisciplinariamente desde 2013. Está radicado en la Universidad Nacional de Córdoba con sede en la Facultad de Matemática, Astronomía, Física y Computación (FaMAF) integrado por docentes y egresados de Computación, Educación, Psicología y Trabajo Social.

Tanto en 2014 como en 2015, se ofreció un curso de capacitación a docentes con planes de clase elaborados desde la perspectiva del aprendizaje por descubrimiento para enseñar programación a partir tres plataformas: Alice para programar animaciones y videojuegos (Cooper, Dann y Pausch, 2003), Chatbot para programar autómatas de chat (Benotti, Schapachnik y Martínez, 2014) y en 2015 se agregó UNC++Duino, una plataforma que permite programar robots (Martínez, Gómez y Benotti, 2014), éstas dos últimas realizadas desde el equipo UNC++ y las tres de libre acceso. También se utilizaron actividades de enseñanza de computación sin computadoras del programa “CS Unplugged”.⁷

Ambos cursos tenían un segmento teórico dictado en la universidad a través de talleres y un segmento de aplicación de actividades prácticas que los docentes debían ejecutar en sus respectivas escuelas. Para la instancia de práctica en el aula, la universidad dispuso de tutores previamente capacitados por el equipo, que visitaba cada una de las escuelas durante una jornada, y colaboraba en el seguimiento y acompañamiento de la propuesta con cada uno de los docentes.

El curso de capacitación abordaba algunos conceptos fundamentales de programación: secuencias, condicionales y ciclos; variables, objetos y atributos; funcionales, métodos; entre otros. Como así también competencias propias de la disciplina tales como comentar el código, auto descubrimiento de una interfaz de usuario, trabajo en grupo, etc.

6 Ambos cursos han sido gratuitos. En 2014 financiado por Google Rise Award y en 2015 por el Ministerio de Ciencia de la Nación.

7 CS Unplugged (<http://csunplugged.org/>) es un conjunto de actividades de aprendizaje que enseñan Ciencias de la Computación adaptando juegos y rompecabezas que utilizan tarjetas, cuerdas, lápices de colores sin necesidad de tener una computadora cerca. El material está disponible de forma gratuita, y es compartido bajo una licencia Creative Commons.

La modalidad de trabajo que utilizamos en la capacitación docente fue el enfoque de Aprendizaje Basado en Problemas (Torp y Sage, 1998). Se planteaban desafíos, como parte de una secuencia didáctica que requería que los docentes investiguen y exploren la posible solución a esa situación que la mayoría de las veces tenía relación con temas cotidianos o cercanos. A medida que los docentes iban descubriendo el camino que los llevaba a la resolución se generaba la necesidad de saber más sobre ese tema y es en ese momento se introducían nuevos conocimientos que eran explicados por los docentes.

Muestra

Los docentes inscriptos al curso eran auto seleccionados, ya que se acercaron por interés propio, y pertenecían a escuelas públicas y privadas, primarias y secundarias de la capital y el interior de Provincia de Córdoba (Argentina). En ambos cursos docentes, para acompañar las horas prácticas, se asignó el acompañamiento de un tutor, el cual asistía 4 horas en el aula con cada docente.

Durante 2014, se capacitó a 38 docentes de 19 escuelas y en consecuencia 667 alumnos. Como registro de trabajo se realizaron pre y post encuestas para obtener datos. En total se consiguieron 263 encuestas de alumnos y 26 de docentes, lo que permitió comparar datos.

En el curso de 2015 se capacitó a 60 profesores de 26 escuelas, analizándose registros de 43 clases observadas las cuales permitieron encontrar recurrencias y emergentes también halladas en 2014.

Análisis

Se realizó un estudio exploratorio construyendo categorías desde la identificación de patrones. Este tipo de estudios, estudia un fenómeno nuevo, como lo es la enseñanza de las CC en la escuela, del cual hay pocos antecedentes de investigación que permitan establecer hipótesis, categorías analíticas o teorías explicativas con el propósito de ganar en la comprensión del fenómeno y contribuir a sentar las bases teóricas, e identificar variables que permitan orientar futuros estudios (Cuthill, 2002). Creswell (2013) diferencia entre estudios exploratorios y confirmatorios y considera que los estudios exploratorios buscan construir categorías a partir de recurrencias en vez de corroborar en diferentes contextos categorías previamente construidas. Tanto los nombres de las escuelas como de los alumnos permanecerán confidenciales.

En esta oportunidad se realizó un análisis de datos de las encuestas pre y post tests tanto de docentes como de alumnos obtenidas en 2014. Algunas con preguntas cerradas y otras abiertas que nos permitieran estudiar los discursos de los jóvenes y los docentes. Estas preguntas fueron repetidas antes y después de la intervención para identificar cambios en los relatos. Como la mayoría de los datos fueron recolectados con preguntas cerradas, de múltiple opción, o de respuestas breves, se trabajó en planillas de cálculo donde cada columna guardaba las respuestas y cada fila el nombre del participante. Se crearon tablas de frecuencia para cada una de las variables.

Para obtener la palabra tanto de docentes como de alumnos, en 2015, se utilizaron los registros de campo de las clases observadas por los tutores. A partir de estos datos se buscaron recurrencias en las características encontradas determinando emergentes que se complementen a la hora de hablar sobre cómo aprenden los estudiantes en Ciencias de la Computación.

En base a los datos de las encuestas se construyeron categorías de manera inductiva. Estas categorías emergentes luego se cruzaron con el análisis de las observaciones realizadas en 2015 que aportan un plus a los datos comparativos registrados en 2014. La observación de las clases permitió aproximarse al punto de vista de los sujetos en estudio, compartiendo con ellos sus experiencias cotidianas; y a la misma vez, la contrastación entre el discurso oral o escrito y lo que se hace en la práctica concreta.

Análisis y resultados

Se presentarán las cuatro situaciones que se observan como recurrentes en el análisis de los datos, las cuales permiten desafiar la estructura tradicional de la escuela formal.

Inclusión e interés de los jóvenes en las clases de Ciencias de la Computación.

Según los datos analizados en este artículo se observaron situaciones que tensionan el supuesto que sólo quienes son “inteligentes” pueden aprender a programar. Así mismo, jóvenes catalogados previamente por el sistema o por sus docentes como “los que nunca se enganchan”, “los repitentes”, “los niños con retraso” han logrado un desempeño que no sólo sorprendió a sus docentes sino que superó el desempeño del resto de sus compañeros.

Teniendo en cuenta las encuestas realizadas a docentes en 2014, ante la pregunta sobre cuáles habían sido resultados no esperados de las clases de programación, un 43% consideró que “aquellos alumnos que generalmente no demuestran interés en la escuela estuvieron muy comprometidos con la tarea”. Así mismo, profundizando en los registros de campo de la práctica docente en 2015 hemos encontrado recurrencia en comentarios, sobre distintos alumnos que tenían un desempeño distinto a la media estándar: niños con integración escolar por Síndrome de Asperger o retraso en el desarrollo cognitivo, niños señalados como los repitentes, los revoltosos de la clase o los que con nada se interesaban. Estos niños señalados han tenido un desempeño que sorprendió no sólo a los tutores sino a sus propios docentes, realizando un alto desarrollo de las actividades incluso en la mayoría de los casos superando a sus compañeros tipificados como “normales”. Ejemplos de esta situación se pueden observar en estos fragmentos de los tutores:

(...) hay un chico que se llama Tomás que según el profe se “porta bien” pero tiene problemas atencionales “vive en su mundo” en palabras del profe, y fue éste el que más rápido asimiló Alice y Frozen y trabajaba muy prolijo y detallista. (Escuela Primaria).

Manuel es el más grande y es la cuarta vez que repite cuarto. Y según el profe nunca se prende a nada. El trabajaba solo y fue uno de los que terminó con una animación muy buena y bastante imaginativa. (Escuela Secundaria).

Había otro chico también del cual decían que le costaba trabajar en grupo, pero no hubo dificultades. Es más, como estuve paseando de grupo en grupo, no pude identificar cuál era este otro chico del que me hablaban. (Escuela Primaria).

De esta manera podríamos ver cómo se reactivan distintas capacidades y desempeños en los alumnos, lo cual muchas veces podría resultar novedoso para los docentes. Una hipótesis respecto

a esto podría tener que ver específicamente con los conocimientos de programación, al ser ésta una disciplina relativamente nueva en la educación formal. Muchas veces, los docentes no cuentan con una experiencia como alumnos para utilizar de referencia o sus experiencias distan bastante del objetivo de programar computadoras en vez de usar computadoras. Alliaud (2004) considera importante tener en cuenta el aprendizaje “en situación” como aquello aprendido “informal” o “implícitamente” en la prolongada estadía que uno pasa por las instituciones escolares.

Otras experiencias aúlicas que fueron registradas por los tutores tienen que ver con situaciones que desafían la integración de jóvenes con discapacidades objetivas a los conocimientos de programación. Uno de los casos es el de un niño con Síndrome de Asperger:

Entre el alumnado, había un caso especial: un chico con síndrome de Asperger. Analía ya me había contado un poco de él, y que a veces le costaba la interacción con sus pares o gente nueva. Me decía: “Te vas a dar cuenta de quién es”; y efectivamente me di cuenta, fue el primero que llegó a la clase. (...) al principio se lo vió un poco tímido en la relación, pero luego me sorprendió. (...) Quizás, en algún sentido, le cueste un poco la interacción con sus pares (puede ser algo típico de la enfermedad), pero demostró un interés enorme por aprender, por desafiarse y por conocer más sobre las nuevas herramientas y la tecnología. (Escuela Primaria).

También se registró otra situación con un niño con integración escolar:

Nico, un nene “integrado”, supuestamente tiene cierto nivel de retardo, así me lo plantearon, mientras todos hacían las actividades estaba solo sentado al lado de su grupo (...) entonces le ofrecí acompañarlo a hacerla. Nos sentamos en un costado los dos y la verdad que el nene la hizo de goma literalmente, lo hizo todo solo y super rápido. Es mas en la página 8 del libro, al final hay un ejercicio extra para “expertos”, y como había terminado antes que los otros, le pregunte “te animas a hacerlo?”, me dijo “sí” (con un poquito mas de confianza) y como no le salía muy bien así en el aire el ejercicio, lo represento con lapices de distinto tamaño! Sinceramente quedé sorprendido! (Escuela Primaria).

En este sentido, podríamos pensar que un motivo que permite este interés en los jóvenes en general, sin distinción de capacidades especiales, tendría que ver: por un lado, por el tipo de actividades que se proponen, las cuales escapan de la lógica del régimen académico tradicional al trabajar principalmente con una dinámica práctica y grupal; y por el otro, por la particularidad que tienen los contenidos de programación al generar un feedback inmediato lo que permite trabajar desde la prueba y el error e ir avanzando sin necesidad de supervisión docente constante.

Muchas veces estas concepciones parten de la dificultad que pueden tener algunos docentes para usar la computadora. Sin embargo, en contenidos específicamente innovadores como lo es la programación, Guskey (2002) considera que los docentes logran cambiar las creencias después de que la innovación ha sido probada y ven los resultados en sus alumnos. De esta manera permite que el “asombro” provocado en los docentes se materialice en experiencias de enseñanza concretas.

El saber práctico en el aprendizaje de la programación

Teniendo en cuenta los datos analizados se observó una preferencia de los docentes por la utilización de actividades prácticas a la hora de presentar los conocimientos de programación. Se

consignaron estrategias más cercanas a situaciones habituales, con una metodología de trabajo que favorecía la exploración y el desarrollo de habilidades propias, dejando en un segundo plano las clases expositivas.

Como mencionamos anteriormente, en las capacitaciones docentes dictadas, se hizo énfasis en que pudieran trabajar los contenidos de sus clases desde un enfoque de Aprendizaje Basado en Problemas. Éste método desarrolla una secuencia didáctica significativa para los alumnos, otorgándoles un papel activo en este proceso, donde se investiga y explora proponiendo posibles soluciones a los desafíos planteados. Tanto en los registros de 2015 como en el 2014 se observa una modificación en su modalidad de trabajo en el aula una vez finalizada nuestra capacitación docente.

Según las encuestas de 2014, antes de la intervención de formación docente en CC, el mayor porcentaje de docentes destinaba la mitad de su planificación a clases expositivas. Mientras que los resultados posteriores a la formación, no solo que duplicaron el mayor porcentaje de docentes, pasando de un 14,3% a un 32%, sino que disminuyeron el porcentaje de incorporación de las explicaciones teóricas de un medio a un quinto. En cuanto a la utilización de clases con desafíos, -propuesta con características opuestas a las meramente expositivas-, el 80% fue el porcentaje más alto que los docentes destinan de sus clases a este tipo de modalidad. A su vez, comparando las pre y post encuestas se observó un aumento en la cantidad de docentes que destina este porcentaje de su clase, pasando de un 15,7% en el pre-test a 24% después de haber recibido la capacitación docente.

Si tomamos los registros de 2015, un 74% de los docentes utilizaron la metodología de desafíos, lo cual permitió no sólo una aprehensión distinta a la tradicional de los conocimientos sino que favoreció altamente el trabajo en grupo. En los registros analizados se puede ver que un 82% de los docentes que trabajaron bajo la modalidad de desafíos también han hecho trabajar en grupo a sus alumnos. Esto se puede ver también en algunos fragmentos de los registros de campo:

Muchos tuvieron problemas al compilar o cargar el código al robot: y eso sirvió para que supieran debuggear el código o meterse a la configuración y solucionarlos. Fue espectacular que ellos pudieran corregir sus propios errores (y las profes también se interesaban y cualquier duda preguntaban también). A veces me acercaba a algún grupo para ver cuál era el problema, y me decían: Esto ya lo probamos, esto también, nos da este error, la configuración está bien, el código también porque compiló, pero el robot está haciendo cualquier otra cosa. (Escuela Secundaria).

Como efecto secundario del ejercicio los alumnos vieron la necesidad de aprender Entrada-Salida para poder preguntar el valor al usuario. No lo hicieron por sí solos, pero se enfrentaron un rato a la problemática de cómo setear ese valor en una variable. Además aprendieron un poco de Tipos dado que el menú de variables les preguntaba un tipo para la variable como así también la función que preguntaba al usuario el dato. (Escuela Secundaria).

La programación nos da la posibilidad de introducir conocimientos utilizando la manipulación de programas o plataformas sencillas. Esto permite una exploración individual la cual hace que los alumnos indaguen, busquen, y es en el momento en que se empieza a complejizar donde puede recurrir a un docente, un compañero que conozca un poco más o incluso un tutorial para introducir y equilibrar ese conocimiento. De esta manera se complementan los dos conocimientos, tanto del alumno como del docente, ubicándolos a ambos en un lugar activo y alejándose de las clases magistrales expositivas

donde se brindaba la sensación de una transmisión de conocimiento unidireccional. En este caso, se presenta una complementariedad de los conocimientos, uno más del orden de lo exploratorio y del descubrimiento y el otro desde la experticia.

Siguiendo a Cabello (2006), una de las razones por las que los docentes tienen resistencia a las innovaciones, en este caso referidas a la computación, tiene que ver con el miedo a romper o borrar algo del programa. Sin embargo, otra característica fuerte de la enseñanza de la computación tiene que ver con la prueba y el error constante. Un error, que desde la mirada constructivista de Piaget, permite acercarse a nuevos conocimientos. Muchas veces la misma práctica nos 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 de los programas 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.

Heterogeneidad y trabajo grupal

En lo que respecta al material analizado, tanto en 2014 como en 2015, los docentes han llevado lo aprendido a sus aulas de maneras muy diversas: talleres fuera del horario de cursado, clases integradas con diferentes materias, diversidad de edades dentro de módulos de clases, clases tutoradas por alumnos más expertos o más grandes e incluso alumnos autodidactas. Estas modalidades mostraron una fuerte tendencia al trabajo grupal. A continuación, algunas situaciones registradas en las clases nos permiten pensar el trabajo grupal y su relación de la programación.

Para el día que yo asistí habían preparado una forma de taller, en donde juntaron los dos cursos, pero 3er año no había visto nunca Alice, entonces trabajan en parejas (1 alumno de segundo año y otro de 3ero) entonces el alumno de 2do año era el encargado de acompañar en el aprendizaje al de 3ero, pero con la metodología de descubrimiento. (Escuela Secundaria).

(...) En un momento, me llama un grupo, y uno de los chicos se ofrece a ir al grupo para ayudarlos y sacarle los errores, y entre ellos mismo iban colaborándose! Eso fue bellissimo! (Escuela Secundaria).

Es importante tener presente la importancia de la interacción con otros en la construcción de conocimientos. Uno no aprende solo y el aprendizaje cooperativo es una buena estrategia didáctica para llevar al aula contenidos específicos como lo son los conocimientos en CC. Sin embargo, Perrenoud nos deja una vigilancia a tener en cuenta a la hora de aplicar esta estrategia, ya que trabajar en equipo no consiste en hacer juntos lo que podría hacerse por separado, menos todavía en “mirar hacer” al líder o al alumno más hábil del grupo. Queda en el docente la difícil tarea de hacer funcional el grupo de alumnos. Un ejemplo de esto podría ser esta situación registrada en una de las tutorías docentes:

A los alumnos que lograron resolver el desafío primero los puse a ayudar a sus compañeros de adelante o atrás de manera que el conocimiento también se transmitiera de a pares. Para el final de la hora contaba con 4 tutores más. (Escuela Secundaria Técnica).

En computación, más concretamente en la industria del software, existe una fuerte tendencia a trabajar en equipos y por proyectos. En lo que respecta a la enseñanza de la programación muchas veces los mismos jóvenes tienden a acompañarse sin necesidad de que sea una manifestación expresa. Incluso esos grupos pueden estar conformados no solo de alumnos de distintas edades sino también

incluir profesores que quieran conocer sobre el tema, dejando de lado la fuerte idea de resistencia docente ante la cultura digital.

Se trabajó en 6 grupos de 6 a 7 personas cada uno. Estaban chochos los chicos, y algunos profes se sumaron a esos grupos para aprender también. Creo que estaban igual de ansiosos que los alumnos, y lo bueno es que preguntaban también, y eso me gustó, porque nos sentíamos todos en el mismo nivel y nos ayudábamos mutuamente. (Escuela Primaria).

Estas observaciones permiten mirar con otros ojos la noción de asimetría docente-alumno pensada desde una pedagogía tradicional, permitiéndonos tener en cuenta ciertas particularidades de los grupos con los que se trabaja en el cotidiano del aula.

El Tiempo que lleva la programación

Otra variable que se registró en los datos analizados es el tiempo de trabajo en el aula, la mayoría de los docentes considera que el tiempo estipulado por el régimen escolar no era suficiente para poder llevar a cabo ciertas actividades en las aulas. Punto importante para pensar la organización de la escuela y la incorporación de la programación.

Según los datos recabados en 2014, ante la pregunta ¿Cuáles fueron las mayores dificultades para llevar a cabo las actividades sugeridas en este curso? El 32% (porcentaje mayor de respuestas) hizo referencia al tiempo y los inconvenientes para incorporar estas actividades al régimen académico estipulado. El argumento que más se registró en estas respuestas tenía que ver con las “pocas horas semanales” con las que disponían como así también una docente expresó “no encontrar tiempos reales para llevar a cabo las actividades”

En los datos analizados en 2015, en la variable tiempo, un 58% de los docentes utilizaron modalidades que excedían el clásico módulo de clase, como ser: actividades extraescolares, compartir módulos con otras materias, jornadas talleres, etc. Sin embargo, fue recurrente que el 42% que reportó en sus tutorías haber utilizado sus módulos correspondientes de 80 minutos, expresaron de manera informal falta de tiempo para llevar a cabo y concluir las actividades.

Algunos ejemplos encontrados en los registros de tutorías docentes:

El tiempo nos quedó corto (aunque fue un módulo), y quedaron con ganas de más pero se venía otra materia. (Escuela Secundaria).

Las profes pidieron horas prestadas para que pudiéramos aprovechar las horas hasta las 16. 13:30 comenzamos la clase. (Escuela Secundaria).

(...) a los chicos los habían citado a las 12 para arrancar con la clase, habían juntado horas y gestionado un cambio de actividad para poder estar las 4 horas trabajando los dos cursos juntos en esta actividad especial. (Escuela Primaria).

Es común que los programadores trabajen bajo la noción de proyectos, tanto en la industria del software como en la educación universitaria. En este último ámbito se suele dar una fecha estipulada de entrega y el tiempo en las clases se utiliza como instructivo para la realización del mismo. De esta manera, el trabajo restante lo realizan extracurricularmente.

Específicamente en el dictado de nuestros cursos se puso énfasis en el planteo de desafíos para abordar los temas a trabajar, dichos desafíos se vieron reflejados en el aula en actividades análogas a los proyectos de programación.

Esto nos da la pauta de la necesidad de considerar los desafíos que nos plantea la integración de los conocimientos de programación en la educación formal y la posibilidad de pensar esta modalidad de trabajo adaptado a la gramática escolar.

Conclusión

Este trabajo analiza cómo enseñar y aprender programación presenta retos a la gramática tradicional de la escuela. Los datos muestran que se provocaron cambios en las decisiones y las prácticas pedagógicas tradicionales de los maestros, así como en las actitudes de muchos estudiantes que asumieron una mayor responsabilidad en sus propios procesos de aprendizaje. Es por ello, que se recuperaron, a partir de las recurrencias en el análisis de los datos, cuatro situaciones en las clases: la inclusión de jóvenes considerados problemas, el saber práctico en los conocimientos específicos de CC, el tiempo que lleva la programación y el trabajo heterogéneo y grupal. Es importante mencionar que, las cuatro situaciones analizadas, en ningún momento se pensaron por separado sino como situaciones que se entrecruzan constantemente.

De esta manera, la escuela, como institución histórica y social no puede resistir al avance de la Computación. Sin embargo, la tarea de incorporar la programación, no resulta sencilla, por lo que reflexionar sobre las prácticas que se llevan a cabo en las escuelas nos permitirá lograr avanzar en su implementación. Repensar la gramática escolar es parte de este avance histórico que atrapa a las instituciones y este trabajo pretende colaborar a ello.

Referencia Bibliográfica

- ALLIAUD, A. (2004). La experiencia escolar de maestros inexpertos. Biografías, trayectorias y práctica profesional. *Revista Iberoamericana de Educación*, 34 (3), pp. 1-11.
- BENOTTI, I.; MARTINEZ, M. C.; SCHAPACHNICK, F.; (2014) Engaging High School Students Using Chatbots, en Clear T y Pears A (Eds) *Proceedings Of The 2014 Conference On Innovation & Technology In Computer Science Education*. New York: ACM (American Computer Machinery). pp. 63-69. ISBN 978-1-4503-2833-3
- BUSANICHE, B. (2011). Analfabetización informática o ¿por qué los programas privados fomentan la analfabetización?, en *Construcción Colaborativa del Conocimiento*. México: Universidad Nacional Autónoma de México. Instituto de Investigaciones Económicas. pp. 177-187. ISBN 978-607-02-2373-0
- CABELLO, R. (2006). *Yo con la computadora no tengo nada que ver. Un estudio de las relaciones entre los maestros y las tecnologías informáticas en la enseñanza*. Ed. Prometeo.
- CUTHILL, M. (2002). Exploratory research: citizen participation, local government and sustainable development in Australia. *Sustainable Development*, 10 (2), 79.
- CRESWELL, J. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- COOPER, S.; DANN, W. P. y PAUSCH, R. (2011). *Learning to Program with Alice (w/CD ROM)*. Prentice Hall Press.

- DUSSEL, I. (2011). *Aprender y enseñar en la cultura digital. Documento Básico del VII Foro Latinoamericano de Educación*. Ed. Fundación Santillana
- DUSSEL, I. y QUEVEDO L. (2011). *Educación y nuevas tecnologías: los desafíos pedagógicos ante el mundo digital*. Ed. Fundación Santillana.
- GIL, C. y SANCHEZ, P. (1999). La atención a la diversidad desde la programación del aula. *Revista interuniversitaria de formación del profesorado*, (36), pp. 107-121.
- GUSKEY, T. (2002). Professional development and teacher change. *Teachers and Teaching: theory and practice*, 8(3), pp. 381-391.
- JACINTO, C. (2009). *Universalizar el acceso y completar la educación secundaria. Entre la meta social y la realidad latinoamericana*. Buenos Aires: SITEAL, IIPE-UNESCO.
- JACINTO, C y TERIGI, F. (2007). *¿Qué hacer ante las desigualdades en la educación secundaria? Aportes de la experiencia latinoamericana*. Ed. Santillana.
- KAPLAN, C. V. (1994). *Buenos y malos alumnos: descripciones que predicen*. Ed. Aique.
- LEVIS, D. (2007). Aprender y enseñar hoy: el desafío informático. *Revista Novedades Educativas*, 203.
- LÓPEZ, P.; BONELLI, E. y SAWADY (2014) *El nombre verdadero de la programación*. Universidad Nacional de Quilmes. En línea: <http://elaulayeltrabajo.proyectoslibres.unq.edu.ar/images/3/35/MartinezLopez-Bonelli-Sawady.pdf> [10/02/2016]
- MARTINEZ, M. Y ECHEVESTE, M. (2015). Representaciones de estudiantes de primaria y secundaria sobre las Ciencias de la Computación y su oficio. *Revista de Educación a Distancia*, (46).
- MARTINEZ, C.; GOMEZ, M. y BENOTTI, L. (2015.). A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform, en *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM. pp. 159-164.
- MEIRIEU, P. (2001). *La opción de educar. Ética y pedagogía*. Barcelona: Ed. Octaedro.
- PERRENOUD, P. (2005). *Diez nuevas competencias para enseñar*. Ed. Siglo XXI, 23.
- SADOSKY, F. (2013). *CC-2016. Una propuesta para refundar las enseñanza de la computación de las escuelas Argentinas*. Buenos Aires.
- TORP, L. y SAGE, S. (1998). *El aprendizaje basado en problemas: Desde el jardín de infantes hasta el final de la escuela secundaria*. Buenos Aires: Amorrortu editores.
- TYACK, D. y CUBAN, L. (2001) *En busca de la utopía. Un siglo de reformas de las escuelas públicas*. Ed. Fondo de Cultura Económica.
- WILLIAM, D. (2007) Assessment, learning and technology: prospects at the periphery of Control. Discurso inaugural de la conferencia Association for Learning Technology de Inglaterra. En línea: https://www.alt.ac.uk/sites/default/files/assets_editor_uploads/documents/alte2007_dylan_wiliam_keynote_transcript.pdf [10/02/2016]
- WING, J.M. (2006). Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. *COMMUNICATIONS OF THE ACM*, 3 (49), en línea: <https://www.cs.cmu.edu/~15110-13/Wing06-ct.pdf> [10/02/2016]