

# *Philosophy of Computer Science and its Effect on Education*

## *Towards the Construction of an Interdisciplinary Group*

Sylvia da Rosa

Instituto de Computación  
Facultad de Ingeniería - UDELAR  
Montevideo, Uruguay  
darosa@fing.edu.uy

Federico Gómez

Facultad de Ingeniería - UDE  
Montevideo, Uruguay  
fgomez@ude.edu.uy

Alejandro Chmiel

Departamento de Lógica y Filosofía de la Lógica  
Facultad de Humanidades y Ciencias de la Educación -  
UDELAR  
Montevideo, Uruguay  
alejandro.chmiel@gmail.com

**Abstract**—This article presents an interdisciplinary experience that brings together two areas of computer science; didactics and philosophy. As such, the article introduces a relatively unexplored area of research, not only in Uruguay but in the whole Latin American region. The reflection on the ontological status of computing, its epistemic and educational problems, as well as their relationship with technology, allows us to elaborate a critical analysis of the discipline and a social perception of it as a basic science.

**Keywords**—philosophy, epistemology, interdiscipline

### I. INTRODUCCIÓN

El Instituto de Computación de la Facultad de Ingeniería de la UDELAR (InCo) y el PEDECIBA Informática, realizaron entre el 2009 y el 2011 una serie de acciones conjuntas con el Profesorado de Informática de la ANEP (PI) a través del Programa de Apoyo al Profesorado de Informática

(PAPI)(<http://www.fing.edu.uy/inco/seminarios/papi/proyectos.html>), con el objetivo de apoyar y fomentar el desarrollo académico de los docentes del profesorado, a través de actividades que les permitieran trabajar desde la interna en grupos de investigación. En esas actividades participaron también docentes de Filosofía de la Facultad de Humanidades y Ciencias de la Educación (FHCE), interesados en problemas filosóficos de la propia disciplina informática. A partir del intercambio generado por problemas teóricos acerca de la conceptualización, de la formalización y del pensamiento computacional, aparecen una serie de intereses compartidos entre docentes investigadores de las áreas Didáctica de la Informática (DI)

y Filosofía de la Ciencia de la Computación (FCC). Un ejemplo es el vínculo entre dos preguntas clásicas de la FCC: ¿qué es un programa?, y ¿qué tipo de conocimiento podemos tener de él? Los diferentes paradigmas sobre la ciencia de la computación en los que pueden elaborarse respuestas, impactan directamente sobre la didáctica de la disciplina, es decir sobre el contenido pedagógico de la misma [1,2].

De esta forma, el grupo de trabajo que se establece a partir de ese momento, inaugura un cierto tipo de reflexión acerca de la computación que no se había desarrollado hasta el momento en el país, por lo menos sistemática y explícitamente. Vale aclarar que la filosofía de la ciencia de la computación, como una disciplina, con ciertas preguntas bien definidas, articuladas dentro de un conjunto de trabajos de referencia, no tiene más de treinta años [3,4], aunque también es importante agregar que en los últimos diez años la intensidad de trabajo en el área ha sido muy importante.

Nuestro grupo elaboró una propuesta de trabajo y la presentó para competir por financiación en el llamado a Núcleos Interdisciplinarios de la UDELAR, siendo aprobada en 2012 y dando lugar, a partir de ese momento, al Núcleo Interdisciplinario Filosofía de la Ciencia de la Computación (NI FCC) [5]. El problema general de la propuesta se define como “la ausencia de una cultura en informática como ciencia básica en la sociedad y de una reflexión teórica rigurosa acerca de la tecnología”. Esta *ausencia* es un factor que determina en gran medida la visión parcial y distorsionada que existe sobre la informática y la tecnología en la sociedad, y constituye un freno para el desarrollo educativo, social y productivo. Aportar en la solución de

este problema constituye el fundamento de nuestro proyecto.

Muchos elementos de la cultura informática no requieren una computadora para plantearse; de hecho, históricamente aparecieron antes que estas máquinas y tienen sus raíces en la lógica y la matemática. Su influencia desde un principio trascendió los límites de estas disciplinas, por ejemplo la noción de algoritmo o procedimiento reglado, si bien aparece originalmente en problemas científicos tiene un claro impacto en reflexiones acerca del pensamiento en un sentido general. Es a través de la FCC que estas cuestiones pueden transformarse en objeto de reflexión y conocimiento, proporcionando una fuente muy rica de problemas interesantes desde el punto de vista tanto filosófico como informático. Sin embargo, al haberse transformado la informática en una omnipresencia que abarca casi todos los espacios de la interacción social, cuesta mucho diferenciarla en un sentido científico y como parte de una herramienta tecnológica, la cual se presenta como transparente o aproblemática. La primera consecuencia de esta aproblematicidad es que se llega a la conclusión generalizada de que manejar un dispositivo tecnológico es saber informática, lo cual obviamente es un error (dado que manejar un dispositivo, seguramente a través de un programa, en principio no tiene ningún vínculo con los conceptos propios de la informática), trayendo como segunda consecuencia una sociedad que se autopercebe como experta en aquello en que es analfabeta. Sin mencionarlo como un consuelo, vale agregar que esta problemática es global, es decir, los problemas que tiene Uruguay en estas temáticas pueden ser encontrados en cualquier país latinoamericano, e incluso en países desarrollados. Sin embargo, desde hace aproximadamente una década existe una marcada y creciente tendencia en universidades norteamericanas y europeas a crear y promover institutos que integren investigadores de diversas áreas del conocimiento para aportar en el desarrollo de una visión crítica de la informática, a partir de una reflexión acerca de su estatus ontológico, sus problemas epistémicos, su didáctica y su vínculo con la tecnología, a partir de lo cual elaborar un cierto tipo de crítica que contribuya, entre otras cosas, al desarrollo de una educación en informática como ciencia básica, en línea con lo que está ocurriendo en los países citados. Es de fundamental importancia para dicha visión crítica, desarrollar algunas líneas de trabajo dentro del área FCC, así como también desarrollar la DI como parte de la disciplina informática. En países como Estados Unidos, Francia y Reino Unido, académicos universitarios de primer nivel en el área de la informática han comenzado a estudiar el problema que supone la ausencia de una reflexión acerca de la informática y su relación con la tecnología. Se han detectado obstáculos y propuesto soluciones que están siendo implementadas para corregir la situación de educar a las jóvenes generaciones en informática, desplazando a la enseñanza en TIC dominante hasta el momento [6,7,8,9,10,11]. Esto contribuye a sentar

las bases para que la educación en informática desde los niveles iniciales del sistema educativo sea parte de la disciplina informática y competa a investigadores y académicos de la ciencia de la computación<sup>1</sup>. En este sentido vale la pena citar palabras de Simon Peyton Jones [12] sobre la formación de profesores para la introducción de la ciencia de la computación en la escuela y en la enseñanza secundaria, que ilustran el tratamiento del tema en comunidades informáticas de otros países y que compartimos plenamente:

*But who will teach the teachers? We have two main routes. First, the Computer Science departments of our universities. For the last two decades university CS departments have been utterly disengaged from school ICT, because the subject was of no interest or relevance to them. Now there is a real prospect that CS will be taught to school children, every university academics suddenly have a real stake in what is happening at school.*

En las siguientes secciones se incluye: en la sección 2, las líneas de investigación llevadas adelante, en la sección 3 una breve descripción de los cursos impartidos por nuestro grupo, en la sección 4 las actividades de extensión y en la sección 5 algunas conclusiones.

## II. LÍNEAS DE INVESTIGACIÓN

El grupo de investigación tiene dos líneas generales de trabajo. Por un lado la filosofía de la computación, con un énfasis en los problemas que se detallan en los puntos B y C, y por otro lado, el desarrollo de un modelo para la didáctica de la informática basado en una teoría epistemológica, que se detalla en el punto D.

### A. Filosofía y Computación

La filosofía y la computación tienen un profundo vínculo en la filosofía contemporánea. La interpretación de la relevancia de dicho vínculo puede variar según la corriente filosófica, el enfoque, el área y el problema al que estemos apuntando. Lo que es innegable es que la relación entre filosofía y computación se ha profundizado y desarrollado, por lo menos desde los trabajos de Alan Turing [13], hasta nuestros días. Las discusiones acerca de la inteligencia artificial se conectan profundamente con las discusiones filosóficas acerca de la noción de inteligencia, los problemas de computabilidad se relacionan directamente con discusiones en filosofía de la lógica y de la matemática, los modelos computacionales tienen un enorme impacto en las ciencias cognitivas, en la filosofía de la mente, e incluso en la filosofía de la ciencia; los ejemplos se pueden seguir listando.

<sup>1</sup> Didáctica de la informática es la denominación del mundo franco-germano y computer science education es la denominación del mundo anglo-sajón para el área.

Podríamos establecer una diferencia importante entre dos modos distintos de conexión entre filosofía y ciencia de la computación. Por un lado, (a) cuando los conceptos propios de la ciencia de la computación son tomados como *insumo* para ciertas reflexiones filosóficas, por ejemplo, en filosofía de la mente o filosofía de la matemática. Una discusión amplia de los vínculos entre filosofía y computación, aparece en Colburn [14]. Por otro, (b) cuando los conceptos propios de la disciplina son tomados desde un punto de vista *metateórico*, con la finalidad de elucidar aspectos de los mismos que quedan por fuera de la discusión disciplinar; aunque tales aspectos marquen el estatus ontológico y epistémico de la disciplina. Una lista de problemas filosóficos de la computación en este sentido, aparece en Turner y Eden [15] y Turner [16].

Podríamos agregar un tercer modo, (c) vinculado a los estudios históricos acerca de la ciencia de la computación, para lo cual, necesariamente, requerimos de elementos vinculados a la filosofía de la ciencia.

Investigar problemas de filosofía de la computación, enmarcados en cualquiera de los tres modos mencionados, requiere de grupos de investigación interdisciplinarios. Nuestro grupo se ha concentrado principalmente en algunos problemas que podemos enmarcar en el punto (b) mencionado más arriba, que desarrollaremos brevemente en la próxima sección.

### B. Filosofía de la ciencia de la computación

La filosofía de la ciencia de la computación, entendida en este sentido, toma algunos conceptos propios de la disciplina a partir de los cuales se discuten aspectos ontológicos y epistemológicos de la misma.

Nuestro grupo de investigación ha trabajado sobretodo alrededor de cuatro problemas que marcan profundamente la discusión. Por un lado, porque la respuesta que demos a estas preguntas fundamentales definirá como entendemos la ciencia de la computación. Por otro, porque dichas respuestas están conectadas, en la medida en que conforman una posición teórica consistente.

Dichos problemas son:

1. El *problema ontológico*: ¿qué es un programa?, Moor [17], Colburn [14].
2. El *problema de la implementación*: ¿cuál es el vínculo entre un algoritmo abstracto y una máquina que lo implementa?, Rapaport [18], [19].
3. El *problema de la corrección*: ¿qué significa que un programa es correcto?, Fetzer [20].
4. La *noción de computar*: ¿qué es computar?, Copeland [21], los límites de la computabilidad y la interpretación de la tesis Turing-Church, vínculo entre programas y máquinas que computan dichos programas, Chalmers [22].

Un excelente resumen de dichos problemas se encuentra en Turner [16]. Aquí vamos a mencionar, brevemente, algunos aspectos del *problema ontológico*, su vínculo con el *problema de la implementación y de la corrección*. Este problema afecta globalmente la visión que tenemos de la ciencia de la computación, en particular, permite discutir en qué sentido la computación es, o no, una ciencia formal.

### C. El problema ontológico

El problema ontológico se plantea fundamentalmente con la pregunta ¿qué es un programa? La respuesta suele estar enmarcada en la dualidad abstracto/concreto, en argumentar a favor de la reducción de la noción de programa a uno de los extremos, o, en argumentar la imposibilidad de tal reducción. Podemos argumentar que un programa es una entidad abstracta, una entidad matemática, independiente a su implementación física, o, podemos argumentar que el programa es una entidad concreta, una entidad física, es decir, la máquina física que efectivamente computa aquello que hace el programa.

La distinción puede no tener relevancia ontológica, como plantea Moor [17], en la medida en que, como él argumenta, el límite entre el software y el hardware es contextual, varía de un caso a otro. Por ejemplo, algo que es implementado por el propio hardware en un caso tiene que ser programado a nivel de software en otro.

De todos modos, como otros autores posteriores han señalado, la distinción dualista no puede ser reducida, en particular Colburn [14]. Aparece un argumento, en este sentido, que es el siguiente, si consideramos que sin una descripción lingüística del programa, es decir, solamente con una implementación física, es imposible plantear un criterio normativo de corrección, Kroes [23], tendremos que concluir que no podemos reducir la noción de programa a su contraparte concreta; a menos que aceptemos la posibilidad de quedarnos sin criterios de corrección.

Dicho de otro modo, si observamos la ejecución de una máquina física, o sea, el cómputo efectivo que realiza, y contamos solamente con esta información, no podemos determinar si el programa está funcionando correctamente. Lo cual se presenta como un caso particular de un problema más general, vinculado a la noción de seguir una regla, Wittgenstein [24], Kripke [25].

Por lo cual, la tensión abstracto/concreto a la hora de determinar ontológicamente qué es un programa, parece que no puede ser reducida al extremo de lo concreto, considerando las razones anteriormente señaladas. La otra posibilidad es reducirla a lo abstracto. En tal caso, diríamos que un programa es una entidad abstracta, aunque no esté implementado, e incluso, aunque no pueda ser implementado. En este sentido, diríamos que la ciencia de la computación es una rama de la matemática, lo cual dejaría

muchas áreas de lo que actualmente se considera como parte de la ciencia de la computación, fuera de ella. Algo que, en principio, no hace justicia con la práctica actual de la ciencia de la computación.

Como bien señala Eden [26], la discusión ontológica acerca de los programas, divide las posiciones teóricas en tres paradigmas, que conectan lo ontológico, con una determinada noción epistemológica y metodológica acerca de la computación:

- 1) *Paradigma racionalista*: los programas son entidades *abstractas*, la ciencia de la computación es una rama de la matemática, y la noción de *corrección* de programas se fundamenta en procedimientos *a priori* puramente deductivos.
- 2) *Paradigma tecnocrático*: los programas son entidades *concretas*, la ciencia de la computación es una rama de la ingeniería, y la noción de *corrección* se fundamenta en procedimientos *a posteriori*, generalmente vinculados al testing.
- 3) *Paradigma científico*: los programas son entidades que no pueden ser reducidas a entidades concretas, ni a entidades abstractas, la ciencia de la computación, en este sentido, es parte de las ciencias naturales, y los métodos adecuados para probar corrección, combinan procedimientos *a priori* y *a posteriori*.

#### D. Epistemología genética y didáctica de la informática

El área *didáctica de la informática* (traducción tomada de la expresión del mundo franco-germánico, que es Computer Science Education en el mundo anglosajón) se plantea el estudio de problemas didácticos de la informática, descritos en el contenido pedagógico fundamentalmente por las preguntas *qué, por qué, para quién y cómo* enseñar la disciplina [1,2]. El avance del área como parte de la ciencia de la computación de los últimos años, ha llevado a considerar que la búsqueda de respuestas a estas preguntas corresponde a los académicos y científicos en informática, dedicados al estudio de la educación en la disciplina [9,11,12]. Con el apoyo de los resultados de las investigaciones se pueden generar pautas didácticas para la práctica en clase, de modo que el tratamiento de los problemas didácticos deje de estar reducido a observaciones de experiencias aisladas o de opiniones personales.

Si bien tenemos posición sobre el *por qué*, el *para quién* y el *qué*, nuestra investigación se centra en el *cómo*. Nuestra posición epistemológica sobre la construcción del conocimiento sigue la teoría *Epistemología Genética* de Jean Piaget [29-35], y nos lleva desde el *cómo enseñar* al *cómo aprenden los estudiantes*. Dicha teoría estudia los *mecanismos del desarrollo del conocimiento* desde su *génesis*, y plantea que no existe *discontinuidad* en ese proceso y que los mecanismos involucrados son los mismos

en todas las etapas de dicha construcción [46]. De ahí que provea un modelo aplicable al estudio de la construcción de conocimiento en todos los dominios y en cualquier etapa del desarrollo, que nosotros usamos para investigar acerca de la construcción de conocimiento sobre estructuras de datos y algoritmos básicos por estudiantes novatos.

La teoría explica la construcción del conocimiento (y en particular, del conocimiento *científico*) a partir tanto de los orígenes psicológicos de las nociones y operaciones elementales en las que se basa dicho conocimiento, como de su sociogénesis. Sostiene que el conocimiento no es algo momentáneo ni estático, sino que se construye a través de un proceso *continuo* de interacción con el medio que parte desde el plano de la *acción* sobre lo concreto, y que la reflexión sobre elementos de dicha interacción (acciones del sujeto y modificaciones impuestas al medio), lleva a la transformación de las acciones en operaciones en el plano del pensamiento, lo que constituye un primer paso hacia la *conceptualización*, o sea la construcción de conceptos. Todo el proceso se desarrolla de manera *gradual* y *dialéctica*, sin que existan límites claramente demarcados entre una etapa y la siguiente. En esta transición permanente, pro y retroactiva, hay una reorganización continua del pensamiento y de las ideas relativas al tipo de conocimiento que se está construyendo.

Con respecto a la cuestión de datos de hecho y datos válidos, podemos decir que la epistemología genética se ocupa de estudiar los procesos y mecanismos por los cuales la mente humana transita desde un estado de menor hacia otro estado de mayor conocimiento. Las nociones de “menor” y “mayor” tienen connotaciones formales inherentes a la naturaleza del tipo de conocimiento de cada disciplina (matemática, física, biología, etc.) que es la que determina si un estado de conocimiento es superior o inferior a otro en términos de validez. La epistemología genética se ocupa de explicar los procesos de construcción de dicho conocimiento, en forma independiente de su naturaleza específica [29].

Los datos empíricos que nutren a la epistemología genética provienen de dos fuentes. Por un lado, de los experimentos realizados por Jean Piaget a lo largo de su vida, relativos al desarrollo de los niveles más elementales del conocimiento (desde el nacimiento hasta la adolescencia), los cuales dieron origen a la *psicología genética*, que es la parte más conocida del trabajo de Piaget y que ha tenido influencia en distintas teorías, como la psicología cognitiva, el constructivismo, la teoría de los modelos mentales y la teoría neo piagetana. Por otro lado, del *análisis histórico crítico* llevado a cabo por Jean Piaget y Rolando García en los últimos años de la vida de Piaget, que estudia la evolución de las teorías científicas y provee el material empírico desde los niveles más formalizados del conocimiento científico [35].

Tratando de extraer los procesos inherentes a toda construcción de conocimiento, los autores elaboran un modelo explicativo basado en un proceso general que denominan la tríada intra-inter-trans, presente en tanto en la génesis del pensamiento del sujeto epistémico como en la sociogénesis del pensamiento científico. Este modelo constituye el mayor aporte de la epistemología genética, que permanece abierta.

Nuestro grupo ha llevado adelante múltiples investigaciones, intentando construir una instancia del modelo mencionado para el estudio de la construcción de conceptos de la ciencia de la computación y su *formalización*, su expresión en un lenguaje formal. Uno de los principios de la teoría es que la fuente de conocimiento está dada por la interacción del sujeto con el medio. Al igual que Brousseau en su teoría de las situaciones didácticas [36,37], que considera al medio como problemas matemáticos a los que se enfrenta el estudiante, nosotros partimos de situaciones en las que el estudiante debe resolver instancias de problemas algorítmicos con los que se trabaja en los cursos iniciales de programación, y que el estudiante logra solucionar en la acción (por ejemplo, ordenar o contar objetos, buscar elementos en una lista). A partir de ese conocimiento instrumental del estudiante por el cual sabe hacer, lo ayudamos a elaborar descripciones en lenguaje natural, cada vez más adecuadas sobre el método empleado, como un primer paso hacia la conceptualización, basándonos en la ley general de la cognición [30,33]. La metodología empleada en las investigaciones comprende la realización de entrevistas individuales cuyo objetivo es que el estudiante elabore una solución algorítmica al problema, no sólo para la instancia planteada sino también para el problema general, y en instancias colectivas donde se introduce un formalismo y los estudiantes son guiados en la expresión de sus soluciones algorítmicas en el formalismo. En [38-44] se describen detalladamente investigaciones realizadas acerca de la construcción de los conceptos de inducción y recursión.

### III. ACTIVIDADES DE ENSEÑANZA

Nuestro grupo tiene entre sus objetivos contribuir a la formación de docentes y estudiantes de la ciencia de la computación aportando una visión sobre los problemas filosóficos, especialmente en el campo de la educación. Sin duda, los profesores de computación deben ser expertos en el contenido de sus cursos, pero al mismo tiempo necesitan conocer con cierto grado de profundidad, aspectos significativos de su disciplina que les permite expandir sus perspectivas sobre el campo y por lo tanto mejorar la calidad de enseñanza [45].

Durante 2013 y 2014 hemos elaborado y dictado el curso de *Introducción a los problemas de la filosofía de la computación*, dirigido tanto a estudiantes y profesores de informática como de Filosofía, con participantes de ambos

campos. En el curso se presentan y discuten los problemas descritos en el ítem B de la sección II, con base en una amplia bibliografía.

El curso *Epistemología Genética y aplicaciones a la Didáctica de la Informática* se dicta cada año desde 2010. Está dirigido a los docentes de informática en general y especialmente a aquellos con intención de dedicarse a la investigación en didáctica de la informática. En el curso se introduce el marco teórico, se presenta un modelo para la investigación aplicada a los conceptos de algoritmos y estructuras de datos básicos, incluyendo ejemplos de estudios empíricos con estudiantes. La información sobre los cursos está disponible en <http://www.fing.edu.uy/grupos/nifcc/cursos.html>.

### IV. ACTIVIDADES DIRIGIDAS AL MEDIO (EXTENSIÓN)

Nuestra propuesta tiene entre sus objetivos, uno de carácter social que busca el esclarecimiento de la opinión pública acerca del estatus de la informática como ciencia, tratando de dismantlar la falsa noción que la asimila al manejo o a la capacidad de uso de tecnología (o de algunos aparatos), que se perciben como inevitables, dado el contexto TICs-dependiente en el que nos hallamos inmersos. También en este sentido coincidimos con pensamientos y acciones que se llevan adelante en otros países, como ilustran las palabras de Simon Peyton Jones [12] al respecto:

*But that was the easy part! Now the ground war begins: school by school, head teacher by head teacher, we must make the case, convey the vision, offer support and teaching materials, and train teachers. Explain what computer science is. We need to find ways to explain what our discipline is, in ways that make sense to parents, civil servants, and politicians, not just to the technical community. Clearly distinguishing disciplines from skills and technologies is helpful.*

Es preocupación de nuestro grupo mantener actividades regulares de divulgación dirigidas a docentes, estudiantes y autoridades de la enseñanza, que abarquen temas como:

- Uso y abuso del término informática
- Beneficios del estudio de la informática en la formación general del ciudadano
- Enseñanza de la informática en la educación formal
- Cómo afecta la informática la enseñanza de otras disciplinas
- Distinción entre informática y sus aplicaciones

y se clarifiquen expresiones como:

- informática: una disciplina o ciencia, con las mismas características y problemas generales de

definición que otras disciplinas o ciencias, como matemáticas, lingüística, o física. Los objetos que estudia la informática son los datos y los mecanismos que permiten su gestión,

- didáctica de la informática: al igual que otras disciplinas, (como por ejemplo matemática), la informática tiene un campo que es el de su didáctica, a la que competen los saberes relacionados con la enseñanza-aprendizaje de los conceptos de informática,
- educación en tecnología: es lo que suele llamarse “alfabetización tecnológica”,
- tecnología en educación: se entiende como el saber integrar y articular herramientas y servicios tecnológicos a las prácticas docentes, como apoyo a los procesos de enseñanza-aprendizaje de cualquier disciplina.

El último punto es especialmente relevante porque pone de manifiesto la necesidad del sistema educativo de encauzar y desarrollar la formación pedagógica de maestros y profesores de modo que sean capaces de utilizar la tecnología como elemento potenciador de prácticas educativas, independientemente de la disciplina de que se trate. La pedagogía y la tecnología siempre han estado relacionadas. El auge que han tenido (y tienen) las tecnologías digitales en la sociedad en los últimos años, ha hecho esa relación mucho más explícita debido a la abundancia de productos y servicios tecnológicos en todas las actividades de la sociedad y especialmente en el sistema educativo. Esa abundancia no ha sido acompañada con la articulación e integración de los recursos e instrumentos de la tecnología en la actividad educativa. El riesgo consiste en que por un lado, el uso instrumental e irreflexivo de tecnología desplaza a la pedagogía de su rol conductor de los procesos de enseñanza-aprendizaje y por otro se instala la creencia equivocada de que la didáctica de la informática debe ocuparse de la enseñanza-aprendizaje del uso de productos o servicios informáticos. Disminuir (o evitar) ese riesgo implica una responsabilidad y un compromiso del sistema educativo. Para ello la formación pedagógica de maestros y profesores debería transformarse de modo de incluir el estudio y la investigación sobre las implicaciones del factor tecnológico en la disciplina pedagogía.

La expresión más acabada de dichas actividades estuvo representada por las ediciones del Encuentro de Educación en Ciencia de la Computación (EECC), realizadas en 2012 y 2013, con invitados de la Universidad de Buenos Aires (UBA) y de la

Universidad de Córdoba (Argentina) respectivamente.

En especial en esta última Universidad existe un grupo que trabaja en filosofía de la ciencia de la computación con el cual mantenemos contacto académico y esperamos ampliar la colaboración. Vale la pena mencionar asimismo la mesa de debate organizada por nuestro grupo en el marco del Congreso Iberoamericano de Educación Superior en Computación (CIESC) en Montevideo en 2014, con participación de investigadores de Argentina, Paraguay y Uruguay, sobre la situación en la región de la relación entre la educación superior y la educación media/terciaria en computación (<http://www.fing.edu.uy/grupos/nifcc/extension.html>)

## V. CONCLUSIONES

La reflexión filosófica acerca de la computación realiza un aporte fundamental a la ciencia de la computación, en por lo menos cuatro aspectos:

- 1) Aclara nociones metateóricas que la disciplina misma no se preocupa de aclarar, por la propia naturaleza de tales cuestiones.
- 2) Logra construir un punto de vista crítico acerca de la disciplina, fundamental para el diálogo interdisciplinario, incluyendo las reflexiones didácticas y pedagógicas.
- 3) Dicha crítica también es fundamental para vincular la disciplina con la sociedad, a nivel del sistema educativo, socioeconómico y cultural, sobretodo, ante la situación actual en la que se confunde ciencia de la computación con uso de tecnologías.
- 4) Brinda la oportunidad a docentes en ciencias de la computación, así como a científicos e ingenieros, de contar con un primer acercamiento a la historia y la filosofía de la computación y a su conexión conceptual con otras áreas del conocimiento humano.

Nuestro grupo de investigación constituye un inicio para la construcción de un área de estudio de estos problemas.

## REFERENCIAS

- [1] Saeli, M., Perrenet, J., Jochems, W., Zwaneveld, B. (2014, Jan.). Teaching Programming for Secondary School: a Pedagogical Content Knowledge Perspective. [Online]. Available: [http://www.mii.lt/informatics\\_in\\_education/pdf/INFE177.pdf](http://www.mii.lt/informatics_in_education/pdf/INFE177.pdf)
- [2] Hubwieser, P. *Towards a Conceptualization of Pedagogical Content Knowledge for Computer Science*. ACM 978-1-4503-2243-0/13/08 (ICER 2013), 2013.
- [3] Center for the Study of Language and Information <http://www-csli.stanford.edu/>
- [4] Language, Proof and Logic <http://ggweb.stanford.edu/lpl/>

- [5] (2015, Jan.). The Núcleo-Interdisciplinario-Filosofía-de-la Ciencia-de-la-Computación-(NIFCC)-website. [Online]. Available: <http://www.fing.edu.uy/grupos/nifcc>
- [6] Bradshaw, P., Woollard, J., (2014, Jan.). "Computing at school: An Emergent Community of Practice for a Re-emergent Subject", [Online]. Available: <http://www.icicte.org/Proceedings2012/Papers/15-2-Bradshaw.pdf>
- [7] (2014, Jan.). The Computer-Science-Teachers-Association website. The New Educational Imperative: Improving High School Computer Science-Education, Tech.rep. [Online]. Available: [http://csta.acm.org/Communications/sub/DocsPresentationFiles/White\\_Paper07\\_06.pdf](http://csta.acm.org/Communications/sub/DocsPresentationFiles/White_Paper07_06.pdf)
- [8] (2013, Jan.). The Association-Enseignement-Public-& -Informatique-(EPI)-website.-[Online]-Available: <http://www.epi.asso.fr/>
- [9] (2015, Jan.). L'enseignement de l'informatique en France, il est urgent de ne plus attendre. [Online]. Available: <http://www.epi.asso.fr/revue/articles/a1312b.htm>
- [10] (2015, Jan.). The Computer Science Teachers Association (csta) web site. [Online]. Available: <http://csta.acm.org/Curriculum/sub/K12Standards.html>
- [11] (2015, Jan.). Computing in the national curriculum. [Online]. Available: [http://www.computingatschool.org.uk/data/uploads/CA\\_SPrimaryComputing.pdf](http://www.computingatschool.org.uk/data/uploads/CA_SPrimaryComputing.pdf)
- [12] Peyton Jones, S. (2015, Jan.). Computing at school in the UK. [Online]. Available: <http://research.microsoft.com/en-us/um/people/simonjp/papers/cas/ComputingAtSchoolCACM.pdf>
- [13] Turing, A., "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42, pp. 230–65, 1936.
- [14] Colburn, T., Philosophy and Computer Science. Armonk, N.Y., M.E. Sharp, 2000.
- [15] Turner, Raymond and Amon Eden, "The Philosophy of Computer Science", Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), 2011.
- [16] Turner, Raymond, "The Philosophy of Computer Science", The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), 2014. [Online]. Available: <http://plato.stanford.edu/entries/computer-science/>
- [17] Moor, J. H., "Three Myths of Computer Science", The British Journal for the Philosophy of Science, 29(3), pp.213–222, 1978.
- [18] Rapaport, W. J., "Implementation Is Semantic Interpretation", The Monist, 82(1), pp.109–130, 1999.
- [19] Rapaport, W. J., "Implementation as Semantic Interpretation: Further Thoughts", Journal of Experimental & Theoretical Artificial Intelligence, 17(4), pp.385–417, 2005.
- [20] Fetzer, J. H., "Program Verification: The Very Idea", Communications of the ACM, 31(9), pp.1048–1063, 1988.
- [21] Copeland, B. J., "What is Computation?", *Synthese*, 108(3), pp.335–359, 1996.
- [22] Chalmers, D., "Does a Rock Implement Every Finite-State Automaton?", *Synthese* (108), pp.309–333, 1996.
- [23] Kroes, P., Technical Artefacts: Creations of Mind and Matter: A Philosophy of Engineering Design. Dordrecht, Springer, 2012.
- [24] Wittgenstein, Ludwig: *Investigaciones filosóficas*, México: UNAM, 1988.
- [25] Kripke, S., Wittgenstein on Rules and Private Language. Cambridge, MA: Harvard University Press, 1982.
- [26] Eden, A., "Three Paradigms in Computer Science", *Minds and Machines* 17(2), pp.135–167, 2007.
- [27] How Computers Work: Computational Thinking for Everyone, Rex Page, Ruben Gamboa, (2015, Jan.) EPTCS 106, pp 1-19, DOI: 10.4204/EPTCS.106 [Online]. Available: <http://rvg.web.cse.unsw.edu.au/eptcs/content.cgi?TFPIE2012>
- [28] The Royal Society website. Computing in Schools, Shut down or restart? [Online]. Available: <http://royalsociety.org/education/policy/computing-in-schools/>
- [29] Piaget, J. (2014, Jan.) *Genetic Epistemology*, (a series of lectures delivered by Piaget at Columbia University, translated by Eleanor Duckworth). [Online]. Available: <https://www.marxists.org/reference/subject/philosophy/works/fr/piaget.htm>
- [30] Piaget, J. *La prise de conscience*. France: Presses Universitaires de France, 1964.
- [31] Piaget, J. *L'équilibration des Structures Cognitives, Probleme Central du Développement*. France: Presses Universitaires de France, 1975 .
- [32] Piaget, J. *Recherches sur la Generalisation*. France: Presses Universitaires de France, 1978a.
- [33] Piaget, J. *Success and Understanding*. Harvard: Harvard University Press, . 1978b
- [34] Piaget, J., & coll. *La Formation des Raisonnements Recurrentiels*. France: Presses Universitaires de France, 1963.
- [35] Piaget, J., & García, R. *Psychogenesis and the History of Sciences*. New York: Columbia University Press, 1980.
- [36] Brousseau, G. *Theory of didactical situations in mathematics*. Dordrecht: Kluwer, 1997.
- [37] Sadovsky, P. (2014, Jan). La Teoría de Situaciones Didácticas: un marco para pensar y actuar la enseñanza de la Matemática. [Online]. Available: ([http://www.buenosaires.gob.ar/areas/educacion/cepa/teoria\\_situaciones.pdf](http://www.buenosaires.gob.ar/areas/educacion/cepa/teoria_situaciones.pdf))
- [38] da Rosa, S., Gómez, F. (2010, Jan.) An educational methodology based on the work of students. The Clei Electronical Journal website. Vol. 13, Nr. 2 [Online]. Available: <http://www.clei.cl/cleiej/volume.php>
- [39] da Rosa, S. *Designing Algorithms in High School Mathematics*. Lecture Notes in Computer Science, vol. 3294, Springer-Verlag, 2004.
- [40] da Rosa, S. (2005, Dec.) The learning of recursive algorithms and their functional formalization. [Online]. Available: [www.fing.edu.uy/~darosa](http://www.fing.edu.uy/~darosa)
- [41] da Rosa, S. *The Learning of Recursive Algorithms from a Psychogenetic Perspective*. Proceedings of the 19th Annual Psychology of Programming Interest Group Workshop, 2007.
- [42] da Rosa, S. *The Construction of the Concept of Binary Search Algorithm*. Proceedings of the 22th Annual Psychology of Programming Interest Group Workshop, 2010.
- [43] da Rosa, S., & Chmiel, A. *A Study about Students' Knowledge of Inductive Structures*. Proceedings of the 24th Annual Psychology of Programming Interest Group Workshop, 2012.
- [44] da Rosa, S., *Psychogenesis of the Concept of Recursion* Proceedings of Congreso Iberoamericano de Educación Superior en Computación, 2003.
- [45] Judith Gal-Ezer and David Harel. (2014, Jan.) What (Else) Should CS Educators Know? [Online]. Available: [http://www.openu.ac.il/Personal\\_sites/download/galezzer/what-else.pdf](http://www.openu.ac.il/Personal_sites/download/galezzer/what-else.pdf)
- [46] Flavell, J., *La psicología evolutiva de J. Piaget*, Buenos Aires: Paidós, 1968.