

# Modelado didáctico para ideas fundamentales de computación

Sylvia da Rosa<sup>1</sup> and Manuela Cabezas<sup>2</sup>

<sup>1</sup> Instituto de Computación, Facultad de Ingeniería, Universidad de la República  
darosa@fing.edu.uy

<sup>2</sup> Facultad de Ciencias de la Educación, Universidad de la Empresa  
mcabezas@ude.edu.uy

**Resumen** En este artículo se describen los primeros pasos del proceso de modelado didáctico para un subconjunto de ideas fundamentales de la ciencia de la computación. A lo largo de los años nuestro grupo de investigación ha elaborado un modelo epistemológico que explica y analiza la construcción de conocimiento sobre algoritmos, estructuras de datos y programas, basado en la Epistemología Genética de Jean Piaget. Hemos determinado un subconjunto de ideas fundamentales de la ciencia de la computación que se relacionan con dichos conceptos. A pesar de que el modelo epistemológico continúa en desarrollo, en este momento consideramos necesario centrar las investigaciones en construir un modelo de aplicaciones didácticas de utilidad en el aula. Para ello hemos conformado un equipo interdisciplinario con expertos en ciencias de la educación, en particular en didáctica, y con profesores de enseñanza media, que ha elaborado el proyecto de investigación “El paradigma de las ciencias computacionales y la educación”, aprobado por la Comisión de Enseñanza y la Comisión Sectorial de Investigación Científica de la UDELAR (Uruguay). El objetivo principal del proyecto es operacionalizar y escalar un proceso de construcción de conocimiento para ideas fundamentales de la ciencia de la computación, como primer paso en el proceso de modelado didáctico para las ciencias computacionales. En una segunda etapa, el equipo investigador junto con un grupo de profesores de matemática, física e informática de enseñanza media, podrá a prueba y validará el modelado de herramientas didácticas en el aula, obteniendo información que permita extraer conclusiones.

**Palabras clave:** didáctica, programación, modelado, interdisciplinaridad.

## 1. Introducción

A lo largo de varios años hemos realizado estudios empíricos sobre la construcción de conocimiento algoritmos, estructuras de datos y programas, enmarcados en la Epistemología Genética de Jean Piaget. El resultado de los estudios teórico-prácticos ha dado lugar a un modelo para la investigación de la cons-

trucción de conocimiento sobre conceptos de informática<sup>3</sup>, en particular para programación [32]. A pesar de que el modelo epistemológico es abierto y continúa en elaboración, consideramos que el grado de desarrollo alcanzado nos permite centrar las investigaciones en diseñar un modelo de aplicaciones didácticas, lo cual ha sido uno de los objetivos de las investigaciones desde el inicio, pero que no hemos podido abordar seriamente hasta ahora.

Como parte del proyecto de investigación “El paradigma de las ciencias computacionales y la educación” que estamos llevando a cabo desde marzo 2022 a marzo 2024, hemos conformado un equipo interdisciplinario con expertos en educación, profesores de matemática, informática y física de enseñanza media y docentes e investigadores en ciencia de la computación. El objetivo principal del proyecto consiste en elaborar y validar un proceso de modelado didáctico para algunas ideas fundamentales de la ciencia de la computación, que sirva como base para introducir conceptos computacionales en la educación de las disciplinas científicas. Las ideas fundamentales se discuten en 3.1.

Como primer paso hemos realizado una revisión de la literatura, especialmente de trabajos de investigadores que vienen aportando ideas y resultados de investigaciones desde hace años en temas relacionados con el impacto de la computación en la educación en ciencias. La literatura es extensa y valiosa, en este artículo nos basamos principalmente en [8–11, 33]. En primer lugar, queremos señalar que uno de los aportes más importantes que hemos encontrado en la literatura consiste en que refuerza los fundamentos para afirmar que es imprescindible educar en computación, al menos desde la educación secundaria y abarcando las carreras de grado. Por ejemplo, en [9], Matti Tedre y Peter Denning describen cómo la computación ha producido dos revoluciones. La primera -un cambio radical en las prácticas científicas- es el producto del rol de la informática como herramienta aplicada a las ciencias, debido al insuperable potencial y versatilidad de la computación y la simulación. La segunda revolución, consiste en considerar el computar como una forma completamente nueva de ver los fenómenos naturales y artificiales, cambiando fundamentalmente la forma en que otros campos se ven a sí mismos y hacen su trabajo. Lo que hoy se llama ciencia computacional, implica una nueva era de la ciencia. Cada disciplina, sean ciencias sociales o naturales como física, biología y química o ciencias formales como matemática, comparte el desafío con la ciencia de la computación de construir una educación acorde a la nueva era de la ciencia, considerando, entre otras cosas, que la didáctica de cada disciplina debe integrar desde niveles tempranos los conceptos computacionales que dicha ciencia utiliza como herramientas y métodos fundamentales en la creación y el desarrollo del conocimiento [8–10].

Otro material en el que nos basamos es la recopilación de trabajos de educadores en ciencias de diversos países europeos presentados en la conferencia “Key Competencies in Informatics and ICT (KEYCIT 2014)” [15], que tuvo lugar en la Universidad de Postdam en Alemania en 2014, para discutir investigaciones,

---

<sup>3</sup> Consideramos las expresiones informática y ciencia de la computación como sinónimos, siendo más usada la primera en los países franco-germanos y la segunda en los países anglosajones

estudios de casos, posiciones y perspectivas de la educación en computación y tecnología, enfocadas en la educación secundaria, universitaria de grado y en la formación de profesores. Como puede verse los conceptos de competencias y competencias clave son una cuestión central. Desde finales de la década de los 90 se han brindado definiciones de la noción de competencia y de competencias clave [2, 37] Asimismo, la mayoría de los autores utiliza para definir su modelo didáctico, algún tipo de taxonomía, por ejemplo en [4] los autores toman la definición de competencia de [37] (page 27: “The existence of learnable cognitive abilities and skills which are needed for problem solving as well as the associated motivational, volitional and social capabilities and skills which are essential for successful and responsible problem solving in variable situations.”) y agregan: “This definition implies that competences are learnable by interventions.” Para el modelo de competencias estos autores usan la taxonomía de Anderson and Krathwohl (AKT), una adaptación de la taxonomía de Bloom a la que agregan dos dimensiones : A) niveles de conocimiento (factual, procedural, conceptual, metacognitivo) y B) clasificación de dominios cognitivos (“remembering, understanding, applying, analysing, evaluating, creating”).

Nuestras investigaciones empíricas, desarrollo teórico y revisión de la literatura nos han llevado a concluir que nuestro modelo de construcción de conocimiento basado en la epistemología de Jean Piaget [32] provee un marco epistemológico específicamente adecuado para el modelado didáctico aplicado a las ciencias computacionales. Nuestro modelo no sólo cumple un rol similar al de competencias y taxonomías usado por los autores revisados, sino que además contribuye con elaboraciones teóricas propias, extendiendo la teoría de Piaget para abarcar la construcción de conocimiento sobre programas (ver sección 2). Coincidimos con la mayoría de los autores en que es necesario determinar las ideas fundamentales de la ciencia de la computación que serán el eje para el modelado de herramientas didácticas.

Organizamos este artículo de la siguiente manera: en la sección 2 presentamos brevemente el modelo epistemológico de construcción de conocimientos y en la sección 3 describimos los elementos principales del proceso de modelado didáctico las ideas fundamentales que definimos. En la sección 4 se presentan brevemente los contenidos en relación con el proceso de modelado didáctico. Finalmente se incluyen reflexiones finales y se enumeran las referencias bibliográficas.

## 2. Fundamentos del modelo de construcción de conocimiento sobre programas

La teoría Epistemología Genética de Jean Piaget estudia el proceso de construcción del conocimiento y explica cómo se da la transición de un nivel inferior de conocimiento a otro nivel que se considera mayor [22, 26].

Los datos que dan sustento a la teoría piagetiana provienen principalmente de dos fuentes: por un lado, de los estudios empíricos acerca de la construcción del conocimiento por parte de los sujetos, desde el nacimiento hasta la adolescencia (que dieron origen a la psicología genética de Piaget y su clasificación en estadios

sensoriomotor, preoperacional, de las operaciones concretas, de las operaciones formales.) [19–21, 23–25]. Por otro lado, de un análisis crítico de la historia de las ciencias, realizado por Piaget y García, para investigar el origen y evolución de las teorías científicas [26], donde constatan un paralelismo entre los mecanismos de desarrollo psicogenéticos relacionados con la evolución de la inteligencia en los niños y el desarrollo sociogenético sobre la evolución de las teorías principales en varios dominios de la ciencia, dando lugar a la idea más importante de la teoría piagetiana que es la tríada de etapas, llamadas intra, inter y trans por los autores. La tríada explica desde una nueva perspectiva la construcción de conocimiento tanto a nivel psicogenético como sociogenético: el proceso de construcción del conocimiento consiste en el paso de una primera etapa enfocada en objetos o elementos aislados (etapa intra), a otra que toma en cuenta las relaciones entre los objetos y sus transformaciones (etapa inter), lo que lleva a la construcción de un *système d'ensemble*, es decir, estructuras generales que involucran tanto elementos generalizados como sus transformaciones (etapa trans), integrando las construcciones de las etapas previas como casos particulares.

En la teoría de Piaget, el conocimiento humano se considera esencialmente activo. Es decir, saber significa actuar sobre los objetos y la realidad, y construir un sistema de transformaciones que pueda llevarse a cabo con ellos [22]. El problema general de todo el desarrollo epistemológico radica en determinar el papel de la experiencia y las estructuras operativas del individuo en el desarrollo del conocimiento, y en examinar los instrumentos mediante los cuales el conocimiento ha sido construido en forma previa a su formalización. Este problema fue estudiado en profundidad por Piaget en sus experimentos sobre la construcción de conocimiento desde el nacimiento a la adolescencia, en base a cuyos resultados formuló una ley general de la cognición [19, 20], que regula la relación entre el conocimiento y la conceptualización, generada en la interacción entre el sujeto y los objetos que tiene que manipular para resolver problemas o realizar tareas. Es una relación dialéctica en la que, a veces, la acción guía el pensamiento y, otras veces, el pensamiento guía las acciones. Piaget representa la ley general de la cognición mediante el siguiente diagrama:

$$C \leftarrow P \rightarrow C'$$

donde P representa la periferia, es decir, la reacción más inmediata y externa del sujeto al confrontar los objetos para resolver un problema o realizar una tarea. Esta reacción se asocia a la búsqueda de un objetivo y al logro de resultados, sin que ni las acciones ni las razones del éxito o del fracaso sean conscientes. Las flechas representan el mecanismo interno del proceso de pensamiento, mediante el cual el sujeto se da cuenta de la coordinación de sus acciones (C en el diagrama), las modificaciones que éstas imponen a los objetos, así como sus propiedades intrínsecas (C' en el diagrama). El proceso de la toma de conciencia descrito por la ley general de la cognición constituye un primer paso hacia la construcción de conceptos.

A lo largo de varios años hemos realizado estudios empíricos con el objetivo investigar el proceso de construcción de conocimiento sobre algoritmos básicos

y estructuras de datos por parte de estudiantes novatos enmarcado en el pasaje de la etapa intra a la etapa inter de la tríada piagetiana. El estudio de la construcción de conocimiento sobre programas, considerados desde un punto de vista ontológico como objetos de naturaleza dual con un parte abstracta (el texto) y una parte concreta (la ejecución), [27, 28, 31] sentó las bases para estudiar el pasaje de la etapa inter a la etapa trans. Un aspecto fundamental a la hora de construir conocimiento sobre el programa es que el estudiante sea capaz de conceptualizar que existe una relación de causa y efecto entre las acciones que ha expresado en el texto del programa y las acciones realizadas por la computadora, así como entre las variaciones sufridas por los objetos formales referidos por dicho texto y las variaciones sufridas por los objetos que existen en memoria durante la ejecución. Esta relación de causa y efecto investigada en [27, 30, 31] llevó a una reformulación de la ley general de la cognición, la cual se ilustra a continuación:

$$\underbrace{C \leftarrow P \rightarrow C'}_{newC \leftarrow newP \rightarrow newC'}$$

donde  $newP$  representa la nueva periferia donde el estudiante ha construido conocimiento sobre la parte textual del programa, es decir, el algoritmo y las estructuras de datos.  $newC$  representa las acciones del algoritmo ejecutadas por la computadora, mientras que  $newC'$  representa las modificaciones que la implementación de las estructuras de datos sufren en memoria durante la ejecución del programa.

El conocimiento sobre la ejecución del programa se construye en la interacción entre el estudiante (centrado en  $newP$ ), y sus observaciones sobre las acciones que realiza el programa cuando se ejecuta ( $newC$ ) y las modificaciones que dichas acciones imponen a las estructuras de datos durante la ejecución ( $newC'$ ). La recorrida señalada por las flechas en el segundo renglón del diagrama simboliza dicha interacción. La relación causal modelada por ambos renglones explica la construcción de conocimiento sobre algoritmos, estructuras de datos y programas.

### 3. Modelo epistemológico y modelado didáctico

El largo trayecto en investigaciones epistemológicas brevemente descrito arriba, no implica que las posibles aplicaciones didácticas sean evidentes. En primer lugar, buscamos desarrollar un modelado didáctico en una disciplina científica que no siempre ha sido reconocida como tal. La informática es una ciencia joven en relación a otras ciencias, y la educación informática particularmente joven en relación a otras áreas didácticas [36].

Hoy en día, la literatura académica en educación informática presenta una gran diversidad en direcciones y propuestas de investigación [17]. Si bien la reciente productividad científica nos hace pensar que estamos hablando de una disciplina con su propio campo de conocimiento y que evoluciona hacia una agenda de investigación, todavía quedan cuestiones pendientes [5, 7].

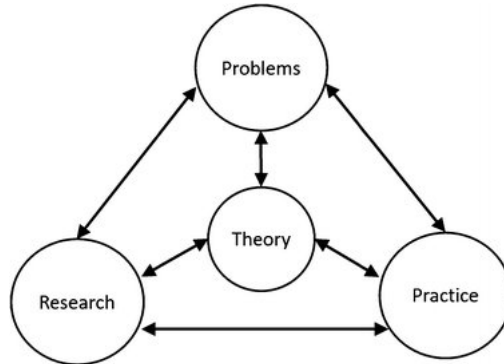


Figura 1: Triángulo académico centrado en la teoría

En su revisión sistemática de investigaciones en educación informática los autores citados encontraron que la mitad de los estudios no explicita marco teórico alguno. Para los estudios que sí lo hacen, sus teorías y modelos conceptuales son tomados de otras áreas como la psicología y presentan un área dispersa con gran variedad de terminología y métodos. Si bien los autores anticipan un crecimiento del campo teórico de la educación informática, al momento de su estudio consideraron el número de estudios con marcos teóricos y conceptuales propios del área tan pequeños que no tendrían suficiente impacto para generar una unificación teórica del área [17].

Al igual que la educación en otras ciencias, con excepción de las matemáticas, el problema al que apuntan los autores es sobre la relación entre teoría y práctica, o investigación y educación, que en parte surge del sub-valorado rol de las investigaciones didácticas dentro de la academia [1].

Nuestra propuesta de investigación para el modelado didáctico para las ideas fundamentales de la computación, comienza con la noción de la didáctica como disciplina científica propia [12] donde la misma va más allá de ser la base del conocimiento docente [34]. Es un campo de investigación entre la teoría y la práctica, que se ocupa de describir y comprender lo que pasa en las aulas de informática en cuanto al proceso de enseñanza y aprendizaje de un *contenido específico* para llegar a interpretarlo en términos de teoría didáctica

En efecto, entendemos el rol de la teoría como central en la construcción de la didáctica como ciencia, representado por el triángulo académico<sup>4</sup> propuesto por [35], que se muestra en la figura 1. El autor plantea la discusión del rol de la teoría en relación a la práctica, la investigación y los problemas educativos.

¿Cuál es, entonces, el proceso de modelado didáctico para la educación informática?

El modelado didáctico es proceso cíclico de diseño y estudio de secuencias didácticas en un dominio específico (contenido/disciplina). Es el producto del esfuerzo colaborativo de investigadores y docentes que buscan lograr mejoras en

<sup>4</sup> El autor usa la expresión “the scholarship triangle”

la enseñanza de un contenido y un curriculum específico. Surge desde el análisis de las prácticas y sus fundamentos, el diseño de secuencias didácticas y la investigación en el aula para poner a prueba lo teorizado, para luego reiniciar el ciclo desde una nueva teorización. Es así que parte siempre desde un posicionamiento en la disciplina y en la justificación de sus fundamentos.

Metodológicamente, el enfoque de análisis didáctico que proponemos se ubica dentro del área conocida como investigación de diseño educativo, que proviene de las ciencias del diseño (ingeniería), las áreas STEM y, particularmente, de la didáctica de la matemática [6, 18].

Es de particular importancia para dicho análisis, y especialmente en un área tan dispersa como la educación informática, que el modelado didáctico pueda guiarse por la formulación de ideas fundamentales para construir contenidos sobre ideas explícitas. Es así que hablamos de modelado para un contenido *específico*, tanto al hablar de didáctica como de contenido cuando nos referimos a investigaciones de modelado didáctico. Hacemos la distinción, ya que en educación es usual encontrar teorías del aprendizaje que se trasladan al proceso de diseño curricular en términos de principios u orientaciones pedagógicas generales, pero no se ocupan de los problemas específicos de qué sucede cuando intentamos introducir un contenido específico y los problemas que la situación presenta [1]. Ciertamente, en investigaciones educativas la palabra teoría a menudo se remite primordialmente a un meta-aprendizaje descontextualizado sin conexiones claras con ningún contenido. Esto lo vemos, por ejemplo, en la construcción del “pensamiento computacional” que ha tenido gran aceptación en el ámbito educativo, y se plantea como una forma de pensar y resolver problemas en términos generales, independiente de los contenidos y conceptos de la computación, y (fundamentalmente) como competencia fácilmente trasladable de un dominio a otro [16]. A diferencia de este enfoque, en las investigaciones didácticas la tarea es contruir y extender los conocimientos teórico-prácticos desde la experiencia de docentes expertos y, fundamentalmente, en relación a contenidos disciplinares específicos.

En cuanto al procedimiento, el modelado es esencialmente un proceso de desarrollo de secuencias didácticas, centrado en teoría, como vemos en la figura 2 [18].

Para nuestro caso, entonces, el punto inicial para la formulación de las ideas fundamentales es el desarrollo teórico propio para la construcción de conocimiento sobre conceptos informáticos, dado por nuestro modelo epistemológico descrito en la sección 2.

### 3.1. Ideas fundamentales

Describimos el papel relevante que juega la teoría propia en la determinación de las ideas fundamentales que funcionan como ejes para el modelado de herramientas didácticas. Las ideas fundamentales agrupan los conceptos centrales y de largo alcance de la informática, permitiendo distinguir el conocimiento de las destrezas.

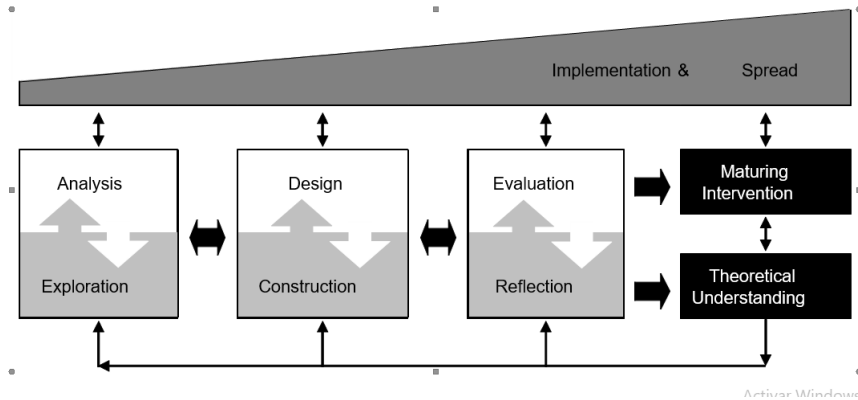


Figura 2: Proceso de diseño centrado en la teoría tomada del modelo “Generic model for conducting educational design research” en McKenney & Reeves, 2019

El trabajo de Andreas Schwill de 1992 sobre las ideas fundamentales en ciencia de la computación [33] es un clásico citado por varios autores como punto de partida para elaboración de modelados didácticos. Schwill define cuatro criterios que debe cumplir una idea fundamental:

- el criterio vertical (la idea aparece en distintos dominios de la disciplina)
- el criterio horizontal (la idea se puede trabajar en cualquier nivel intelectual)
- el criterio de tiempo (la idea se puede observar a lo largo del tiempo de evolución de la disciplina)
- el criterio de sentido común (la idea tiene sentido en un contexto informal, pre-teórico y pre-científico y puede ser expresada en lenguaje natural)

y señala como ideas fundamentales (con sub-ideas) la algoritmización, la disección estructural y el lenguaje. En [11] Gilles Dowek agrega como fundamental la idea de máquina relacionada con la noción de programa como objeto ejecutable. En [3], los autores proponen diez ideas fundamentales, enumeradas a continuación, que abarcan las de Schwill y Dowek y agregan otras relativas al desarrollo ulterior de la ciencia de la computación (redes, seguridad, simulaciones).

1. La información se representa en forma digital.
2. Los algoritmos interactúan con los datos para resolver problemas computacionales.
3. El rendimiento de los algoritmos se puede modelar y evaluar.
4. Algunos problemas computacionales no pueden ser resueltos por algoritmos.
5. Los programas expresan algoritmos y datos en una forma que se puede implementar en una computadora.
6. Los sistemas digitales están diseñados por humanos para satisfacer las necesidades humanas.
7. Los sistemas digitales crean representaciones virtuales de fenómenos naturales y artificiales.



8. La protección de los datos y los recursos del sistema es fundamental en los sistemas digitales.
9. Las operaciones dependientes del tiempo en sistemas digitales deben coordinarse.
10. Los sistemas digitales se comunican entre sí mediante protocolos.

Le especificidad del proceso de modelado implica que nos hemos enfocado en la construcción de conocimiento sobre algoritmos, estructuras de datos y programas. En consecuencia, tomamos las ideas fundamentales relacionadas con dichos conceptos (1, 2, 3, 4, 5), separándolas de las relacionadas con cuestiones como seguridad, redes y ética (el resto).

La formulación de los autores de las ideas 2, 4 y 5 nos llevó a indagar acerca de las nociones de problema no computable, problemas sin solución computable y problema no computacional ya que es relevante su claridad para este subconjunto de ideas fundamentales. Es así que encontramos que en el documento complementario a su artículo<sup>5</sup>, los autores profundizan en cada una de las diez ideas fundamentales propuestas, expresando sobre la idea 2: “The term ‘computational problem’, ‘algorithmic problem’, or simply ‘problem’ in this context is often used to refer to the task that needs to be computed e.g. searching for a word, sorting values into order, finding the shortest route on a map, or finding a face in a photo.” En este punto es donde se observa claramente la importancia de nuestra elaboración teórica, en la que se distingue el “mundo algorítmico” del “mundo computacional” y que nos lleva a decir que “problema algorítmico” y “problema computacional” no son lo mismo (¡y menos simplemente “problema”!), aún en un contexto informático. Como señalamos en nuestros trabajos previos, por ejemplo en [27], la construcción de conocimiento acerca de una solución computacional a un problema algorítmico es regulada por la extensión de la ley de Piaget. La ley original de Piaget regula la construcción de conocimiento sobre la solución algorítmica y nuestra extensión establece la necesidad de comprender la relación dialéctica entre una y otra, representada por los dos renglones de la ley extendida (ver sección 2).

Cabe señalar que Dowek en [11] considera a “la máquina” como uno de los conceptos fundamentales de la informática, cuyo vínculo esencial con el desarrollo histórico, filosófico y científico de nuestra disciplina es innegable.

En consecuencia, las ideas fundamentales para las cuales nos planteamos diseñar secuencias didácticas y validarlas en el aula son las siguientes:

1. La información se representa en forma digital.
2. Los algoritmos interactúan con los datos para resolver problemas algorítmicos.
3. Los programas implementan algoritmos y datos en una forma que se puede ejecutar en una computadora.
4. El rendimiento de los algoritmos y programas se puede modelar y evaluar.

<sup>5</sup> <https://www.canterbury.ac.nz/media/documents/oexp-engineering/BigIdeas-webdocument.pdf>

Para cada una de ellas definiremos sub-ideas fundamentales, relacionadas por ejemplo con la distinción entre información y datos, la definición de problema algorítmico, paradigmas de programación y lenguajes, cuestiones de eficiencia y complejidad.

#### 4. El contenido, el modelado y sus productos

Al día de hoy, estamos cerrando la etapa exploratoria de análisis (figura 2) y comenzando a formular los principios para la etapa de diseño y construcción. Aquí resaltamos algunos aspectos importantes: partiendo de las ideas fundamentales 2 y 3 enumeradas arriba, en las que se distingue la noción de algoritmo de la de programa de acuerdo a la extensión de la ley de la cognición (sección 2), ¿cuál es el contenido específico que usaremos para la introducción de las cuatro ideas fundamentales sobre algoritmos y programas que hemos determinado? Si bien no existe una definición formal de algoritmo, sí la hay para *problema algorítmico*, dada por D. Harel en [14] (pág. 16), según la cual un problema algorítmico consiste en:

1. una caracterización de una colección legal, posiblemente infinita, de potenciales conjuntos de entrada,
2. una especificación de las salidas deseadas en función de las entradas.

Partiendo de esta definición el autor define solución algorítmica, enfocándose en que la misma será ejecutada por un computador e implementada en un lenguaje imperativo (de hecho más adelante en su libro, págs. 19 y 20 habla de las estructuras de control propias de un lenguaje imperativo).

Si bien el autor no distingue solución algorítmica de programa, consideramos válido tomar como punto de partida su definición de problema algorítmico y derivar de ella algunas sub-ideas para las ideas fundamentales, para las que pueden elaborarse actividades de aula en torno al concepto de *función*, como por ejemplo:

1. Formulación de un problema como problema algorítmico: información, datos, entrada, salida
2. Diseño de una solución algorítmica: función de una entrada en una salida
3. Definición de una solución: ¿cómo se expresa la función?
4. Del algoritmo al programa: ¿quién ejecuta la función? ¿cómo?

Con respecto a la idea fundamental 4 las actividades serían en torno a:

- ¿cómo saber que la solución algorítmica es correcta?
- ¿cómo obtener un programa más eficiente en términos de recursos de la máquina?

Como hemos señalado, el autor se enfoca en una implementación en un lenguaje imperativo de la solución algorítmica. Sin embargo, esto no tiene por qué ser así, y dado que el instrumento que usamos para la introducción de las ideas

fundamentales es el concepto de función, consideramos que el paradigma funcional es más adecuado pues dicho concepto es asimismo central en los lenguajes funcionales. Contamos para ello con el lenguaje funcional MateFun, desarrollado en nuestro instituto, que ya hemos usado con relativo éxito en actividades con estudiantes y docentes [29].

El proceso de diseño de secuencias didácticas exige una planificación minuciosa de cada paso de la secuencia, con foco en lo que entendemos como esencial en la computación representado por las ideas fundamentales. Utilizamos técnicas de diseño en las que el docente investigador debe planificar cada detalle desde la justificación teórica e intentar describir el resultado esperado para cada paso de la secuencia. Si bien el diseño paso a paso organiza y prepara a los docente para el proceso, su función principal es producir una serie de instrumentos, a través de las múltiples decisiones que debe tomar el investigador para elaborar todos los componentes de la secuencia. Éstos pasan a funcionar como instrumentos de medición y validación empírica de los supuestos teóricos, y a su vez, son la base empírica de los productos teóricos del modelado.

En este sentido, el objetivo final del proceso de modelado siempre es la teoría. Más específicamente, como plantean los autores en [6] en el proceso de diseño la teoría es, por un lado un recurso fundamental y, por el otro, también su producto. Esto no quita valor a las relaciones, lecciones y materiales didácticos que nos dejan las secuencias como recursos sobre los cuales docentes y maestros podrán volver a poner a prueba sus ideas e hipótesis sobre la enseñanza y el aprendizaje de sus estudiantes. Pero en el modelado didáctico, cada guía, secuencia y tarea diseñada para aportar a la investigación, queda limitada al momento y contexto de la secuencia particular que describe. Recae sobre los docentes investigadores saber qué datos recaudar, por qué, y cómo analizarlos a la hora de aprovechar cada secuencia dentro su contexto específico, y paulatinamente construir los modelos a nivel de un contenido, una serie de contenidos o una disciplina científica. Es tarea del docente investigador asimismo saber detectar y mapear problemas y dificultades para descubrir nuevas interrogantes escondidas en los “errores” de los estudiantes. Esta etapa sucede cuando los estudiantes recorren una secuencia y se “trancan”. Las nuevas interrogantes son producto directo de la recaudación y análisis de estas rupturas en las secuencias, y uno de los momentos más ricos del proceso de modelado didáctico.

## 5. Reflexiones finales

En estos primeros meses de desarrollo del proyecto hemos constatado que para satisfacer los objetivos académicos que nos hemos planteado es imprescindible que logremos llevar a cabo una *investigación interdisciplinaria*, entendida como lo plantea Rolando García en [13]: una investigación interdisciplinaria se diferencia de investigaciones muti (o trans) disciplinarias en “el modo de concebir una problemática y en el común denominador que comparten los miembros de un equipo de investigación. Mientras que en el caso de las investigaciones multidisciplinares se suman los aportes que cada investigador realiza desde su

disciplina en torno a una problemática general, en el caso de la interdisciplina la integración de los diferentes enfoques está en la delimitación de la problemática ... cuyo estudio requiere de la coordinación de enfoques disciplinarios que deben ser integrados en un enfoque común.” En este sentido, entendemos que la problemática no está definida en el punto de partida de la investigación sino que su definición surge en el transcurso de la propia investigación y para este caso en particular. En este trabajo presentamos los primeros pasos de la construcción de nuestro modelado didáctico, concebida como una investigación interdisciplinaria.

## Referencias

1. Ametller, J., Leach, J., Scott, P.: Using perspectives on subject learning to inform the design of subject teaching: an example from science education. *The curriculum journal*, 18(4) pp. 479–492 (2007)
2. Athey, R., Orth, M.: *Emerging competency methods for the future*. John Wiley and Sons, Inc. (1999)
3. Bell, T., Tymann, P., Yehudai, A.: The big ideas in computer science for k-12 curricula. *Bulletin of EATCS*, 1(124). (2018)
4. Bröker, K. Kastens, U., Magenheimer, J.: Competences of undergraduate computer science students. In *KEYCIT – Competencias clave en informática y TIC*. Torsten Brinda, Nicholas Reynolds, Ralf Romeike, Andreas Schwill (Eds.) (2014)
5. Caspersen, M., Gal-Ezer, J., McGettrick, A., Nardelli, E.: *Informatics for all. the strategy*. ACM Europe & Informatics Europe (2018)
6. Cobb, P., Gravemeijer, K.: Experimenting to support and understand learning processes. In A. E. Kelly, R. A. Lesh, & J. Y. Baek (Eds.), *Handbook of design research methods in education*. London: Routledge. (2008)
7. Denning, P.: Computational thinking in science. *Sigma Xi, The Scientific Research Society, American Scientist* 105 (2017)
8. Denning, P., Tedre, M.: Shifting Identities in Computing: From a Useful Tool to a New Method and Theory of Science. In Hannes Werthner and Frank van Harmelen, Eds. *Informatics in the Future, Proceedings of the 11th European Computer Science Summit* (2015)
9. Denning, P., Tedre, M.: *Computational Thinking*. Cambridge, MA : The MIT Press (2019)
10. Denning, P., Tedre, M.: Computational thinking: A disciplinary perspective. *Informatics in Education*. 20(3), 361–390 (2021)
11. Dowek, G.: Les quatre concepts de l’informatique. *Bulletin de l’Association EPI* (2012)
12. Fensham, P.J.: *Defining an identity: The evolution of science education as a field of research*. Springer Science & Business Media. Vol. 20. (2004)
13. García, R.: *Sistemas Complejos. Conceptos, método y fundamentación epistemológica de la investigación interdisciplinaria*. Editorial Gedisa S.A. (2006)
14. Harel, D., Feldman, Y.: *Algorithmics - The Spirit of Computing*. Addison-Wesley Publishers Limited 1987, 1992, Pearson Education Limited 2004 (2004)
15. Key competencies in informatics and ict. <https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/7032/file/cid07.pdf> (2014)
16. Lodi, M., Martini, S.: Computational thinking, between papert and wing. *Science & Education* volume, 30 pp. 883—908 (2021)

17. Malmi, L., Sheard, J., Bednarik, B., Helminen, J., Kinnunen, P., Korhonen, A., Myller, N., Sorva, J., Taherkhani, A., Simon: Theoretical underpinnings of computing education research: what is the evidence? ICER14: Proceedings of the tenth annual conference on International computing education research pp. 27–34 (2014)
18. McKenney, S., Reeves, T.: *Conducting Educational Design Research*, 2nd edition. Routledge, ISBN 9781138095564 (2019)
19. Piaget, J.: *La Prise de Conscience*. Presses Universitaires de France (1964)
20. Piaget, J.: *Success and Understanding*. Harvard University Press (1974)
21. Piaget, J.: *L'équilibration des Structures Cognitives, Problème Central du Développement*. Presses Universitaires de France (1975)
22. Piaget, J.: *Genetic Epistemology, a series of lectures delivered by Piaget at Columbia University, translated by Eleanor Duckworth*. Columbia University Press (1977)
23. Piaget, J.: *Recherches sur la Généralisation*. Presses Universitaires de France (1978)
24. Piaget, J., Beth, E.: *Mathematical Epistemology and Psychology*. D.Reidel Publishing Company, Dordrecht-Holland (1966)
25. Piaget, J., coll.: *La Formation des Raisonnements Recurrentiels*. Presses Universitaires de France (1963)
26. Piaget, J., Garcia, R.: *Psychogenesis and the History of Sciences*. Columbia University Press, New York (1980)
27. da Rosa, S.: Piaget and computational thinking. CSERC '18: Proceedings of the 7th Computer Science Education Research Conference pp. 44–50 (2018)
28. da Rosa, S., Chmiel, A., Gómez, F.: Philosophy of computer science and its effect on education: Towards the construction of an interdisciplinary group. *Clei Electronic Journal (CLEIej)*, ISSN 0717-5000 19(1) (2016)
29. da Rosa, S., Viera, M., García-Garland, J.: *Mathematics and matefun, a natural way to introduce programming into school*. (2020)
30. da Rosa, S.: Studying preconceptions of novice learners about program execution. *Proceedings of Psychology of Programming Interest Group (PPIG)* (2016)
31. da Rosa, S., Aguirre, A.: Students teach a computer how to play a game. *Lecture Notes in Computer Science*, vol. 11169, Springer-Verlag pp. 55–67 (2018)
32. da Rosa, S., Gómez, F.: Towards a research model in programming didactics. *Proceedings of 2019 XLV Latin American Computing Conference (CLEI)* p. 1–8 (2019)
33. Schwill, A.: Computer science education based on fundamental ideas. *Proceedings of the IFIP TC3 WG3.1/3.5 joint working conference on Information technology: supporting change through teacher education* pp. 285–291 (1997)
34. Shulman, L.: Knowledge and teaching: Foundations of the new reform. *Harvard educational review*, 57(1) pp. 1–23 (1987)
35. Silver, E., Herbst, P.: *Theory in mathematics education scholarship*. Second handbook of research on mathematics teaching and learning. VOL1. pp. 39–7 (2007)
36. Tedre, M.: From a black art to a school subject: Computing education's search for status. *ITiCSE '20: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* pp. 3–4 (2020)
37. Weinert, F.E.: Concept of competence: A conceptual clarification. In D. S. Rychen, and L. H. Salganik (Eds.), *Defining and selecting key competencies*. Seattle, WA: Hogrefe and Huber Publishers. pp. 45–65 (2001)