

## USO DE MAPAS CONCEPTUALES PARA DESCUBRIR IDEAS PODEROSAS Y GUIAR EL DISEÑO DE UN CURRÍCULO DE PENSAMIENTO COMPUTACIONAL

### (USE OF CONCEPT MAPPING TO DISCOVER BIG IDEAS AND GUIDE THE DESIGN OF A COMPUTATIONAL THINKING CURRICULUM)

Andrés Rodríguez<sup>1</sup>, Natalia Zamora<sup>1</sup>, Carol Angulo<sup>1</sup> & Alberto J. Cañas<sup>2</sup>  
<sup>1</sup>Fundación Omar Dengo, Costa Rica

<sup>2</sup>Institute for Human & Machine Cognition (IHMC), USA

Email: {Andres.Rodriguez, Natalia.Zamora, Carol.Angulo}@fod.ac.cr, acanas@ihmc.us

**Abstract.** The “Programa Nacional de Informática Educativa (PRONIE MEP-FOD)” is the national computers-in-education program for the public-school system of Costa Rica, and is carried out since 1988 through an alliance between the Ministry of Education and the Omar Dengo Foundation. Students throughout K-9 are involved in computer programming with the purpose of developing students’ problem solving and thinking skills, based on a constructionist approach to learning. A redesign of the curriculum was initiated several years ago with the intention of going more in-depth into programming skills and problem solving. Additionally, we intended to introduce into the curriculum the more fundamental concepts of computing and have instructors make them evident throughout the programming projects, providing students a better understanding of current and future technologies. To determine what those fundamental concepts of computing were, and the big ideas that unify them, we used concept mapping, where a large team collaborative constructed a large concept map that helped discover the big ideas and guide the curriculum design.

**Keywords:** concept mapping, curriculum design, powerful ideas, computational thinking

## 1 Introducción

Durante el rediseño del currículo de K-9 en informática educativa del Programa Nacional de Informática Educativa (PRONIE MEP-FOD) (Fundación Omar Dengo, 2006; Zamora, 2012) que se lleva a cabo en alianza desde 1988 el Ministerio de Educación Pública y la Fundación Omar Dengo en Costa Rica, se busca preparar a los estudiantes para los retos que enfrentarán en las próximas décadas, tanto por confrontar nuevas tecnologías como por resolver problemas nuevos y complejos. El rediseño contempla una mayor profundización en temas que forman parte del pensamiento computacional ‘tradicional’, como la programación, así como introducir la computación física (e.g. robótica, sensores) en todas las escuelas y colegios del programa, y fortalecer la capacidad de resolución de problemas. Al mismo tiempo, inspirados por Wing (2006) se busca darle a docentes y estudiantes una mejor comprensión de los conceptos básicos de computación (mejor referenciada por el término en inglés, *computing*) que deben dominar para entender tecnologías que aún no han sido inventadas, y comprender a fondo la habilidad principal que desarrollan: la programación.

El análisis de experiencias y proyectos desarrollados en otros países que pudieran servir como referencia al rediseño permite identificar que la implementación de la programación y el pensamiento computacional en la mayoría de los países y proyectos está todavía a nivel de proyecto ‘después de clases’, de clubes, de lecciones optativas, o de inmersiones cortas, y frecuentemente están empezando en su implementación; en contraste con treinta años de experiencia del PRONIE MEP – FOD en Costa Rica. Más aún, no se encuentra un estudio que describa los conceptos básicos de computación o las ideas poderosas que deben ser los hilos conductores de los fundamentos de computación que se aprenden a través de los diez años de estudio de la Educación General Básica (K-9). Por ejemplo, las siete ideas poderosas de la computación presentadas por el College Board y la National Science Foundation de Estados Unidos (ver: Grover & Pea, 2013) están orientadas para un curso avanzado en ciencias de la computación a nivel de secundaria (AP Computer Science Principles).

Bajo este panorama, una tarea necesaria, antes de continuar con el diseño fue descubrir cuáles son los conceptos que deben ser comprendidos a través de estos diez años, y las ideas poderosas que sirven de hilos conductores para integrar todos esos conceptos. Durante el proceso de descubrimiento, el grupo a cargo de este rediseño, construye un mapa conceptual ‘gigante’ que incluye todos los conceptos que forman parte del currículo anterior del PRONIE, más los conceptos adicionales que consideran deben o pueden ser incluidos al profundizar en los fundamentos de computación.

En este artículo, se presenta el proceso de construcción de este mapa conceptual, y cómo el mapa permitió identificar cuatro ideas poderosas y los conceptos asociados a estas ideas, sobre las cuales está diseñado el nuevo currículum del PRONIE: LIE++ pensar, crear, programar.

## 2 Antecedentes

En 1988 la Fundación Omar Dengo (FOD) establece una alianza con el Ministerio de Educación Pública de Costa Rica (MEP), quienes conforman el Programa Nacional de Informática Educativa (PRONIE MEP-FOD), con el objetivo de desarrollar las capacidades de las personas, enfocándose en estudiantes del sistema educativo público, desde preescolar hasta el noveno grado por medio de propuestas educativas innovadoras, apoyadas en el aprovechamiento de nuevas tecnologías (Fundación Omar Dengo, 2013). El Programa implementa un modelo pedagógico centrado en el aprendizaje de y con la programación, orientado a desarrollar capacidades cognitivas y sociales de alto nivel, como el razonamiento lógico-matemático, la creatividad, la resolución de problemas y la colaboración, basándose desde el inicio en el construccionismo propuesto por Papert (Harel & Papert, 1991; Papert, 1993). El PRONIE cubre un 87,9% de los estudiantes K-9 del sistema de educación pública, y de estos 72,6% tienen dos lecciones por semana para trabajar en los proyectos de programación propuestos por el Programa, bajo la guía de un Tutor preparado para esa labor.

A partir del 2014, avances en tecnología unidos al auge en el ámbito internacional de la importancia de enseñar a los estudiantes a programar (*coding*), a las tendencias mundiales de implementación del pensamiento computacional y la cultura Maker (Papavlasopoulou, Giannakos, & Jaccheri, 2017) en las escuelas, y la mayor disponibilidad y menor costo de equipamiento para robótica y computación física, establecen la necesidad de actualizar la propuesta pedagógica del PRONIE. Se plantea desarrollar una propuesta que lleve al estudiante a desarrollar el pensamiento, prácticas y habilidades del pensador computacional, orientado por competencias, que permiten la inclusión de nuevas tecnologías, metodologías, destrezas y habilidades requeridas para resolver situaciones que puede que aún no existan y que emergerán en un futuro muy cercano, por la rápida evolución de las mismas tecnologías y su inclusión en la sociedad. Se propone la importancia de considerar la tecnología más allá del laboratorio de informática educativa, incorporando los teléfonos inteligentes, las redes sociales, los avances en la comunicación, el acceso a la información inmediata y en tiempo real, el control de dispositivos a distancia y programables a través del Internet de las cosas (*Internet of Things -- IoT*) y avances que están por venir.

Se inicia con una construcción curricular y didáctica, orientada hacia el logro de competencias por parte de los estudiantes, a partir de un análisis desde lo procedimental, actitudinal y conceptual del pensamiento computacional. Para el 2015, el equipo diseñador propone un currículo para Informática Educativa (IE) en el que se especifica por nivel educativo una serie de indicadores que describen para cada dimensión de competencia lo que se espera ver hacer al estudiante. En este primer análisis, el equipo diseñador también hace evidente los conceptos del cómputo que consideran se deben abordar. En la Tabla 1 se muestra este currículo para el primer y segundo grado.

Este currículo consiste de una serie de actividades o proyectos de resolución de problemas y diseño que requieren programación a lo largo de los 10 años (K-9), considerando la edad de los estudiantes. Adicionalmente, en diversos grados se hace uso de programación física (e.g., sensores, actuadores, etc.) y robótica. A través de los años el nivel de complejidad de los problemas y la programación requerida aumenta, profundizando en ambos con respecto al currículo anterior. Para cada grado o proyecto se lista los conceptos de programación y computación que deben ser cubiertos.

Sin embargo, pronto se identifica que, a pesar de que había una coherencia a través de los años en cuanto a los niveles de programación, complejidad de la programación esperada, y complejidad de los problemas a resolver, no había una coherencia en el abordaje de los conceptos ni en el tiempo (de un grado a otro) ni dentro de los proyectos en sí. Más aún, los conceptos abordados se derivaban de la tecnología y programación usada en el grado o proyecto particular sin que existiera una idea más amplia de los conceptos e ideas poderosas que se querían abordar a través de los años. Sí se había logrado una uniformidad de terminología con la intención de que los estudiantes empezaran desde pequeños a usar los términos correctos y formales, por ejemplo ‘variable’ en lugar de ‘cajita’, y en los pilotos se evidencia que este objetivo se estaba logrando con los estudiantes.

Al arrancar el PRONIE hace 30 años, la computación como ciencia (Ciencias de la Computación) no había llegado a la madurez en que se encuentra hoy día. Siguiendo la premisa de que en la mayoría de las disciplinas hay un número

pequeño de conceptos e ideas poderosas (*big ideas*) que nos permiten comprender la disciplina (National Research Council, 2012; Hazen & Trefil, 2009), es como se empieza a explorar la posibilidad de que en computación hay un número de ideas poderosas y conceptos que permiten comprender no solo la programación sino también la tecnología de la computación (TI), y que su comprensión nos facilita entender nuevas tecnologías que se van desarrollando. Dado que los conceptos fundamentales de la computación se han mantenido a través de muchos años, no son conceptos que habría que estar actualizando frecuentemente, como sí sucede con conceptos relacionados con la tecnología. Así, además de los conceptos que tradicionalmente se estudian como CPU, memoria, entrada/salida, variable, red, etc., se comienza a trabajar con conceptos más fundamentales, como máquina (i.e., máquina de estado finito, máquina de Turing), estado, procesos, y abstracción computacional. Esto lleva al equipo diseñador a formularse preguntas como: ¿cuáles son esos conceptos fundamentales de la computación (*computing*)? ¿podemos diseñar un currículum de K-9 basado en estos conceptos? ¿Y cuáles son las ideas poderosas que enmarcan esos conceptos? No se encuentra en la literatura ningún ejemplo de un currículo que cubra varios años (en el caso de Costa Rica, 10 años), que empiece desde preescolar, y que cubra los conceptos de la computación al nivel esperado. En la mayoría de los casos estos conceptos se abordan en cursos específicos universitarios o avanzados de secundaria, sin proveer la coherencia o hilo conductor que interesa para este nuevo currículo. Para descubrir cuáles son esos conceptos, sus relaciones, y esas ideas poderosas, y entenderlos, es que se recurre a los mapas conceptuales.

4	<p>Selecciona y opera de manera segura una gama de dispositivos de hardware para llevar a cabo tareas específicas y utilizar procedimientos básicos de solución de problemas.</p> <ul style="list-style-type: none"> <li>Contexto</li> </ul> <p>Conoce de la existencia y uso básico de los principales sistemas operativos y algunos de sus programas asociados.</p> <ul style="list-style-type: none"> <li>Sistema Operativo</li> <li>Peopleware</li> <li>Aplicación</li> </ul> <p>Describe los principales componentes y funciones de los sistemas y redes de computación.</p> <p>Navega por la web y lleva a cabo búsquedas avanzadas para recopilar contenidos digitales. (US)</p> <ul style="list-style-type: none"> <li>Seguridad</li> <li>Metadato</li> </ul> <p>Desarrolla criterios para la compra o actualización del hardware y del software de un sistema de computación. (US)</p> <ul style="list-style-type: none"> <li>Eficiencia</li> </ul> <p>Explica los múltiples niveles de hardware y software que apoyan la ejecución de un programa (por ejemplo, los compiladores, intérpretes, sistemas operativos, redes).</p> <ul style="list-style-type: none"> <li>Compilador</li> <li>Intérprete</li> </ul> <p>Identifica métodos de transmisión de datos e información en redes.</p> <ul style="list-style-type: none"> <li>Modelo</li> </ul>	<p>Desarrolla experiencias de resolución para la creación de algoritmos en pseudocódigo:</p> <ul style="list-style-type: none"> <li>Descomposición</li> <li>Refinamiento</li> <li>Depurar</li> <li>Diagrama</li> <li>Pseudocódigo</li> <li>Ejecución</li> <li>Flujos</li> <li>Abstracción</li> </ul> <p>Construye soluciones demostrativas y documentales, con modelos de representación del entorno natural o social mediante: Simulaciones.</p> <ul style="list-style-type: none"> <li>Simulación</li> <li>Prototipo</li> </ul> <p>Explora formas aplicables más estructuradas (7 pasos) de solución a problemas de la vida real.</p> <ul style="list-style-type: none"> <li>Empatía</li> <li>Problema</li> <li>Enunciado</li> <li>Solución</li> <li>Cuestionar</li> </ul> <p>Se interesa por aventurarse en la creación de diversas soluciones y el autoaprendizaje. (PC)</p> <p>Participa en redes de colaboración en línea y comparte con otros soluciones implementadas a problemas diversos. (PC)</p> <ul style="list-style-type: none"> <li>Conectar</li> <li>Expresar</li> </ul> <p>Usa la computación para hacer preguntas que le den sentido a cosas computacionales en el mundo (PC)</p> <ul style="list-style-type: none"> <li>Cuestionar</li> </ul>	<p>Elabora en la computadora soluciones planificadas con anticipación a diferentes entornos y lenguajes de programación utilizando de manera correcta conceptos como: Secuencias, ciclos controlados, eventos accionados por diferentes entradas y métodos, condicionales anidadas, operadores y datos (PC).</p> <ul style="list-style-type: none"> <li>Parámetros</li> <li>Alto Nivel</li> <li>Entidad</li> <li>Objeto</li> <li>Herencia</li> <li>Iteración</li> <li>Agrupación</li> <li>Periodo</li> <li>Finitud</li> <li>Precisión</li> </ul> <p>Modulariza soluciones empleando métodos, procedimientos y funciones para construir programas completos uniendo colecciones de partes más pequeñas. (PC)</p> <ul style="list-style-type: none"> <li>Reutilizar</li> <li>Objeto</li> <li>Herencia</li> <li>Procedimiento</li> <li>Función</li> </ul> <p>Desarrolla soluciones programadas en colaboración con otros.</p> <ul style="list-style-type: none"> <li>Conectar</li> <li>Expresar</li> </ul> <p>Utiliza su capacidad de abstracción para anticipar los resultados.</p> <ul style="list-style-type: none"> <li>Abstracción</li> </ul> <p>Explica la programación desarrollada en relación con el problema planteado y otras alternativas de programación posibles.(CO)</p> <ul style="list-style-type: none"> <li>Diagrama</li> <li>Modelo</li> </ul>	<p>Conoce la relación entre la representación de datos y calidad de los datos.</p> <ul style="list-style-type: none"> <li>Modelo</li> <li>Información</li> </ul> <p>Sabe lo que es una base de datos relacional, y entiende los beneficios del almacenamiento de datos en varias tablas.</p> <ul style="list-style-type: none"> <li>Ámbito</li> <li>Base de Datos</li> <li>Relación</li> </ul> <p>Consulta datos de las en una tabla utilizando un lenguaje típico de consultas.</p> <ul style="list-style-type: none"> <li>¿?</li> </ul> <p>Realiza búsquedas complejas de información por ejemplo, utilizando operadores booleanos y relacionales.</p> <ul style="list-style-type: none"> <li>Lógica</li> </ul> <p>Entiende cómo y por qué los valores pueden ser datos escritos en diferentes lenguajes cuando son manipulados dentro de los programas y entre dispositivos.</p> <ul style="list-style-type: none"> <li>Alto Nivel</li> <li>Información</li> </ul> <p>Utiliza la programación para gestionar datos masivos y usarlos de manera intencionada dentro de un programa.</p> <ul style="list-style-type: none"> <li>Big Data</li> </ul>
PC: pensamiento computacional; CR: Creatividad; US: Uso seguro y responsable de las tecnologías; CD: Colaboración				

**Tabla 1.** Dimensiones de la malla curricular para los laboratorios de Informática Educativa (versión inicial, 4 nivel, no implementada).

### 3 Diseño Curricular con Mapas Conceptuales

Los mapas conceptuales facilitan expresar de forma gráfica la relación entre conceptos a través de proposiciones (Novak & Cañas, 2008). Más aún, por su naturaleza, los mapas conceptuales obligan a examinar con detenimiento esas relaciones entre conceptos, determinar cuáles son las más relevantes en el contexto del mapa, y escoger las frases de enlace que mejor expresan esas relaciones. Durante la construcción colaborativa de mapas conceptuales, la construcción del mapa, y en especial la construcción de las proposiciones, llevan a una negociación de significado entre los participantes en el proceso. Cuando esta construcción social se da entre un grupo de estudiantes se da un mayor aprendizaje; cuando esta construcción se da en un grupo de trabajo lleva a consenso y comprensión colectiva del tema.

Es por esto que el equipo diseñador considera que los mapas conceptuales pueden ser la herramienta que les permita de manera colectiva, descubrir los conceptos y sus relaciones, por lo que se dan a la tarea de revisar la literatura sobre uso de mapas conceptuales en el diseño de currículo para confirmar que era una decisión correcta.

Poco después de su introducción como una herramienta para aprender, los mapas conceptuales fueron identificados como una herramienta que podría utilizarse en el diseño curricular (Novak & Gowin, 1984; Stewart, 1979). Ya en 1990 Starr & Krajcik (1990) argumentan que los mapas conceptuales pueden ayudar a los docentes de ciencias a desarrollar currículo de ciencias que sea jerárquico, integrado y conceptualmente motivado, y reportan en detalle los mapas construidos por docentes de sexto grado. Edmondson (1995) fue de los primeros en reportar en la literatura el resultado de usar mapas conceptuales para desarrollar un currículo completo, basado en problemas, para veterinaria incluyendo cursos específicos y ejercicios basados en casos. Encontró que utilizar mapas conceptuales en el proceso de planeamiento resultó en cursos más accesibles y fácilmente integrados por parte de los estudiantes, y que los mapas conceptuales fueron particularmente útiles para crear cursos y ejercicios interdisciplinarios. McDaniel (2005) se enfoca en el uso de mapas conceptuales para el diseño y gestión curricular, comunicando claramente los enlaces entre competencias y el contenido curricular, y reporta el uso de la herramienta para diseñar el currículum y mantener programas académicos que satisfacen competencias estipuladas que llevan a certificación y acreditación. Sherborne (2008) nos muestra como los mapas conceptuales pueden ser utilizados en las diferentes etapas del desarrollo curricular: diseño, comunicación e implementación del currículo. En general, al ser concisos y visuales por naturaleza, los mapas conceptuales le ayudan a recordar a los educadores las áreas principales de énfasis (Edmondson, 1995), a ser más motivados conceptualmente (Starr & Krajcik, 1990), a una mayor integración del contenido, y a establecer o recordar relaciones claves entre conceptos que posiblemente se hayan perdido en una malla curricular lineal (Cañas & Badilla, 2005). Los mapas conceptuales también proveen a los docentes un entendimiento más comprensivo de lo que los estudiantes deben aprender y ayudan a eliminar errores de secuencia en el planeamiento para sus lecciones (Martin, 1994).

El uso reportado en la literatura de los mapas conceptuales para diseño curricular no era exactamente el que se tenía pensado. Desde la perspectiva del equipo diseñador, la secuencia de ‘contenidos’ del currículo en cuestión es una serie de proyectos y actividades de programación sugeridas, y no una secuencia de conceptos. Siguiendo un enfoque constructorista, el docente hace evidentes los conceptos en el momento apropiado durante los trabajos de programación del estudiante. Por lo tanto, no hay una secuencia estricta de conceptos, ni momentos previstos en los cuales se deben abordar. De ahí que la intención no es que el mapa conceptual (o los mapas conceptuales, según sea el caso) provea(n) una secuencia de conceptos o temas como es el caso de los esfuerzos reportados en la literatura, sino más bien que guíen en determinar cuáles son los conceptos que deben cubrirse y las ideas poderosas que los integra, como veremos más adelante en este artículo.

La integración de los mapas conceptuales con la tecnología también ha permitido que sean utilizados como organizadores de contenido para cursos (Chang, Sung, & Chiou, 2002; Chiou, Huang, & Hsieh, 2004; Coffey & Cañas, 2003; Ghatasheh, Najdawi, Abu-Faraj, & Faris, 2013; Walsh, 2017). Más recientemente, se ha propuesto el uso de mapas conceptuales como Itinerarios (Cañas & Novak, 2010) que permiten a los estudiantes decidir entre diferentes caminos para abordar los contenidos, ofreciendo un currículo flexible, y estos han sido utilizados con éxito en educación primaria y secundaria (e.g., Agudelo Velásquez, Salinas Ibáñez, & Zea Restrepo, 2012). De nuevo, en un enfoque constructorista donde los estudiantes desarrollan proyectos de programación, y es durante esos proyectos que los conceptos se hacen evidentes por parte de los docentes en el momento oportuno y sin secuencia preestablecida, los itinerarios no se prestan para presentar el contenido del currículo. A pesar de que en la literatura no se encontró un uso de los mapas conceptuales en diseño curricular como el que se pretende realizar, la experiencia del equipo diseñador con mapas conceptuales para descubrir ideas poderosas en otros contextos (Wu, 2017) respalda la intención de usarlos para encontrar los conceptos fundamentales y las ideas poderosas de la computación y el pensamiento computacional.

## **4 Construcción Colectiva del Mapa Conceptual**

### *4.1 La Lista de Conceptos*

Treinta años generan una gran experiencia y conocimiento en cuanto a (1) la posible progresión en los niveles y complejidad de programación y de resolución de problemas para estudiantes de diferentes edades y experiencia, y (2)

los conceptos relacionados con la programación que los estudiantes van comprendiendo a través de los diez años en el programa. La secuencia de proyectos y programas del nuevo currículo se basa en (1). La lista de conceptos resultado de (2) fue insumo para el siguiente paso.

A esta lista de conceptos se le agrega un conjunto nuevo de conceptos que se identifican relacionados con la profundización en programación y la introducción de computación física y robótica. La computación física y la robótica se han trabajado en la Fundación Omar Dengo por muchos años – nunca a nivel de todas las escuelas y colegios del PRONIE – por lo cual había en el equipo de trabajo un dominio de los conceptos involucrados.

La intención de trabajar a un nivel más fundamental los conceptos de computación implicaba crear una tercera lista de conceptos. Sin embargo, este esfuerzo trajo a relucir que la mayoría de los miembros del equipo de trabajo, muchos con títulos universitarios en informática educativa, desconocían esos conceptos. Verdaderamente se evidenció que se estaban introduciendo conceptos que tradicionalmente no se manejan a este nivel preuniversitario, y a nivel universitario solamente en cursos de Ciencias de la Computación. ¿Cómo lograr que los estudiantes comprendieran conceptos que los instructores de sus docentes no conocían? Se comienza por nivelar el conocimiento del grupo mediante talleres en los cuales se introducen estos conceptos. Al mismo tiempo, un grupo pequeño construye en forma conjunta el mapa conceptual sobre Pensamiento Computacional de la Figura 1 como un inicio de comprensión del tema a nivel de grupo.

Con tres listas de conceptos, cada una, de una fuente de diferente, y cada concepto con su ‘historia’ (e.g., “conceptos que se abordaban en quinto grado en el currículo anterior”; “conceptos asociados a los nuevos proyectos de segundo grado”; “concepto que se estudia en robótica”, etc.), y cada lista con su categorización y tablas, lo primero que se hizo fue que cada concepto se ‘olvidara’ de dónde venía. El equipo se imaginó metiendo todos los conceptos en una bolsa, como se muestra en la Figura 2, y batiendo la bolsa para que cada concepto “se olvidara de dónde venía.”

Con una única lista de conceptos, se da inicio a la construcción colectiva del mapa conceptual, se parte del mapa de la Figura 1, y se agregan uno por uno varios cientos de conceptos. Los conceptos incluyen el pensamiento computacional (a como se entiende en la literatura -- definición más restringida que la que presentamos en la Figura 1), computación física, las prácticas y actitudes que se desarrollan a través de la programación, conceptos ‘tradicionales’ relacionados con computadoras, así como conceptos propios de la programación, y conceptos fundamentales de la ‘computación’ (*computing*).

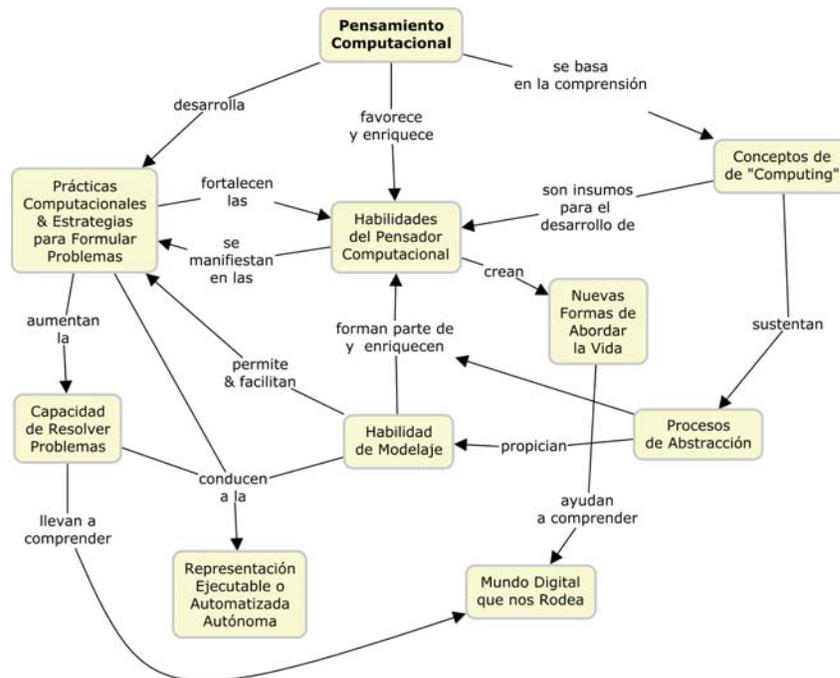


Figura 1. Mapa conceptual sobre Pensamiento Computacional.



**Figura 2.** Que cada concepto se olvide de dónde viene.

#### 4.2 La Construcción del Mapa Conceptual

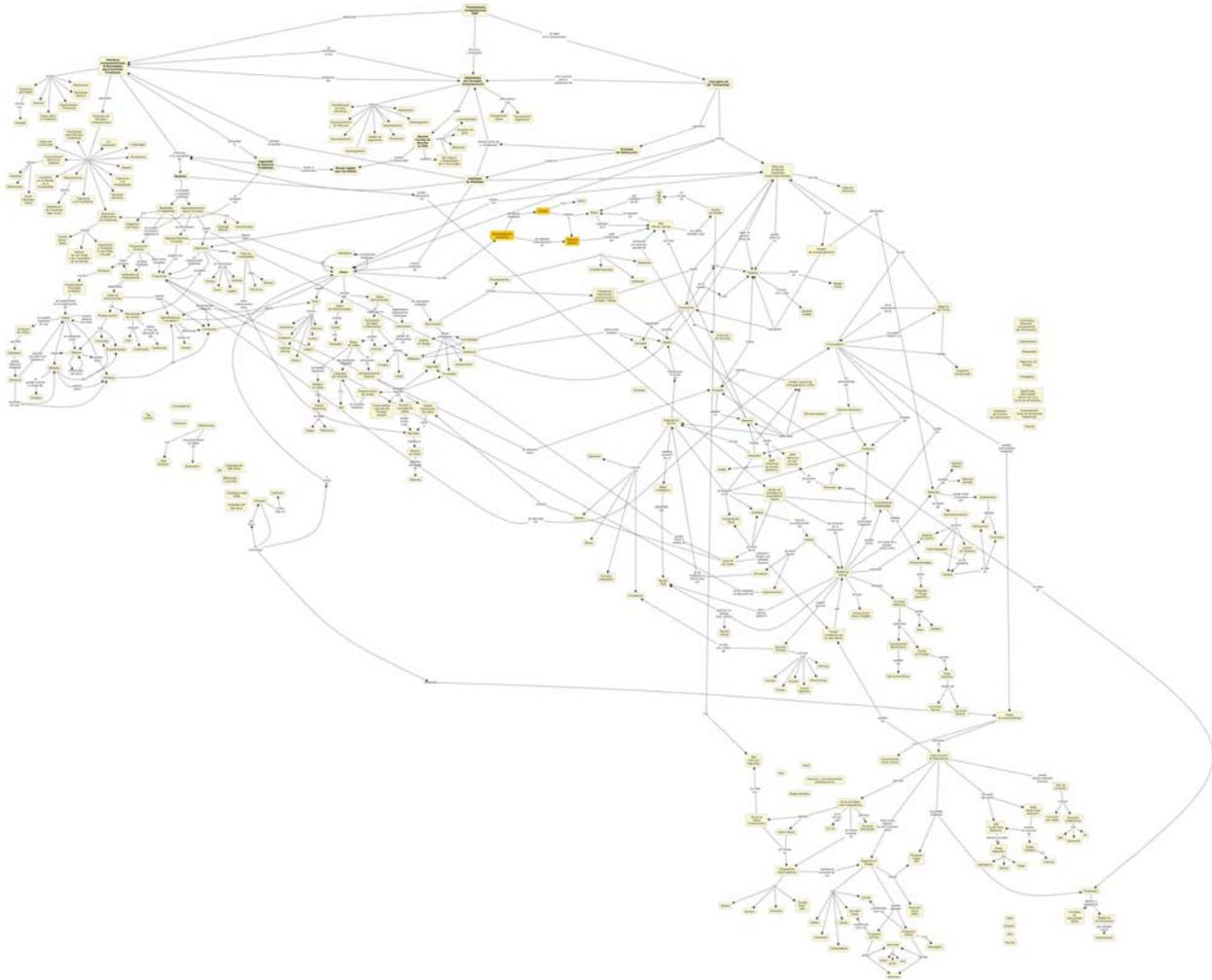
La construcción colectiva del mapa tomó 7 sesiones de 8 horas con un equipo que llegó a tener en algunas sesiones más de 20 personas al mismo tiempo (se proyectaba el mapa en una pantalla grande en un auditorio). El estado actual del mapa se muestra en la Figura 3. El trabajo de refinar el mapa ha continuado, con un grupo más pequeño. Los conceptos se iban introduciendo de uno en uno, y cada concepto se analizaba generando un proceso de discusión académico a lo interno del grupo, que permitió a todos los miembros clarificar o comprender el concepto, así como ir identificándolo dentro de la propuesta curricular. El proceso de construcción del mapa sirvió como complemento de los talleres que se habían impartido, permitiendo que los participantes terminaran de apropiarse de conceptos que hasta hace poco eran desconocidos. Como era de esperarse, la introducción de casi todos los conceptos implicaba un reacomodo del mapa. Un resultado de la construcción por parte del grupo, es que al final el mapa es ‘de todos’, y todos los miembros del equipo entienden el mapa, vivieron su construcción, y entienden cada uno de sus conceptos.

La Figura 3 muestra el estado actual del mapa al escribir este artículo. Algunas partes, como las secciones arriba a la izquierda, ya han sido refinadas. El objetivo del mapa no era generar la secuencia de conceptos que se debían cubrir en los 10 años de currículo. Ni siquiera se espera que todos los 300 conceptos en el mapa sean cubiertos en los 10 años.

#### 4.3 El Análisis del Mapa Conceptual

De lo más importante de un mapa conceptual es *lo que nos dice el mapa* (Cañas, Novak, & Reiska, 2015). Durante la construcción del mapa se hace evidente como ciertos nuevos conceptos, o nuevas secciones del mapa, constantemente se enlazaban a algunos pocos conceptos. Uno de estos conceptos, *Datos*, resulta ser el concepto “central” del mapa, en el sentido de que es el concepto con más enlaces de entrada y salida. Esto permite a través del mapa recordar y reforzar al equipo la idea de que las computadoras no son más que máquinas que procesan datos. El término que hace unas décadas era de uso común, *procesamiento de datos*, es en realidad una forma acertada de describir el trabajo que realizan las computadoras. Si se ve el mapa como un conjunto de grupos de conceptos (“clusters”, en inglés), el concepto Datos tiene una gran cantidad de enlaces a otros grupos, cualquiera que sea. Esto hace evidente que los Datos tienen relevancia al abordarse casi todos los otros conceptos. Los Datos, y las Operaciones sobre Datos, son una *Idea Poderosa (Big Idea)* dentro de este nuevo currículo de Pensamiento Computacional. Era importante tomar esta Idea Poderosa y seguirla a través de los proyectos y programas que los estudiantes elaboran durante los 10 años, de manera que puedan tener una comprensión global del concepto Datos. Por ejemplo, en preescolar se puede empezar a conversar sobre meta-datos (sin usar el término, por supuesto) al describir los colores o las imágenes que aparecen en una foto, o de qué se trata un vídeo. Al empezar la programación en los primeros grados, las constantes son datos, y luego las variables representan valores (datos), y las entradas y salidas son datos. Al avanzar en la programación se empieza a trabajar con estructura de datos, matrices, y el trabajo con sensores implica entrada y salida de datos físicos. Al manejar archivos se le está dando persistencia a los datos. Y ya en secundaria se empieza a manipular cantidades mayores de datos en Bases de Datos. El Internet de las Cosas se basa en la comunicación de datos por parte de computadoras que se encierran en todo lado, lo que lleva a entender Big Data. Todos estos conceptos se van haciendo evidentes y se discuten según vayan siendo necesitados en el desarrollo de los proyectos y programas, según la necesidad y el criterio del docente. Igual sucede con las operaciones sobre Datos y la representación de Datos. En vez de

especificar, por ejemplo, que trabajar con operaciones sobre hileras de caracteres (*strings*) se lleve a cabo en 5 grado, el equipo ha dividido los 10 años en 4 niveles (Nivel 1: Preescolar, 1 y 2 grado; Nivel 2: 3 y 4 grado; Nivel 3: 5 y 6 grado; nivel 4: 7-9 grado), se recomienda el nivel en que se cree que los estudiantes pueden empezar a trabajar con hileras. De acuerdo con los proyectos y programas que desarrollen, que pueden ser los recomendados o pueden ser los que el docente defina, este tiene un par de años durante los cuales sus estudiantes deben trabajar y llegar a dominar el concepto.



**Figura 3.** Mapa conceptual con conceptos del Pensamiento Computacional (mapa en elaboración, intencionalmente ilegible para que se aprecie su tamaño).

Por supuesto que hay secuencias en el orden en que deben abordarse los conceptos. Por ejemplo, el concepto de protocolo se puede analizar como una máquina de estados finita que controla el intercambio de datos entre dos procesos, donde se integran los conceptos protocolo, máquina de estados finita, procesos y datos. El comprender los conceptos de máquina de estados, estado, datos y procesos permite una mejor comprensión del concepto protocolo. El análisis del mapa permite definir esas secuencias, o “ríos” como les llama el equipo diseñador, y determinar para los diferentes niveles esos conjuntos de conceptos. La Figura 4 muestra cómo se usó el mapa impreso para definir uno de esos ríos.

Continuando con *lo que dice* el mapa conceptual de la Figura 3, los grupos de conceptos (*clusters*) que se fueron formando igualmente iban sugiriendo o confirmando conceptos claves, y en otras ocasiones permiten reflexionar acerca de que conceptos como Red de Computadoras, se comprende mejor como intercambio de datos entre procesos corriendo en diferentes máquinas, basándose en la integración de los conceptos Datos, Procesos y Máquinas. Lo que parecía como una Idea Poderosa, Redes de Computadoras, deja de serlo. El grupo de datos (*cluster*) de arriba a la

izquierda de la Figura 3 corresponde a Programación y Actitudes del Pensador Computacional. Como era de esperarse, Programación se mantuvo como una de las Ideas Poderosas, y el mapa conceptual lo confirmó. La construcción colectiva del mapa llevó a que fuera por consenso la determinación de los conceptos más relevantes. Es así como se acuerda trabajar con cuatro Ideas Poderosas: *Datos y Operaciones sobre Datos*, *Máquinas y Programas*, *Abstracción y Modelos* y *Programación*. De estas ideas poderosas, y basándose en el mapa conceptual, se derivan los conceptos que los docentes deben ir haciendo evidentes durante el desarrollo de los proyectos y programas por parte de los estudiantes. Las ideas poderosas y los ríos de conceptos dan coherencia a los proyectos de programación, y promueve en los estudiantes una mayor comprensión de la tecnología de la computación, preparándolos para enfrentar las tecnologías del futuro.

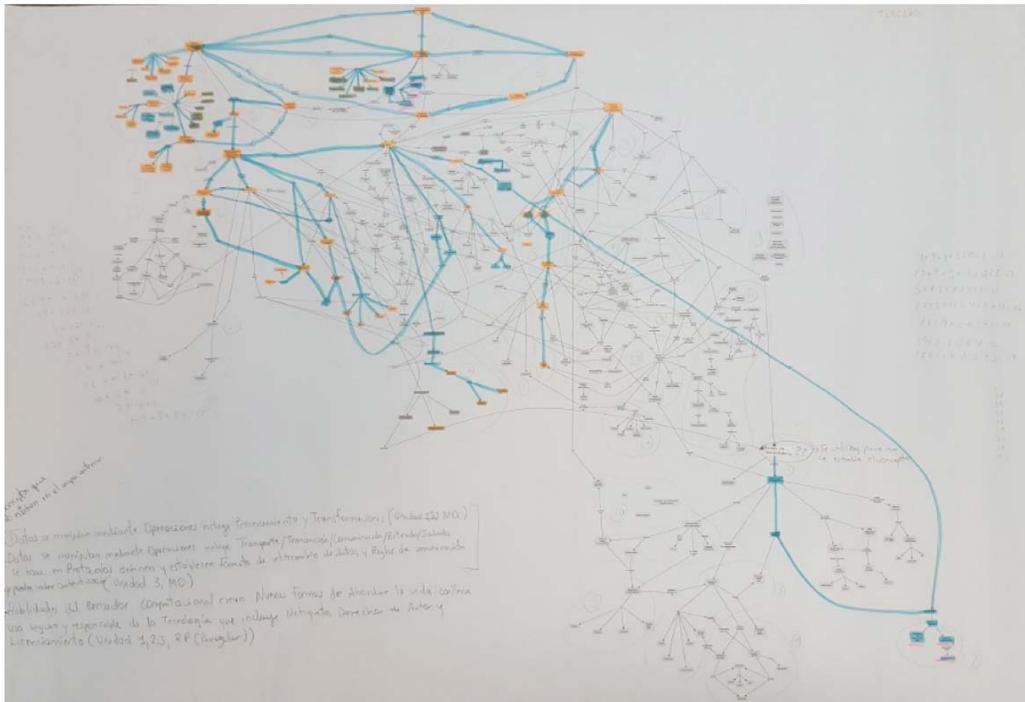


Figura 4. Ríos de conceptos a partir del Mapa Conceptual.

## 5 Conclusiones

La construcción colectiva del mapa conceptual sirvió no solamente para descubrir las ideas poderosas del pensamiento computacional que guían el nuevo currículo de informática educativa LIE++ y extraer ríos de conceptos relacionados para los diferentes niveles académicos, sino que permitió nivelar conocimiento sobre pensamiento computacional y computación entre los miembros del equipo, y ayudó a que todos los miembros del grupo se apropiaran del mapa y entendieran de dónde venían las ideas y los conceptos.

El mapa sirvió de guía para entender la evolución de los conceptos asociados a cada idea poderosa a través de los 10 años de currículo, y para la asignación de los conceptos a los diferentes niveles educativo. La discusión a nivel de ideas poderosas y la designación de conceptos a niveles llevó a que se comprendiera la importancia de que los conceptos no fueran asignados a puntos específicos en el tiempo, sino que se deben hacer evidentes por los docentes en los momentos en que son relevantes durante los proyectos de programación. Como resultado, se logra un nuevo currículo con mayor profundidad en la programación, la resolución de problemas, y el pensamiento computacional, siguiendo el hilo conductor de las ideas poderosas, siempre dentro de un enfoque de aprendizaje constructorista.

## 6 Agradecimientos

Este trabajo es el resultado del trabajo en equipo, que incluyó a un gran número de personas en la Fundación Omar Dengo, incluyendo a Tania Kaimowitz, Suhany Chavarría, Rosa Elena Chacón, Randall Alcázar, Oscar Pérez Ramírez, Magaly Zúñiga, Leonardo Correa, Leda Muñoz, Jorge Emilio López, Emmanuel Fonseca, Elena Carreras, Efraín López, Diana Matarrita, Ana Viria Hernández, Ana Lourdes Acuña, Ana Gabriela Castro, y Allan Otárola.

## Referencias

- Agudelo Velásquez, O. L., Salinas Ibáñez, J., & Zea Restrepo, C. (2012). Desarrollo de Competencias apoyado en Itinerarios de Aprendizaje Flexibles basados en Mapas Conceptuales. En A. J. Cañas, J. D. Novak, & J. Vanhear (Eds.), *Concept Maps: Theory, Methodology, Technology. Proc. of the Fifth Int. Conference on Concept Mapping*. Valletta, Malta: University of Malta.
- Cañas, A. J., & Badilla, E. (2005). Pensum No Lineal: Una Propuesta Innovadora para el Diseño de Planes de Estudio. *Actualidades Investigativas en Educación, Facultad de Educación, Universidad de Costa Rica, 5 Edición Especial*.
- Cañas, A. J., & Novak, J. D. (2010). Itineraries: Capturing 'Experience Using Concept Maps as Learning Object Organizers. En J. Sánchez, A. J. Cañas, & J. D. Novak (Eds.), *Concept Maps: Making Learning Meaningful. Proceedings of the Fourth International Conference on Concept Mapping* (Vol. 1). Viña del Mar, Chile: Universidad de Chile.
- Cañas, A. J., Novak, J. D., & Reiska, P. (2015). How good is my concept map? Am I a good Cmapper? *Knowledge Management & E-Learning: An International Journal (KM&EL)*, 7(1), 6-19.
- Chang, K.-e., Sung, Y.-t., & Chiou, S.-k. (2002). Use of Hierarchical Hyper Concept Map in Web-Based Courses. *J. Educational Computing Research*, 27(4), 335-353.
- Chiou, C., Huang, H., & Hsieh, J. (2004). Applying Hypermedia Assisted Concept Maps to Construct Accounting Inventory Teaching Material. *Journal of National Taipei Teachers College*, 17(2), 57-84.
- Coffey, J. W., & Cañas, A. J. (2003). LEO: A Learning Environment Organizer to Support Computer-Mediated Instruction. *Journal for Educational Technology*, 31(3), 275-290.
- Edmondson, K. M. (1995). Concept Mapping for the Development of Medical Curricula. *Journal of Research in Science Teaching*, 32(7), 777-793.
- Fundación Omar Dengo (2006). Educación y Tecnologías Digitales: cómo Valorar su Impacto Social y sus Contribuciones a la Equidad. In: Fundación Omar Dengo, San José, CR.
- Ghatasheh, N. A., Najdawi, A. R., Abu-Faraj, M. M., & Faris, H. (2013). Corporate e-Learning Environment Using Concept Maps: A Case Study. *International Review on Computers and Software*, 8(May), 2655-2662.
- Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. doi:10.3102/0013189X12463051
- Harel, I., & Papert, S. (1991). *Constructionism: Research Reports and Essays, 1985-1990*. Norwood, N.J.: Ablex Pub. Corp.
- Hazen, R. M., & Trefil, J. (2009). *Science Matters: Achieving Scientific Literacy*: Anchor.
- Martin, D. J. (1994). Concept Mapping as an Aid to Lesson Planning: A Longitudinal Study. *Journal of Elementary Science Education*, 6(2), 11-30. doi:10.1007/BF03173755
- McDaniel, E., Roth, B., & Miller, M. (2005). Concept Mapping as a Tool for Curriculum Design. *Issues in Information Science and Information Technology*, 2, 505-513.
- National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*: National Academies Press.
- Novak, J. D., & Cañas, A. J. (2008). *The Theory Underlying Concept Maps and How to Construct and Use Them* (IHMC CmapTools 2006-01 Rev 01-2008). Retrieved from Institute for Human and Machine Cognition (IHMC): <https://cmap.ihmc.us/docs/theory-of-concept-maps>

- Novak, J. D., & Gowin, D. B. (1984). *Learning How to Learn*. New York, NY: Cambridge University Press.
- Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2017). Empirical Studies on the Maker Movement, a Promising Approach to Learning: A Literature Review. *Entertainment Computing, 18*, 57-78.
- Papert, S. (1993). *Mindstorms : Children, Computers, and Powerful Ideas* (2nd ed.). New York: Basic Books.
- Sherborne, T. (2008). Mapping the Curriculum: How Concept Maps can Improve the Effectiveness of Course Development. In *Knowledge Cartography* (pp. 183-198): Springer.
- Starr, M. L., & Krajcik, J. S. (1990). Concept Maps as a Heuristic for Science Curriculum Development: toward Improvement in Process and Product. *Journal of Research in Science Teaching, 27*(10), 987-1000.
- Stewart, J. (1979). Concept Maps: A Tool for Use in Biology Teaching. *American Biology Teacher, 41*(3), 171-175.
- Walsh, N. (2017). Engaging with a Living Textbook: An Exploratory Study on the Way in which Students and Teachers Interact with and Perceive a Web Based Concept Map Visualization Tool.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3), 33-35.
- Wu, J. (2017). *Teach Less, Learn More*. Beijing: Posts & Telecom Press.
- Zamora, J. C. (2012). Programa Nacional de Informática Educativa (PRONIE-MEP-FOD), Costa Rica. *En: Las Tecnologías Digitales frente a los Desafíos de una Educación inclusiva en América Latina: algunos casos de Buenas Prácticas. Santiago: CEPAL, 2012. p. 53-77. LC/L. 3545.*