Sistemas & Telemática
ISSN: 1692-5238
EditorSyT@icesi.edu.co
Universidad ICESI
Colombia

Zúñiga Muñoz, René Fabián; Hurtado Alegría, Julio Ariel; Paderewsky Rodríguez, Patricia
Discovering the mechanisms of abstraction in the performance of work teams in children to solve computational problems
Sistemas & Telemática, vol. 14, núm. 36, 2016, pp. 69-87
Universidad ICESI
Cali, Colombia

Original Research / Artículo original - Tipo 1

Discovering the mechanisms of abstraction in the performance of work teams in children to solve computational problems

René Fabián Zúñiga Muñoz / fabianmunoz@unicauca.edu.co / Universidad del Cauca, Popayán-Colombia

Julio Ariel Hurtado Alegría / ahurtado@unicauca.edu.co / Universidad del Cauca, Popayán-Colombia

Patricia Paderewsky Rodríguez / patricia@ugr.es / Universidad de Granada, España

ABSTRACT The development of skills that allow children to perform satisfactorily in their training process and, later, in their work or social life, has become an objective for all educational and training models developed. This article deals with the relationship between thinking development skills, shared mental models and abstraction mechanisms, from a theoretical review and application with children aged between eight and twelve, from the Childprogramming methodology in a public education institution in Colombia. The results recorded at the end of the practices with this group of students, especially when assessing the progressive use of abstraction mechanisms in the Scratch environment, are presented, using the Dr. Scratch platform.

KEYWORDS Abstraction; computational thinking; teaching programming; children; study.

Descubriendo los mecanismos de abstracción en el desempeño de equipos de trabajo en niños al resolver problemas computacionales

RESUMEN El desarrollo de competencias que permitan a los niños un desempeño satisfactorio en su proceso de formación y, posteriormente, en su vida laboral o social, se ha convertido en un anhelo de todos los modelos educativos y de formación que se han desarrollado. El presente artículo trata de la relación entre el desarrollo de habilidades de pensamiento, los modelos mentales compartidos y los mecanismos de abstracción, a partir de una revisión teórica y la aplicación con niños de entre ocho y doce años, de la metodología Childprogramming, en una institución de educación pública en Colombia. Se presentan los resultados registrados al término de las prácticas realizadas con este grupo de estudiantes, especialmente al evaluar el uso progresivo de los mecanismos de abstracción en el entorno de Scratch, utilizando la plataforma Dr. Scratch.

PALABRAS CLAVE Abstracción; pensamiento computacional; enseñanza de la programación; niños; estudio.

Descobrindo os mecanismos de abstração no desempenho das equipes de trabalho com crianças para resolver problemas computacionais

RESUMO O desenvolvimento de competências que permitem às crianças um desempenho satisfatório no seu processo de formação e mais tarde no seu trabalho ou vida social, tornou-se um anseio de todos os modelos de educação e formação que têm sido desenvolvidos. Este artigo trata da relação entre o desenvolvimento de habilidades de pensamento, modelos mentais compartilhados e mecanismos de abstração, a partir de uma revisão teórica e a aplicação em crianças entre oito e doze anos, da metodologia Childprogramming em uma instituição de ensino pública na Colômbia. São apresentados os resultados registrados com este grupo de estudantes no final das práticas, especialmente quando se avalia a utilização progressiva de mecanismos de abstração no ambiente Scratch, usando a plataforma Dr. Scratch.

PALAVRAS-CHAVE Abstração; pensamento computacional; ensino de programação; crianças; estudo.

Received / Recepción: Marzo 15, 2016 - Accepted / Aceptación: Marzo 30, 2016 *doi: 10.18046/syt.v14i36.2216*

I. Introduction

The development of thinking skills that enable satisfactory performance in work teams, especially when working with children is involved, has become a subject of study that relates to the search that parents make for educational institutions that tend to offer an integral and contextualized training for their children (MEN, 2008).

The inclusion of ICT in education in our environment has become massive since 2001, with the creation, by the Government, of the Computers for Schools program (MINTIC, 2001), which has been established as an initiative to improve the access to technology in education, as a strategy to boost the development of the thinking skills and technology skills that can promote human development.

This state policy necessarily affects the educational dynamics having these technological means to support teaching and learning, and has forced it not only to think about what it can offer the child to perform a particular procedure (educational software), but how it should teach learning itself, extending to the mechanisms and ways of solving problems, applying a concrete and practical process abstraction. The advent of social networking and multimedia tools for learning and problem-solving through them have led to today's children becoming a new and very different generation. Bautista, Granados, Ortiz and Reinoso (2009) indicate that over time there have been various studies concerning the development of thinking in children, aiming to study how they think; the authors base their work on the formulation of questions with the purpose of defining, for example, what mental representation children have about thinking. Likewise, they present a comparison between three pedagogical contemporary references (Piaget, Vygotsky and Bruner), where they take a theoretical journey that goes from understanding the child as a unique being, through to social and cultural influences. It is not possible to assess the development of thinking skills alongside aspects of cognitive development as these authors present, and even more so when looking for work teams to present a satisfactory performance in development processes on computing solutions.

The development of computer applications involves hard work done individually. Developing skills for collaborative work and leadership in students is a key objective in education today (Hurtado, Collazos, Cruz, & Rojas, 2012), as is learning to solve computational problems effectively (Feierherd, Depetris, & Jerez, 2001). Jeannette Wing (2006) makes explicit reference to the role that analysis plays in education,

I. Introducción

El desarrollo de habilidades de pensamiento que permitan un desempeño satisfactorio en los equipos de trabajo, especialmente cuando se involucra el trabajo con niños, se ha convertido en un tema de estudio que se relaciona con la búsqueda que los padres de Familia hacen de instituciones educativas que propendan por ofrecer una formación integral y contextualizada para sus hijos (MEN, 2008)2006

La inclusión de las TIC en la educación en nuestro medio se ha masificado a partir de 2001, con la creación, por parte del Gobierno del programa *Computadores para Educar* (MINTIC, 2001), el cual se establece como una iniciativa para mejorar el acceso de la tecnología en la educación, como una estrategia para impulsar el desarrollo de habilidades de pensamiento y habilidades tecnológicas que promuevan el desarrollo humano.

Esta política estatal obligatoriamente afectó las dinámicas educativas al tener éstos medios tecnológicos como apoyo a los procesos de enseñanza y aprendizaje, lo cual obligo, no sólo a pensar en lo que se puede ofrecer al niño para realizar un determinado procedimiento (software educativo), sino que debe ir hacia el aprendizaje mismo del aprender, hacia los mecanismos y las formas de resolver los problemas, aplicando un proceso de abstracción concreto y práctico. La llegada de las redes sociales y de las herramientas multimedia para el aprendizaje, y la solución de problemas a través de las mismas, han llevado a los niños de hoy hacia una nueva y muy diferente generación. Bautista, Granados, Ortiz y Reinoso (2009) indican que a lo largo del tiempo se han realizado diferentes estudios referentes al desarrollo del pensamiento en los niños, intentando estudiar cómo piensan ellos; las autoras basan su trabajo en la formulación de preguntas para llegar a definir por ejemplo: ¿qué representaciones mentales tienen los niños sobre el pensamiento?, así mismo, presentan una comparación entre tres referentes pedagógicos contemporáneos (Piaget, Vigotsky y Bruner), en donde hacen un recorrido teórico que va, desde comprender al niño como un ser único, hasta pasar por la influencia social y cultural. No es posible evaluar el desarrollo de habilidades de pensamiento dejando de lado aspectos de desarrollo cognitivo como los que estos autores presentan, más aún al buscar que los equipos de trabajo presenten un desempeño satisfactorio en procesos de desarrollo de soluciones computacionales.

El desarrollo de aplicaciones informáticas implica un trabajo que difícilmente se logra de manera individual. El desarrollar competencias de trabajo colaborativo y liderazgo en los estudiantes, es un objetivo fundamental en la educación de hoy (Hurtado, Collazos, Cruz, & Rojas, 2012), así como el aprender a resolver problemas computacionales de forma efectiva (Feierherd, Depetris, & Jerez, 2001). Jeannette Wing (2006) hace una referencia explícita al papel que el análisis juega en la educación, argumentando que para desarrollar el pensamiento computacional, "se deben tener en cuenta ciertas habilidades para resolver problemas, diseñar sistemas, entender el comportamiento humano utilizando conceptos informáticos y una serie de herramientas mentales que reflejan la amplitud de las

aplicaciones informáticas". Según Wing, cuando una persona se enfrenta a un problema, intenta aplicar, de la mejor manera posible, ciertas habilidades; por ejemplo, al modelar una solución, cuando se describe el comportamiento del sistema se asegura que ese modelo funciona y soluciona el problema. Igualmente, combinar la recursión y la abstracción ofrece una gran herramienta para construir soluciones por capas de abstracción, con lo que se facilita enfocar los esfuerzos en cada capa y posteriormente pensar en las relaciones entre ellas.

Childprogramming (Hurtado et al., 2012) se presenta como una estrategia de desarrollo de software con niños, que aplica habilidades de pensamiento lógico matemático y competencias sociales, al privilegiar el trabajo colaborativo. Se aplican metodologías ágiles de desarrollo de software y el entorno del *Scratch* como ambiente de programación. Brennan y Resnick (2012) sugieren considerar las diferentes formas de conocimiento de los aprendices, debido a que no es suficiente estar en capacidad de definir y darles un concepto, por ejemplo: ¿qué es un ciclo?, ya que es realmente el aprendiz quien puede darle sentido al concepto cuando lo lleva a la práctica; sugieren que las evaluaciones de este tipo de pensamiento deben explorar las distintas formas de conocimiento; es allí donde hemos pretendido indagar sobre los procesos mentales de abstracción, los cuales se hacen evidentes a temprana edad, cuando los estudiantes buscan resolver problemas básicos relacionados con el desarrollo de software.

Jonker, Van Riemsdijk, y Vermeulen (2011) hacen referencia a los modelos mentales como una de las herramientas para entender problemas y proponer soluciones. Sin embargo, cuando la comprensión y la solución deben realizarse en el contexto de un equipo, que debe entender los procesos, las herramientas, los problemas, las soluciones y las acciones en forma compartida y alrededor de ello planear y coordinar, lograr esta comprensión, planeación y coordinación como equipo no es una tarea sencilla, lo cual se evidencia particularmente en el proceso *Childprogramming* como una dificultad para abstraer las tareas que conducirán al alcance de la misión (objetivo principal del proyecto de desarrollo). Aunque en *Childprogramming* se motiva el trabajo en equipo y la descomposición de tareas para alcanzar el objetivo final, el trabajo no estudia ni define estrategias computacionales (por ejemplo un método) para abstraer y descomponer la aplicación, ni para su posterior integración, lo cual dificulta hacer más operativa la estrategia metodológica como equipo. Estos aspectos han sido evaluados en un entorno real en la Institución Educativa Técnico Industrial de Popayán (Colombia), actividad cuyos resultados se presentan en este artículo.

Cómo nos preparamos para resolver problemas

Las ciencias humanas, en especial la pedagogía, se han interesado desde sus inicios en comprender qué sucede en el cerebro humano cuando nos enfrentamos a procesos de pensamiento. Bautista et al., (2009) desarrollaron su trabajo de investigación alrededor de los procesos de pensamiento que presentan los niños de edades entre ocho y nueve años. De su trabajo se extrae la comparación presente en la **Tabla 1**.

arguing that to develop computational thinking "it should take into account certain skills in solving problems, designing systems, understanding human behavior using computer concepts and a series of mental tools that reflect the breadth of applications". According to Wing, when a person faces a problem, he or she tries to apply, in the best way possible, certain skills; for example, when a solution is modeled, when the system behavior is described it ensures that the model works and solves the problem. Similarly, combined recursion and abstraction offers a great tool to build solutions by layers of abstraction, which makes it possible to focus efforts first on each layer and then think about the relationships between them.

Childprogramming (Hurtado et al., 2012) is presented as a software development strategy with children, which applies mathematical logical thinking skills and social skills, favoring collaborative work. Agile software development methodologies and the Scratch programming environment are applied. Brennan and Resnick (2012) suggest considering the different forms of knowledge of learners, because it is not enough to be able to define and give them a concept – for example: what is a cycle?– because it is really the learner who can make sense of the concept when put into practice; they suggest that evaluations of this kind of thinking should explore different forms of knowledge; this is where we have tried to investigate the mental processes of abstraction, which become evident at an early age, when students seek to solve basic problems related to software development.

Jonker, Van Riemsdijk, and Vermeulen (2011) refer to mental models as one of the tools to understand problems and propose solutions. However, when the understanding and the solution should be in the context of a team, which must understand the processes, tools, problems, solutions and actions on a shared basis and around this plan and coordinate, achieving this understanding, planning and coordination as a team is not an easy task, which is particularly evident in the *Childprogramming* process as a difficulty in abstracting the tasks that lead to the scope of the mission (the main objective of the development project). Although teamwork and task decomposition are motivated in *Childprogramming* to achieve the ultimate goal, the work does not study or define computational strategies (for example, a method) for abstracting and decomposing the application, or for further integration, making it difficult to make the methodological strategy operational as a team. These aspects have been evaluated in a real environment in the Industrial Technical Educational Institution of Popayan (Colombia), in an activity the results of which are presented in this article.

	Piaget	Vigotsky	Bruner
Thinking development / *Desarrollo de pensamiento.*	Relationship between representation and thought from biological development. / *Relación entre la representación y el pensamiento a partir del desarrollo biológico.*	Part of Piaget's postulates, but takes into account the social life of the child. / *Parte de los postulados de Piaget, pero tiene en cuenta la vida social del niño.*	The child learns according to the cultural needs. / *El niño aprende de acuerdo con las necesidades culturales.*
Aspects to consider for this investigation / *Aspectos a tener en cuenta para la presente investigación.*	Consider that the child is not an isolated being, but depends on social relationships to solve problems. / *Considerar que el niño no es un ser aislado, sino que depende de relaciones sociales para resolver problemas.*	Social life and language, helps children interpret the world. / *La vida social así como el lenguaje, ayuda al niño a interpretar el mundo.*	Importance of context, the child as being competent and regulatory author. / *Importancia del contexto, el niño como ser competente y autor regulador.*

Table 1. Definitions related to the concept of thinking development in children / Definiciones relacionadas con el concepto de desarrollo del pensamiento en los niños (Bautista et al., 2009)

How we prepare to solve problems

The human sciences, particularly pedagogy, have been interested from the beginning in understanding what happens in the human brain when faced with thought processes. Bautista et al. (2009) developed their research around the thought processes that children present aged between eight and nine years. From this work this comparison in **Table 1** is extracted.

Development of a way of thinking in finding solutions

Proposing a solution to a problem might seem a trivial matter. However, it is important to consider raising it, as affirmed by Wing (2006), who defined computational thinking as an approach to solve problems, design problems and understand human behavior, and as a skill to be developed around the world and not only by computer scientists. Wing makes explicit reference to the role of education in this development when she invites teachers to include analysis as part of their work in the classroom, arguing that to develop computational thinking must take into account certain skills in order to: solve problems, design systems and understand human behavior using computer concepts and a series of mental tools that reflect the breadth of computer applications.

The methods and computer models will provide the human capacity to solve problems and design systems as solutions. Reading, writing and doing arithmetic require computational thinking (Wing, 2014). Computational thinking involves a series of mental tools that reflect the broad spectrum of computer science:

- it is used to solve a difficult problem using another already known way of solving, either by reduction, or composition, transformation or simulation;
- it refers to thinking recursively, interpreting code as information and information as code, to solve a problem

Desarrollo de una forma de pensar en búsqueda de soluciones

Proponer una solución a un determinado problema pudiera parecer un asunto trivial, sin embargo, consideramos importante lo que plantea Wing (2006), quien ha definido el pensamiento computacional como una aproximación para resolver problemas, diseñar problemas y entender el comportamiento humano; como una habilidad que debe desarrollar todo el mundo y no únicamente los científicos de la computación. Wing hace referencia explícita al papel de la educación en este desarrollo cuando invita a los docentes a incluir el análisis como parte de su quehacer en el aula, argumentando que para desarrollar el pensamiento computacional se deben tener en cuenta ciertas habilidades para: resolver problemas, diseñar sistemas y entender el comportamiento humano utilizando conceptos informáticos y una serie de herramientas mentales que reflejan la amplitud de las aplicaciones informáticas.

Los métodos y modelos computacionales le brindan al ser humano las capacidades para resolver problemas y diseñar sistemas como soluciones. Leer, escribir y hacer operaciones aritméticas requieren del pensamiento computacional (Wing, 2014). El pensamiento computacional involucra una serie de herramientas mentales que reflejan el amplio espectro de la ciencia de la computación:

- consiste en resolver un problema difícil usando otro que ya sabemos resolver, bien sea por reducción, por composición, por transformación o por simulación;
- se refiere a pensar recursivamente, esto es a interpretar código como información e información como código, a solucionar un problema en términos de sí mismo hasta solucionar el problema trivial, más por su claridad y elegancia, que por su eficiencia.
- es usar la abstracción y la descomposición al abordar una tarea compleja o al diseñar un sistema complejo, es escoger la representación adecuada a un problema o modelar los aspectos relevantes de un problema para hacerlo manejable.

Una revisión inicial de los mecanismos de abstracción

Las ciencias de la computación referencian varios trabajos donde las entidades son caracterizadas como abstractas y varias actividades humanas son caracterizadas como actividades de abstracción. Uno de estos casos es programar, actividad que involucra la definición de tipos abstractos de datos y cuyo proceso involucra hacer abstracción desde la perspectiva de la información y su manipulación (Parnas, 1972). Las abstracciones en la ciencia de la computación están en constante evolución, lo cual requiere de variados mecanismos de abstracción, Algunos de esos mecanismos reportados en la literatura son:

- recursión: representar un problema o una solución en términos de sí mismo hasta llegar a un punto de solución trivial (League, 2000);
- descomposición/composición: descomponer un problema en sub-problemas de tal forma que se solucionen los sub-problemas y luego se obtenga la solución final por la composición de las piezas (Parnas, 1972);
- generalización: representar una familia de problemas o de soluciones en términos de un esquema reutilizable, parametrización o polimorfismo (Page-Jones, 1992);
- ocultamiento de la información: separar las piezas de software de tal forma que cada parte conozca pocos detalles de implementación de las otras (Parnas, 1972); y
- encapsulamiento: organizar un conjunto de conceptos relacionados –datos y funcionalidad, por ejemplo–, bajo una estructura unificada (Page-Jones, 1992).

Brennan y Resnick (2012) presentan una experiencia en el uso de *Scratch* como un entorno de programación amigable que permite la creación de historias, juegos y simulaciones interactivas y, mediante una comunidad en línea, además de compartir luego esas creaciones con otros programadores alrededor del mundo. Hace referencia al proceso de abstracción –construir algo grande uniendo colecciones de partes más pequeñas– como una práctica importante en el desarrollo de soluciones a problemas informáticos. Estos autores brindan un conjunto de elementos clave que servirán de punto de partida para la deducción y representación de los mecanismos de abstracción que en este trabajo se presentan, como el conjunto de referencia para determinar algunos aspectos prácticos sobre el uso de éstos mecanismos en el desarrollo del pensamiento computacional.

Ellos han definido un *framework* para el estudio y la evaluación del desarrollo del pensamiento computacional, en este marco de referencia se diferencian claramente tres dimensiones: conceptos computacionales, prácticas computacionales y perspectivas.

Su trabajo se direcciona en el uso *Scratch* como herramienta de programación, en este recorrido identifican siete conceptos que son claves a la hora de desarrollar los proyectos de *Scratch*, así como la transferencia a otros entornos en la enseñanza de la programación de computadores, estos son: repeticiones, paralelismo, eventos, condicionales, operadores y datos.

in terms of itself to solve a trivial problem, more for clarity and elegance, than for efficiency.

- it uses abstraction and decomposition in approaching a complex task or designing a complex system, choosing the appropriate representation of a problem or modeling the relevant aspects of a problem to make it manageable.

An initial review of the mechanisms of abstraction

Computer science refers to several jobs where entities are characterized as abstract and various human activities which are characterized as abstraction activities. One of these cases is programming, an activity that involves the definition of abstract data types and a process that involves abstracting from the perspective of information and handling (Parnas, 1972). Abstractions in computer science are constantly evolving, which requires various mechanisms of abstraction. Some of these mechanisms reported in the literature are:

- recursion: this presents a problem or a solution in terms of itself until it reaches a point of a trivial solution (League, 2000);
- decomposition/composition: breaking a problem into sub-problems so that the sub-problems are solved and then the final solution is obtained by the composition of the pieces (Parnas, 1972);
- generalization: representing a family of problems or solutions in terms of a reusable scheme, parameterization or polymorphism (Page-Jones, 1992);
- concealment of information: separating pieces of software so that each party knows only a few details of the implementation of the other (Parnas, 1972); and
- encapsulation: data organized into a set of related concepts and functionality, for example, under a unified structure (Page-Jones, 1992).

Brennan and Resnick (2012) present an experiment using *Scratch* as an environment-friendly programming tool that allows the creation of stories, games and interactive simulations and, through an online community, will then share these creations with other programmers around the world. It refers to the process of abstraction –building something great by uniting collections of smaller parts– as an important part of the development of practical solutions to computer problems. These authors provide a set of key elements that will serve as a starting

point for the deduction and representation of abstraction mechanisms in this work, as the reference set to determine some practical aspects on the use of these mechanisms in the development of computational thinking.

They have defined a *framework* for the study and evaluation of the development of computational thinking, in which there are three clearly different dimensions: computational concepts, computational practices and perspectives.

Their work is addressed in using *Scratch* as a programming tool, in which journey we identify several concepts that are key when developing *Scratch projects*, and also transferring to other environments in teaching computer programming. These are: repetitions, parallelism, events, conditionals, operators and data.

The practices which are developed in their proposal are focused on the processes of thinking and learning, to try to understand what is being learned and how students are learning, and also to confirm some situations that have been evident in the development of this work, which are concerned with how the children develop skills using multimedia environments. The following practices have been defined and show some important details:

- use of incremental and interactive elements, which refers to the need to plan the design of a solution as an adaptive process that may involve changes, especially considering that the solution is achieved by making small steps;
- test and practice, which define sequential steps to develop these elements: identify the root of the problem, review the scripts, experiment with the scripts, try to write the scripts again, find scripts that function properly, talk to someone else about the problem and rest;
- reuse and mix, using something that is already developed and has been implemented long ago in the programming; also the authors present three interesting questions: what is reasonable to copy from others?, how is credit given appropriately to the other? how to evaluate cooperative work and collaborative work?;
- abstraction and modularity, which means building something great by uniting smaller parts, something that in *Scratch* is an important practice for the whole design and implementation of a solution; this abstraction tool is applied in different ways, initially with

Las prácticas sobre las cuales desarrollaron su propuesta se enfocan en los procesos de pensamiento y aprendizaje, para tratar de entender qué se está aprendiendo y cómo se está aprendiendo, así mismo confirman algunas situaciones que han sido evidentes en el desarrollo del presente trabajo, que tienen que ver con las habilidades que desarrollan los niños, utilizando entornos multimediales. A continuación se presenta las prácticas que han definido y se muestra algunos detalles importantes:

- uso de elementos interactivos e incrementales, que se refiere a la necesidad de planear el diseño de una solución como un proceso adaptativo que puede involucrar cambios, sobre todo pensando en que la solución se alcanza realizando pequeños pasos;
- pruebas y prácticas, que definen unos pasos secuenciales para desarrollar estos elementos: identificar la raíz del problema, revisar los scripts, experimentar con los scripts, tratar de escribir los scripts nuevamente, encontrar scripts que funciones correctamente, hablar con alguien mas acerca del problema y descansar;
- reutilización y mezcla, utilizar algo que ya esta desarrollado se ha implementado desde hace mucho en la programación, así mismo los autores plantean tres preguntas interesantes: ¿qué es razonable al copiar de otros?, ¿cómo se da crédito apropiadamente a los otros?, y ¿cómo evaluar el trabajo cooperativo y el trabajo colaborativo?;
- abstracción y modularidad, esto es construir algo grande uniendo partes mas pequeñas, algo que en *Scratch* es una importante practica para todo el diseño y la aplicación de una solución, en esta herramienta se aplica la abstracción en diferentes maneras, inicialmente con la concepción del problema para luego traducirlo en el uso de personajes y bloques de código.

La abstracción como componente critico de la formación en ciencias computacionales

Serna (2011) presenta un trabajo basado en su experiencia como docente en áreas relacionadas con el desarrollo de software a nivel universitario, formula diversos interrogantes cruciales para la investigación que se propone; creer que la capacidad de desarrollar software es cuestión de inteligencia o de simple entrenamiento es, para el autor, una equivocación; por el contrario, considera que la clave está en la capacidad de realizar y aplicar pensamiento abstracto y en las habilidades que se posea para ella. El autor plantea dos soluciones: una, relacionada con la formación al interior de las universidades, en la cual se pueda medir el desarrollo de las habilidades de abstracción; otra, que tiene que ver con los mecanismos que deberían seguir para evaluar si se poseen habilidades de abstracción al inscribirse en la universidad.

Moroni (2001) presenta una forma de enseñar a programar aplicando diversos entornos computacionales, específicamente el entorno *Cubik* y su entorno de verificación. El primero permite el trabajo con programación estructurada y modular, mientras que el segundo permite visualizar la conducta del

algoritmo y el programa durante la ejecución. En esta propuesta no se busca introducir un paradigma de desarrollo, sino extraerlo empíricamente desde la práctica y posteriormente proponer un método que facilite técnicas de descomposición e integración cercanas al modelo mental de los niños.

Herramientas computacionales para desarrollar el pensamiento computacional

Brennan y Resnick (2012) describen el uso de *Scratch* como un entorno de programación amigable que permite la creación de historias, juegos y simulaciones interactivas, mediante una comunidad en línea, además de compartir luego esas creaciones con otros programadores alrededor del mundo; hacen una referencia al proceso de abstracción –construir algo grande uniendo colecciones de partes más pequeñas– como una práctica importante en el desarrollo de soluciones a problemas informáticos.

Igualmente se encontramos a Snap! (Mönig, 2015) lenguaje de programación visual que se presenta como una extensión de *Scratch*, adicionando herramientas y funcionalidades que permiten el desarrollo de cursos iniciales de programación en entornos de educación superior, que cuentan con el apoyo de la *National Science Foundation* [NSF] para desarrollar sus proyectos de aplicación.

Los modelos mentales, una característica humana que debemos potenciar

Letsky, Warner, Fiore, y Smith (2008) hacen referencia a diversos autores (Laird, Rouse & Morris, Mohammed y McComb) quienes definen los modelos mentales de diferentes maneras, desde una definición simple como que ellos son caracterizaciones simples de sus propios mundos, así como que los modelos mentales son utilizados para describir, explicar y predecir su entorno, y que constan de contenido y relaciones y estructuras entre ellos.

Moreira (1999) presenta una revisión de la teoría relacionada con los modelos mentales enfocados desde las propuestas hechas por Laird, así mismo, un acercamiento a la aplicación de estos modelos en la enseñanza de las ciencias. Este trabajo centra sus afirmaciones en conceptos como representaciones mentales, maneras de "re-presentar" el mundo externo a partir de la construcción de representaciones del mismo, las cuales pueden ser representaciones analógicas y proposicionales, que se diferencian a partir de la manera cómo se organizan en la mente –siguiendo reglas o de manera imprevista, a medida que se encuentra la información– y tienen en cuenta cómo son expresadas y representadas utilizando el lenguaje humano. Aunque el autor se refiere al lenguaje mental, como la forma de representación que interactúa con el lenguaje verbal para representar lo que sucede a nuestro alrededor.

Según Johnson-Laird (1986) estas imágenes se construyen internamente, a partir de un punto de vista particular que no necesariamente se relaciona con la apropiación de los modelos mentales explicativos y predictivos. Como resultado de este estudio el autor plantea un punto de vista que, para la presente

the conception of the problem and then translated into the use of characters and code blocks.

Abstraction as a critical component of training in computer science

Serna (2011) presents a work based on his experience as a teacher in related software development at university level, in which many crucial questions are formulated for the proposed research areas. The belief that the ability to develop software is a matter of intelligence or simple training is, for the author, a mistake. On the contrary, he believes that the key is the ability to perform and apply abstract thinking and to possess the skills for it. The author proposes two solutions: the first one, related to training within universities, which can measure the development of the skills of abstraction; the second, which has to do with the mechanisms to assess if a student –at the moment of enrollment in college– has the abstraction skills required.

Moroni (2001) presents a way to teach programming using various computing environments, specifically the Cubik environment and its verification environment. The first allows working with structured and modular programming, while the second makes it possible to visualize the behavior of the algorithm and the program during execution. This proposal does not seek to introduce a development paradigm, but is empirically removed from practice and then a method is proposed to facilitate decomposition techniques and close integration with the mental model of children.

Computational tools to develop computational thinking

Brennan and Resnick (2012) describe the use of *Scratch* as a friendly programming environment that allows the creation of stories, games and interactive simulations, using an online community, which will then share these creations with other programmers around the world. They refer to the process of abstraction –building something great by uniting collections of smaller entities– as an important part of the development of practical solutions to computer problems.

Later, the Snap! (Mönig, 2015) visual programming language is presented as an extension of *Scratch*, adding tools and features that allow the development of initial programming courses in higher education environments, which are supported by the *National Science Foundation* [NSF] to develop their implementation projects.

Mental models, a human characteristic that we must strengthen

Letsky, Warner, Fiore, and Smith (2008) refers to several authors (Laird, Rouse & Morris, Mohammed and McComb) who define mental models in different ways, from a simple definition as simple characterizations of their own worlds, stating that mental models are used to describe, explain and predict their environment, and consist of content and relationships and structures among them.

Moreira (1999) presents a review of the theory related to mental models focused on the proposals made by Laird, and also an approach to the application of these models in science education. This work focuses its assertions on concepts such as mental representations, ways of "re-presenting" the external world from the construction of representations of itself. These may be analog and propositional representations, which differ in the way they are organized in the mind –following rules or spontaneously, as is the information– and take into account how they are expressed and represented using human language. Moreover, the author refers to mental language, as the form of representation that interacts with verbal language to represent what happens around us.

According to Johnson-Laird (1986), these images are internally built, from a particular point of view that is not necessarily related to ownership of the explanatory and predictive mental models. As a result of this study, the author presents the point of view that, for this investigation, we will take as a pillar of the same: that education should develop conceptual models, materials and instructional strategies that allow students to build mental models correctly, bearing in mind that investigación, tomaremos como pilar de la misma: que en la enseñanza se deben desarrollar modelos conceptuales, materiales y estrategias instruccionales que permitan a los estudiantes la construcción de modelos mentales adecuados, pensando que las personas no desarrollan modelos claros y elegantes (como lo afirma el autor) sino que más bien presentan modelos mentales desordenados, incompletos e inestables.

De acuerdo con Letsky (2008), de los modelos mentales se desprende una definición que sirve de soporte a la presente investigación; sin embargo, los modelos mentales de cada uno de los integrantes del grupo deben converger, hacerse similares y compartirse, hasta convertirse en un modelo mental del grupo, conocido como modelo mental compartido. Estas estructuras mentales permiten a los equipos formar explicaciones precisas de las tareas asignadas para coordinar sus acciones y proponer una solución a la misma.

Estos autores han realizado una meticulosa revisión de la evolución de los conceptos relacionados con los modelos mentales compartidos y el trabajo en equipo, en la Tabla 2 se muestran algunas de las definiciones, donde se identifican varios detalles que se refieren a los modelos mentales como elementos fundamentales en el rendimiento, las relaciones y el ambiente del trabajo en equipo, sin descuidar cómo son estructuradas las tareas que se les asignan. De acuerdo con Staats (1996)Arthur Staats synthesizes more than four decades of research, theory, and study into a new generation of behaviorism that offers insights and future directions for researchers, professionals, and students. Staats's unified theory of psychological behaviorism builds on current theories in child development, personality, psychological measurement, and abnormal behavior. His theoretical model provides new ways to consider human behavior as a whole that will have implications for research, theory, and practice.","ISBN":"978-0-8261-9311-7","shortTitle":"BehaviorandPersonality","language":"en","author":[{"family":"Walter W. Staats","given":""}],"issued":{"date-par

Authors / *Autores*	Definitions / *Definición*
Johnson-Laird (1983).	Mental models are simple characterizations that humans create from their world. / *Los modelos mentales son caracterizaciones simples que los humanos crean de su mundo.*
Rouse y Morris (1986).	Individuals use mental models to describe, explain and predict their environment. / *Los individuos usan los modelos mentales para describir, explicar y predecir su entorno.*
Mohammed et al. (2000).	Mental models are formed by content and any structure or structure between this content. / *Los modelos mentales se constituyen de contenido y cualquier relación o estructura entre este contenido.*
McComb (2007); McComb & Vozdolska (2007).	When individuals interact, their mental models converge, which is a scenario where each individual mental models become similar or shared with this or that model or models of their peers. / *Cuando los individuos interactúan, sus modelos mentales convergen, de lo que resulta un escenario donde los modelos mentales de cada individuo se vuelven similares o compartidos con tal o cual modelo o modelos de sus compañeros.*
Cannon-Bowers et al., (1993).	Knowledge structures held by team members that allow them to be accurate explanations and expectations for the task and, in turn, coordinate their actions and adapt their behavior to the demands of the task and other team members. / *Estructuras de conocimiento sostenidas por los miembros de un equipo que les permiten formar explicaciones exactas y expectativas para la tarea y, a su turno, coordinar sus acciones y adaptar su comportamiento a las demandas de la tarea y otros miembros de equipo.*

Table 2. Definitions related to the concept of mental models / Definiciones relacionadas con el concepto de modelos mentales

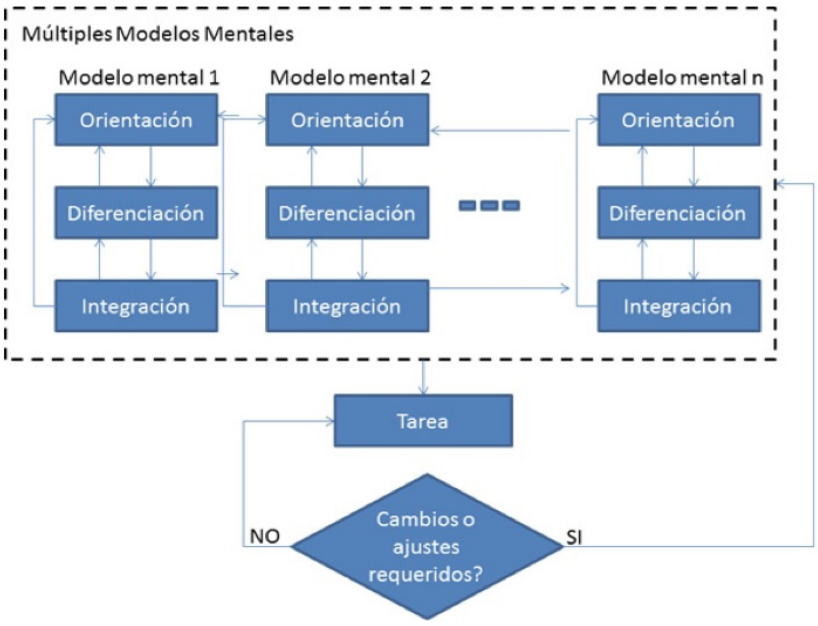

Figure 1. Convergence process of mental models / Proceso de convergencia de los modelos mentales

ts":[["1996",9,3]]}}}],"schema":"https://github.com/citation-style-language/schema/raw/master/csl-citation.json"} , realizar tareas no estructuradas, hace que se desarrollen muchos más modelos mentales. Hemos encontrado diversas definiciones y a continuación las presentamos en la **Tabla 2**.

Letsky (2008) presenta tres fases en las cuales los equipos desarrollan sus modelos mentales: orientación, dirigirse hacia su propio dominio; diferenciación, crear su propio punto de vista de la situación, similar o no al de sus compañeros; e integración, permitir el desarrollo o la evolución de su propia perspectiva hacia la visión del equipo. La Figura 1 presenta el proceso de convergencia propuesto.

Los procesos de aprendizaje, en cualquier nivel educativo, se ven afectados por diversas corrientes y guías pedagógicas, pensar en la capacidad mental de los estudiantes, no solo en términos de retención de información, sino en la posibilidad de aprender a partir del registro que se tiene en la mente, es algo que la teoría de los modelos mentales trata de abordar. A continuación se presentan una serie de conceptos y experiencias relacionadas con el tema, las experiencias que se han tomado como referencia para el presente trabajo se han desarrollado en niveles de educación superior.

Rendón y Parra (2009) han hecho referencia a los modelos mentales como un tema relacionado con la psicología cognoscitiva, en donde se trata de descubrir los procesos que ocurren en la mente, tales como: procesamiento de información, organización, recordación y uso de la información. Las representaciones de los conceptos tienen que ver con la combinación de conocimientos, habilidades y actitudes que conducen a un desempeño adecuado y oportuno en varios contextos. No es sólo saber sobre algo, es hacerlo con determinadas actitudes

people do not develop clear and elegant models (as stated by the author) but rather they present messy, incomplete and unstable mental models.

According to Letsky (2008), the definition of mental models that supports the present research is clear; however, the mental models of each of the group members must converge, become similar and shared, in order to become a mental model of the group, known as a shared mental model. These mental structures allow teams to form precise explanations of the tasks assigned, to coordinate their actions and propose a solution to them.

These authors have conducted a thorough review of the evolution of the concepts related to shared mental models and teamwork. Table 2 presents some of the definitions, where several details relating to mental models like elements are identified including fundamental performance, relationships and the teamwork environment, without neglecting how the tasks assigned to them are structured. According to Staats (1996), performing unstructured tasks means many more mental models are developed. We have found various definitions and present these in **Table 2**.

Letsky (2008) presents three phases in which teams develop their mental models: orientation, addressed to its own domain; differentiation, creating one's own view of the situation, similar or not to one's peers; and integration, allowing the development or evolution of one's own perspective on the team's vision. Figure 1 shows the proposed convergence process.

Learning processes, at any level of education, are affected by various trends and teaching guides, thinking about the mental capacity of students, not only in terms of the retention of information, but the ability to learn from the record that one has in mind, and this is something that the theory of mental models seeks to approach. Below a number of concepts and experiences related to the topic are presented, experiences that have been taken as a reference for this work that have been developed at the higher education level.

Rendon (2009) has referred to mental models as a topic related to cognitive psychology, where it comes to discovering the processes occurring in the mind, such as:

information processing, organization, recall and use of information. Representations of concepts have to do with the combination of knowledge, skills and attitudes that lead to adequate and timely performance in various contexts. Not only do you know about something, but you do it with certain attitudes (measured knowledge, skills, attitudes to provide a result, "do good things").

In his study he refers to Arbelaez, who analyzed mental representations as the organization in conceptual, procedural and attitudinal structures. Rendón and Parra (2009) refer to such representations as a mixture of knowledge, skills and attitudes that are looking for tools to allow proper performance in different contexts. That is, there is not only knowledge, but know-how and being, to include personal attitudes; for our work we include collaborative work as an essential component of the development model on which the research is based.

Rodriguez, Marrero, and Moreira (2001) conducted a study related to the application of the theory of mental models in a university course for students studying the cell concept. We consider this a valuable study because it refers to a specific component of knowledge and our research refers to the process of learning the concepts related to the solution of computational problems. The authors base their work on the concepts of Johnson-Laird (1986), who has become an obligatory reference in several works related to studies of the theory of mental models. Johnson-Laird says that mental models arise from the need to explain the processes of cognition (comprehension and inference), and proposes three states to represent the content in mind: recursive procedures, propositional representations and models. He also uses images, prototypes or schematics that can manage variables in the models. Proposed as data sources are: initial and final questionnaires, surveys conducted during the course, concept maps, interpretations, development of a drawing of the structure and operation, and final interview. At the end of the study it presents results for four mental models on which we base the first field activities in our research:

- Structure, nonfunctional, single, static image, without inferring or deducing anything;
- Dual, builds structural and functional models, dual model, few inferences and deductions;
- Causal discursive, integrated model structure/inferences function and deductions made, causality, static and simple images; and

(mide conocimientos, habilidades, actitudes para proporcionar un resultado "hacer bien las cosas").

En su estudio referencia a Arbeláez, quien analizan las representaciones mentales como la organización en estructuras conceptuales, procedimentales y actitudinales. Rendón (2009) hace referencia a esa representación como una mezcla de conocimientos, habilidades y actitudes que se busca que sean las herramientas para permitir un desempeño apropiado en contextos diferentes. Es decir, no sólo el saber, sino el saber hacer y el ser, al incluir actitudes personales; para nuestro trabajo incluimos el trabajo colaborativo como componente indispensable del desarrollo del modelo sobre el cual se basa la investigación.

Rodríguez, Marrero, y Moreira (2001) realizan un estudio relacionado con la aplicación de la teoría de los modelos mentales en un curso de estudiantes universitarios al estudiar el concepto de la célula. Consideramos valioso su estudio porque hace referencia a un componente de conocimiento específico y nuestra investigación se refiere al proceso de aprendizaje de los conceptos relacionados con la solución de problemas computacionales. Los autores basan su trabajo en los conceptos de Johnson-Laird (1986), quien se ha convertido en un referente obligado en diversos estudios relacionados con la teoría de los modelos mentales. Johnson-Laird expresa que los modelos mentales surgen de la necesidad de explicar los procesos de cognición (comprensión e inferencia), y propone tres estados para representar el contenido en la mente: procedimientos recursivos, representaciones proposicionales y modelos. Igualmente utiliza imágenes, prototipos o esquemas que pueden administrar variables en los modelos. Propone como fuentes de datos: cuestionarios iniciales y finales, exámenes realizados durante el curso, mapas conceptuales, interpretaciones, elaboración de un dibujo relativo a la estructura y al funcionamiento, y entrevista final. Al finalizar su estudio presenta como resultado cuatro modelos mentales sobre los cuales basamos las primeras actividades de campo en nuestra investigación:

- Estructural, no funcional, imagen única y estática, sin inferir ni deducir nada;
- Dual, construye modelos estructural y funcional, doble modelo, pocas inferencias y deducciones;
- Causal discursivo, modelo integrado estructura/funcionamiento inferencias y deducciones elaboradas, causalidad, imágenes estáticas y simples; y
- Casual imagístico, modelo integrado, estructura/funcionamiento, inferencias y deducciones elaboradas, causalidad, imágenes dinámicas-complejas y/o varias imágenes.

Estudio

Teniendo en cuenta que la investigación tiene como base el método *Childprogramming*, y que la herramienta informática para trabajar con los estudiantes es *Scratch*, se definieron dos estudios de caso: el primero, dirigido a una población objetivo formada por estudiantes de séptimo grado (algunos de ellos habían participado en la definición del método Childprogram-

ming dos años atrás; el segundo, dirigido a estudiantes de noveno grado, sin experiencia en el método Childprogramming.

El presente trabajo busca aportar a la investigación, a la innovación y a la sociedad:

- Investigación: contribuir a la generación de nuevo conocimiento en relación con la interacción de las TIC con la educación, específicamente en el estudio de la aplicación de los modelos mentales y la exploración de mecanismos de abstracción en la solución de problemas computacionales en estudiantes de nivel básico, aplicando técnicas de trabajo colaborativo enmarcadas en el modelo de Childprogramming. La investigación se presenta en dos modelos: exploratorio y descriptivo, y busca responder: ¿es posible aplicar la teoría de los modelos mentales compartidos en la solución de problemas computacionales en niños de edad escolar básica? y, ¿cuáles son las mejores prácticas que se pueden desarrollar en el aula para lograr un rendimiento eficiente en la solución de los problemas computacionales?
- Innovación: las prácticas de aula se implementarán utilizando *Scratch* y serán documentadas y organizadas como guías de trabajo para los docentes, las que, a su vez, podrán ser adaptadas posteriormente a cualquier tipo de práctica relacionada con el desarrollo de aplicaciones en entornos educativos relacionados con la formación en programación de computadoras.
- Social: el haber realizado las prácticas en un entorno real de aprendizaje contribuirá a mejorar las prácticas de aula y a aprovechar al máximo las capacidades de trabajo de los estudiantes, promoviendo así un aprendizaje significativo en las escuelas.

Las preguntas de investigación son: ¿cuál es el impacto en términos de efectividad de los modelos mentales compartidos (usados como un mecanismo para facilitar la abstracción compartida de problemas y soluciones) en el contexto de Childprogramming?, y ¿cuáles son los mecanismos de abstracción que utilizan los niños entre los ocho y doce años para solucionar y descomponer problemas computacionales en el contexto de Childprogramming?

II. Método

Se usó, como marco, el método general de investigación científica, específicamente el método científico (Bunge, 2000); para la evaluación de la propuesta se utilizó el método de estudio de caso (Runeson & Höst, 2009).

Al iniciar la investigación se realizó una revisión de la teoría relacionada con los procesos de abstracción y la aplicación de los modelos mentales compartidos en la solución de problemas computacionales. Posteriormente se realizaron sesiones de trabajo con grupos de estudiantes en un ambiente real de formación, las cuales buscan, inicialmente, lograr el acercamiento a la herramienta *Scratch* con la que se desarrollarán los proyectos de aula; para cada una de estas sesiones se elaboró una guía de trabajo que se ha definido como un instrumento de recolección primaria de datos.

- Causal imagistic, integrated model, structure/function, inferences and deductions made, causality, dynamic-complex images and/or multiple images.

Study

Given that this research is based on the *Childprogramming* method, and the computer tool to work with students is *Scratch*, two case studies were defined: the first, aimed at a target population consisting of seventh graders (some of whom had participated in defining the *Childprogramming* method two years ago); the second, aimed at students from the ninth grade, inexperienced in the *Childprogramming* method.

The present paper seeks to contribute to research, innovation and society:

- Research: contribute to the generation of new knowledge regarding the interaction of ICT with education, specifically in the study of the application of mental models and exploring abstraction mechanisms in solving computational problems in entry-level students applying techniques of collaborative work framed in the *Childprogramming* model. The research is presented in two models: exploratory and descriptive, and seeks to answer: is it possible to apply the theory of shared mental models in solving computational problems in basic school age children? and, what are the best practices that can be developed in the classroom to achieve efficient performance in solving computational problems?
- Innovation: classroom practices will be implemented using *Scratch* and will be documented and organized as working guidelines for teachers, which, in turn, may subsequently be adapted to any type of practice related to application development in educational settings related with training in computer programming.
- Social: having made this practical in a real learning environment will help to improve classroom practice and to maximize the students' work capabilities, thus promoting meaningful learning in schools.

The research questions are: what is the impact in terms of effectiveness of shared mental models (used as a mechanism to facilitate the shared abstraction of problems and solutions) in the context of *Childprogramming*?, and what are the mechanisms of abstraction used by children aged between eight and twelve years to solve computational problems and decompose in the context of *Childprogramming*?

II. Method

The general method of scientific research was used as a framework, specifically the scientific method of (Bunge, 2000); for the evaluation of the proposed method case study (Runeson & Höst, 2009) was used.

Initiating the investigation, a review of the theory related to the processes of abstraction and the implementation of shared mental models in solving computational problems was performed. Later, sessions with groups of students were conducted in a real training environment, seeking, initially, an approach to using the *Scratch* tool that the classroom projects will develop; for each of these sessions a working guide that has been defined as an instrument of primary data collection was developed.

These classroom activities were held for ten months in sessions every eight days, lasting six hours each. The activities were documented so that they became inputs to organize teacher guides.

Instruments

- The student group must have clarity at the beginning of each work session, the teacher introduces the type of activity to be performed and resolves any doubts that the students have about it, later allowing each team of students to work as they choose in the computer room.

Estas actividades de aula se realizaron durante diez meses, en sesiones cada ocho días, con una duración de seis horas en cada sesión. Las actividades se documentaron de tal manera que se convirtieron en insumos para organizar las guías de los docentes.

Instrumentos

A continuación se describen los momentos de trabajo con los grupos, sin el ánimo de convertir el texto en una receta para ser seguida en estricto orden, puesto que es claro que existen factores diferenciadores en cada institución. Las tres prácticas desarrolladas hasta el momento son:

- El grupo de estudiantes debe tener la claridad de que al inicio de cada sesión de trabajo, el docente socializa el tipo de actividad a realizar y resuelve las dudas que los estudiantes tengan de la misma, para posteriormente permitir que cada equipo de estudiantes trabaje a su gusto en la sala de cómputo.
- Las primeras dos prácticas se proponen como ejercicios para conocer la herramienta y entender los conceptos computacionales básicos relacionados con el paradigma de programación estructurada, sin hacer referencia explícita a dichos temas, sino buscando que a partir de la práctica se interioricen empíricamente.
- En la tercera práctica los equipos de trabajo proponen una solución a un problema a partir de las indicaciones iniciales que el docente socialice, en esta práctica se busca afirmar conceptos como manejo de variables y estructu-

Practice Nº 1. / *Práctica Nº*	I recognize what items I can use scratch / *Aquí se coloca un título descriptivo.*
Purpose: the teacher determines the purpose of the practice from the computational point of view. / *Propósito: El docente determina el propósito de la práctica desde el punto de vista computacional.*	
Working format / *Formato de trabajo:*	
What ask me: description of the activity that the teacher determines for each working group / *Qué me piden: descripción de la actividad que el docente determina para cada grupo de trabajo.*	
What I have available / *Qué tengo a disposición:*	
List drawn up by the teams where they determine the elements of the computational tool that considered necessary to propose a solution / *Listado elaborado por los equipos de trabajo donde determinan los elementos de la herramienta computacional que consideran necesarios para proponer una solución.*	
How I think the solution: at this point children should write, in their own words, how can it be the solution, without going into detail, it should be clear to them that what they write is, generally, a reference to this solution. / *Como pienso la solución: en este punto los niños deben escribir, en sus propias palabras, cómo puede ser la solución, sin entrar en los detalles, debe ser claro para ellos que lo que escriban es, en general, una referencia a esta solución.*	
How I organize tabs: here should explain that children have to organize, organized and sequential manner, the steps to be followed so that the sprite smoothly perform the movements and actions they consider to achieve the purpose. / *Cómo organizo las fichas: aquí se debe explicar que los niños tienen que organizar, de manera organizada y secuencial, los pasos que se deben seguir para que el sprite realice sin inconvenientes los movimientos y las acciones que ellos consideren para lograr el propósito.*	
¿What I did work?: yes __ no __ / *¿Lo que hice funcionó?: si __ no __*	
If it was not appropriate, you must complete the form / *Si no fue apropiada, debe completar el formulario.*	
What should I change: it is necessary that children are able to describe what failed and what to do to make the changes that define fulfill the purpose. / *Qué debo cambiar: es necesario que los niños sean capaces de describir en qué fallaron y qué deben hacer para que los cambios que definan cumplan con el propósito.*	
Practice result: / *Resultado de la práctica:*	
Each team should make a description of the result of the activity, taking into account positive aspects and moments to take into account in subsequent activities. / *Cada equipo de trabajo debe hacer una descripción del resultado de la actividad, teniendo en cuenta aspectos positivos y momentos para tener en cuenta en posteriores actividades.*	

Table 3. Job template document that teams should work with / Plantilla de trabajo que debieron documentar los equipos de trabajo

Practice Nº 1. / *Práctica Nº 1.*	I recognize what items I can use scratch / *Reconozco qué elementos del scratch puedo utilizar*
Purpose: To ensure that students make recognition of scratch tools environment that have learned to use from the recognition session of the tool. / *Propósito: Lograr que los estudiantes hagan un reconocimiento de las herramientas del entorno scratch que han aprendido a utilizar a partir de la sesión de reconocimiento de la herramienta.*	
Why ask me: Write a list of the tools using the scratch program / *Qué me piden: Escribo una lista de las herramientas que se cómo utilizar del programa scratch*	
Component name / *Nombre del componente*	Use / *Para qué me sirve*
Result of practice / *Resultado de la práctica:*	
Mark where think is right (can mark several options). / *Marque donde crea correcto (pueden marcar varias opciones).*	
1- The list was made by only one member of the group / *La lista la hizo solo un integrante del grupo.* ____	
2- The list was made by both members of the group / *La lista fue realizada por ambos integrantes del grupo.* _____	
3- There are elements that one of the members did not remember to use / *Hay elementos que uno de los integrantes no recordaba utilizar.* ____	
4- All elements are known to the group / *Todos los elementos son conocidos por los integrantes del grupo* _____	
5- There are components that are not explained in the previous class, but that was then discovered to do practice outside of class time / *Hay componentes que no se explicaron en la clase anterior, pero que se descubrieron luego de hacer la practica fuera del tiempo de clase.* ____	
6- Having several days without using affect the number of tools that recall the work team program / *El tener varios días sin utilizar el programa afecto el número de herramientas que recuerdan en el equipo de trabajo.* _____	
7- In the group one of its members working in the program after the previous class / *En el grupo alguno de sus integrantes trabajo en el programa luego de la clase anterior.* _____	
8- In the group, all members worked in the program after the previous class / *En el grupo todos sus integrantes trabajaron en el programa luego de la clase anterior*_____	

Table 4. Revised job template document / Plantilla de trabajo que debieron documentar los equipos de trabajo

ras de repetición. Los equipos, en este momento, deben controlar su desempeño a partir del uso de la plantilla que muestra la **Tabla 3**.

Esta plantilla fue adaptada (ver **Tabla 4**) cuando los equipos ya habían desarrollado las actividades iniciales con la herramienta.

Procedimiento

Las prácticas que se proponen para los grupos van incrementando su complejidad, los equipos van recorriendo y practicando conceptos computacionales sin que sea explícito en qué se está en una actividad de aprendizaje formal (**Figura 2**).

Las practicas posteriores, donde se aplica el formato de la **Tabla 4**, ya se presentan más elaboradas (**Figura 3**). En la Figura 4 se verán las prácticas donde los equipos crean sus propios métodos, con lo cual se evidencia un mayor proceso de abstracción.

III. Resultados

En los primeros casos de estudio los grupos evidenciaron, en su mayoría, el uso de mecanismos de abstracción tales como la separación de intereses –evidenciada por la organización en algoritmos distintos para cada evento de un mismo personaje– y la composición de elementos compuestos a partir de elementos simples –la organización de elementos de control que incluyeron elementos de control–; sin embargo, no fueron evidentes, debido a la naturaleza de los ejercicios, aspectos tales como la generalización –respecto a la definición de variables generalizadoras–, la recursión –especificación fun-

- The first two practices are proposed as exercises to learn the tool and understand basic computing concepts related to the structured programming paradigm, without explicit reference to those issues, but with the aim of internalizing them empirically from practice.
- In the third practice the teams work to propose a solution to a problem from the initial indications that the teachers introduce. This practice seeks to assert management concepts such as variables and repeat structures. The teams, at this stage, should monitor their performance through the use of the template shown in **Table 3**.

This template was then adapted (see **Table 4**) when the teams had already developed initial activities with the tool.

Process

The practices proposed for the groups increase in complexity, with the teams covering and practicing computational concepts without explicitly considering this as a formal learning activity (**Figure 2**).

Subsequent practices, where the format of **Table 4** is applied, are presented in greater detail (**Figure 3**). In Figure 4 the practices where teams create their own methods

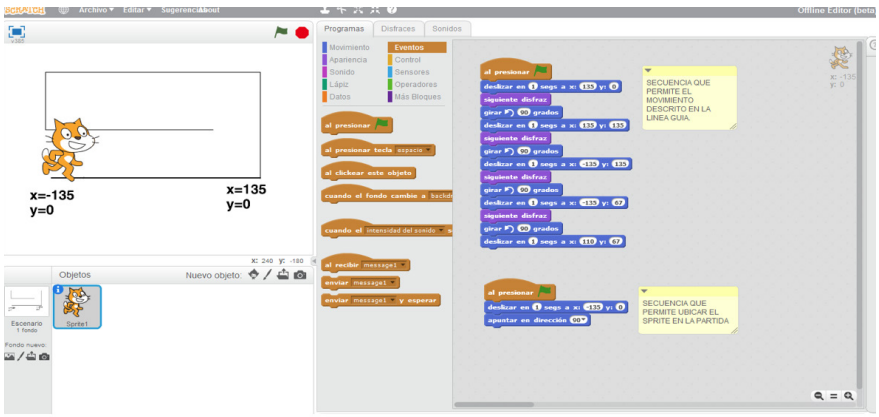

Figure 2. Example of work done at the initial stage / Ejemplo de trabajo realizado en la etapa inicial

will be seen, from which a further process of abstraction will be evident.

III. Results

In the first study cases, the groups show, mostly, the use of abstraction mechanisms such as separation of interests –evidenced by the organization of different algorithms for each event of the same character– and the creation of composite elements from simple elements –the organization of control elements that include control elements. However, due to the nature of the exercises, there was no evidence of aspects such as generalization –regarding the definition of generalizing variables–, recursion –a functional specification that uses the same definition–, and concealment of information –without global variables between the characters scene. Encapsulation occurred naturally, but was due to the ability of the software tool to achieve it (as a character program). Later, having the opportunity to work with the second group for a longer period of time, teams showed evidence of a significant improvement in their application of abstraction mechanisms.

Also, with the use of the *Dr. Scratch* tool, the process of abstraction became more evident, allowing them to assess the following characteristics related to computational thinking: parallelism, logical thinking, flow control, user interactivity, information representation, abstraction and synchronization.

Table 5 shows three possible activities that are designed to enable work teams to gradually achieve better performance in their use of computational thinking skills, as evaluated with *Dr. Scratch*.

cional que use su misma definición– y el ocultamiento de la información –sin variables globales entre personajes de la escena–. El encapsulamiento se dio en forma natural, pero fue debido a la capacidad de la herramienta de software para lograrlo (puesto que se programa un personaje). Posteriormente, al tener la posibilidad de trabajar con el segundo grupo durante un espacio de tiempo mayor, los equipos lograr evidenciar un mejora significativa en su aplicación de mecanismos de abstracción.

Asimismo, el uso de la herramienta *Dr. Scratch* hizo más evidente el proceso de abstracción, al permitir evaluar las siguientes características relacionadas con el pensamiento computacional: paralelismo, pensamiento lógico, control de flujo, interactividad con el usuario, representación de la información, abstracción y sincronización.

En la **Tabla 5** se presentan tres posibles actividades que se han diseñado de manera progresiva para que los equipos de trabajo logren un mejor desempeño en el uso de las competencias del pensamiento computacional, al ser evaluadas con el *Dr. Scratch*.

IV. Conclusiones

Al iniciar esta exploración con un grupo de estudiantes en su entorno real es importante concluir que a los niños no es necesario llenarlos con una gran cantidad de información relacionada con la herramienta o con los conceptos computacionales, puesto que esto tiende a aburrirles, mientras que entrar en la práctica estimula en ellos el trabajo en equipo y motiva su pensamiento abstracto. Esta experiencia práctica evidencia que los niños son capaces de llegar a soluciones, así cuenten con pocos elementos o herramientas; sin embargo, el proceso de abstracción se hace más fluido siempre y cuando ellos tengan claridad sobre el uso de las herramientas y su interacción.

Se debe proponer material educativo que haya sido probado en un entorno real de aprendizaje, apoyándose en los docentes del área de tecnología e informática, que facilite la comprensión de conceptos computacionales, atendiendo las recomen-

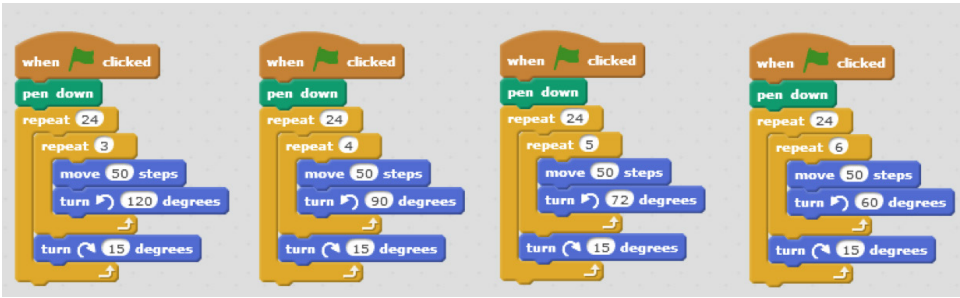

Figure 3. Example of work in the intermediate stage / Ejemplo de trabajo en la etapa intermedia

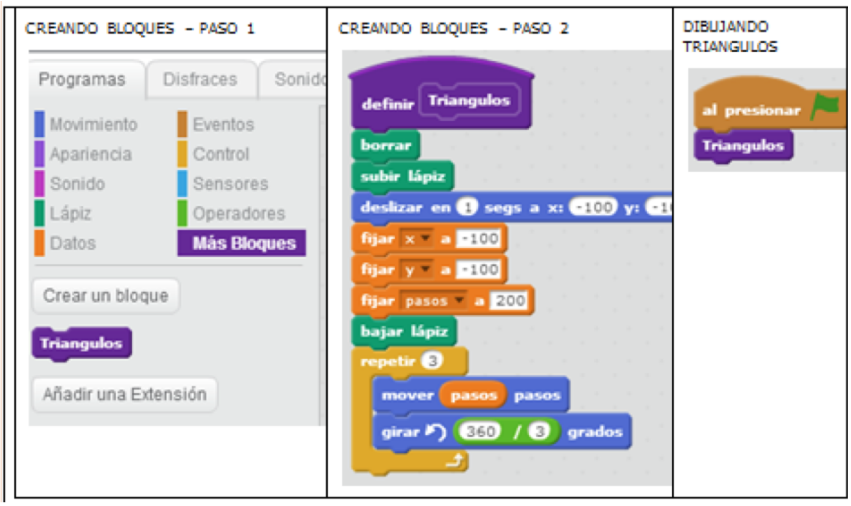

Figure 4. Detailed example of activity, for creating procedures /
Ejemplo de actividad más elaborada, por la creación de procedimientos

daciones estructuradas a partir de la caracterización de los mecanismos de abstracción que permitan definir un orden, y un conjunto de estrategias que delineen caminos para el proceso de enseñanza y aprendizaje.

Como trabajo futuro se espera, a partir de éste primer acercamiento, el estudio sistemático de un conjunto de mecanismos, siguiendo un enfoque híbrido, desde la perspectiva de cuantificación de los resultados. Este tipo de ejercicio se ha replicado, a modo de prueba, en entornos universitarios; ha sido interesante ver cómo los estudiantes reaccionan frente a este tipo de trabajo que utiliza un entorno grafico, que para ellos representa algo totalmente diferente a las herramientas formales de programación, puesto que tienen las fichas disponibles y su funcionamiento en el mismo entorno, lo que deja más espacio para el desarrollo de habilidades de pensamiento relacionadas con la creatividad y la abstracción. Esta idea de trabajo a nivel universitario será desarrollada utilizando la herramienta *Snap!*, en la cual se utilizan conceptos de programación más avanzados. S&T

IV. Conclusions

When starting this exploration with a group of students in their real environment, it is important to determine that children do not need to be filled with a lot of information related to the tool or computational concepts, since this tends to bore them, whereas putting it into practice encourages their teamwork and motivates abstract thought. This practical experience shows that children are able to reach solutions, having only a few elements or tools; however, the process of abstraction becomes more fluid as long as they are clear about the use of the tools and their interaction.

It is recommended to use educational material that has been tested in a real learning environment, based on teachers in the area of technology and information, to facilitate the understanding of computer concepts, following the recommendations structured from the characterization of the mechanisms of abstraction to define an order, and a set of strategies that outline ways of teaching and learning.

Future work is expected, based on this first approach, to make a systematic study of a set of mechanisms, following a hybrid approach, from the perspective of quantification of the results. This type of exercise has been replicated, on a trial basis, in university environments; it has been interesting to see how the students react to this type of job using a graphical environment, which for them represents something totally different from formal programming tools, since they have the cards available and they operate in the same environment, which leaves more space for the development of thinking skills related to creativity and abstraction. This idea of college-level work will be developed using the *Snap!* tool, in which more advanced programming concepts are used. S&T

Computational thinking concept / *Concepto pensamiento computacional*	Basic Scratch	Project name / *Nombre del proyecto*	Description / *Descripción*	Characters / *Personajes*
Abstraction and decomposition of problems / *Abstracción y descomposición de problemas*	Over a script and a sprite / *Más de un script y más de un sprite*	La lleva / *La lleva*	Two characters move about randomly stage (a ball and cat), when the ball touches the cat the cat costume changes and moves to the lower right corner of the screen and remains so for 10 seconds. To start moving again. / *Dos personajes se mueven sobre el escenario de manera aleatoria (un balón y el gato), cuando el balón toque al gato el gato cambia de disfraz y se mueve hacia la esquina inferior derecha de la pantalla y se queda así por 10 segundos. Para iniciar a moverse de nuevo.*	Cat / Gato *Ball / Pelota*
Parallelism / Paralelismo	Two scripts a green flag / *Dos scripts una bandera verde*			
Logical thinking / *Pensamiento Lógico*	Yes use / *Uso de Si*			
Synchronization / Sincronización	*Wait* / Esperar			
Flow control / *Control de Flujo*	block sequence / *Secuencia de bloques*			
User interactivity / *Interactividad con el usuario*	Green flag / *Bandera Verde*			
Data Representation / *Representación de Datos*	Modifies the sprites properties / *Modifica las propiedades de los sprites*			

Table 5.1 Proposed activities to be evaluated with Dr. Scratch / Actividades propuestas para ser evaluadas con Dr. Scratch

Computational thinking concept / *Concepto pensamiento computacional*	Basic Block Definition	Project name / *Nombre del proyecto*	Description / *Descripción*	Characters / *Personajes*
Abstraction and decomposition of problems / *Abstracción y descomposición de problemas*	Over a script and a sprite / *Definición de Bloques*	A que no me tocas	The cat starts its operation by clicking on it, then the stone begins to move from one side to another and the idea is that the cat cannot stop touching the stone holding down the key up and so avoid the rock. It should show a variable to count the times that the cat has been hit by the stone, this game must be programmed for 10 seconds so you can record who dodges better the stone. (Draw two bars of color to also count the times that the cat is toouching) / *El gato inicia su funcionamiento al hacer clic sobre él, en ese momento la piedra comienza a moverse de un lado a otro y la idea es que el gato no se deje tocar por la piedra manteniendo presionada la tecla hacia arriba y así esquivar la roca. Se debe mostrar una variable que cuente las veces que el gato ha sido golpeado por la piedra, este juego se debe programar por 10 segundos para que se pueda registrar quien esquiva mejor la piedra. (Dibujar dos barras de color para que también se cuenten las veces que el gato las toca)*	Cat / *Gato* Stone / *Piedra*
Parallelism / *Paralelismo*	Two scripts (by pressing key), pressing two scripts own sprite / *Dos scripts (presionando tecla), dos scripts presionando el propio sprite*			
Logical thinking / *Pensamiento Lógico*	Yes or if not use / *Uso de si y sino*			
Synchronization / *Sincronización*	When you receive a message (stop everything, stop the program, stop programs sprite) / *Cuando reciba un mensaje (detenga todo, detenga programa, detenga programas del sprite)*			
Flow control / *Control de Flujo*	Repeat forever / *Repetir por siempre*			
User interactivity / *Interactividad con el usuario*	Press key, press sprite, question and wait (mouse) / *Presionar tecla, presionar sprite, pregunta y esperar (ratón)*			
Data Representation / *Representación de Datos*	Operations and variables / *Modifica las propiedades de los sprites*			

Table 5.2 Proposed activities to be evaluated with Dr. Scratch / Actividades propuestas para ser evaluadas con Dr. Scratch

Computational thinking concept / *Concepto pensamiento computacional*	Basic Scratch	Project name / *Nombre del proyecto*	Description / *Descripción*	Characters / *Personajes*
Abstraction and decomposition of problems / *Abstracción y descomposición de problemas*	Using clones	A que no me tocas La historia	It is needed to identify a story in which various characters are used, they must move from one scene to another, at the time that meets a procedure. / *Es necesario identificar una historia en la que se utilicen diversos personajes, los que deben pasar de una escena a otra, en el momento que cumpla con algún procedimiento.*	Defined by work teams according to their story / *Definidos por los equipos de trabajo según su historia*
Parallelism / *Paralelismo*	Two scripts where a message is received a clone, two scripts when a condition is met, two scripts are created when the background change			
Logical thinking / *Pensamiento Lógico*	Logical operations			
Synchronization / Sincronización	Wait until, when it changes the background wait			
Flow control / *Control de Flujo*	Repear Unitil / Repetir hasta / Esperar hasta, cuando cambie el fondo esperar			
User interactivity / *Interactividad con el usuario*	When > , audio and video / Cuando > , audio y video			
Data Representation / *Representación de Datos*	List Operations / Operaciones con listas			

Table 5.3 Proposed activities to be evaluated with Dr. Scratch / Actividades propuestas para ser evaluadas con Dr. Scratch

References / Referencias

Bautista V., J. Granados, J., Ortiz, D., & Reinoso, L. (2009). *Un estudio de caso de las representaciones mentales que seis niños de 8 y 9 años del colegio San Bartolomé La Merced tienen del pensamiento* [thesis]. Universidad Javeriana: Bogotá, Colombia. Available at: http://www.javeriana.edu.co/biblos/tesis/educacion/tesis46.pdf

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* [online]. Retrieved from: http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Bunge, M. (2000) . *La ciencia, su método y su filosofía*. México DF: Siglo XXI.

Serna, E. (2011). La abstracción como componente crítico de la formación en ciencias computacionales. *Revista Avances en Sistemas e Informática, 8(3), 79-83.*

Feierherd, G. E., Depetris, B. O., & Jerez, M. (2001). *Una evaluación sobre la incorporación temprana de algorítmica y programación en el ingreso a Informática* [paper in VII Congreso Argentino de Ciencias de la Computación]. Retrieved from: http://hdl.handle.net/10915/23229

Hurtado, J. A., Collazos, C. A., Cruz, S. T., & Rojas, O. E. (2012). Child programming: una estrategia de aprendizaje y construcción de software basada en la lúdica, la colaboración y la agilidad. *Revista Universitaria RUTIC, 1*(1), 9-14. Available at: http://ciclope.unicauca.edu.co/rutic/index.php/rutic/article/view/4

Mönig, J. (2015). Snap: Build your own blocks [online]. Retrieved from: https://github.com/jmoenig/Snap--Build-Your-Own-Blocks

Johnson-Laird, P. N. (1986). *Mental models*. Cambridge, MA: Harvard University Press.

Jonker, C. M., Van-Riemsdijk, M. B., & *Verme*ulen, B. (2011). Shared mental models. In M. De-Vos, N. Fornara, J. V. Pitt, & G. Vouros (Eds.), *Lecture Notes in Computer Science: Vol. 6541. Coordination, organizations, institutions, and norms in agent systems VI* (pp. 132–151). Berlin-Heidelberg, Germany: Springer Verlag.

League, C. (2000). Lambda calculi: A guide for computer scientists by Chris Hankin. ACM SIGACT News, 31(1), 8-13. http://doi.org/10.1145/346048.568490

M. Letsky, N. Warner, S. Fiore, & C. A. P Smith (Eds.). (2008). *Macrocognition in teams: Theories and methodologies*. Hampshire, UK: Ashgate Publishing.

Ministerio de las Tecnologías de la Información y las Telecomunicaciones [MINTIC]. (2001). *Computadores para educar: Historia*. Retrieved from: http://www.computadoresparaeducar.gov.co/PaginaWeb/index.php/es/nosotros-2/historia

Moreira, M. A. (1999). *Modelos mentais. Investigações em Ensino de Ciências, 4*(1), 193-232.

Moroni, N. (2001). *Entornos para el aprendizaje de la programación* [paper in III Workshop de Investigadores en Ciencias de la Computación]. Available at: http://hdl.handle.net/10915/21707

Ministerio de Educación Nacional [MEN]. (2008). *Orientaciones generales para la educación en tecnología* [Serie Guías No. 30]. Bogotá, Colombia: MEN. Available at: http://www.mineducacion.gov.co/1621/articles-160915_archivo_pdf.pdf

Page-Jones, M. (1992). Comparing techniques by means of encapsulation and connascence. *Communications of the ACM, 35*(9), 147-151. http://doi.org/10.1145/130994.131004

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM, 15*(12), 1053-1058. http://doi.org/10.1145/361598.361623

Rendón, M. A., & Parra, P., (2008). *Estudio sobre las representaciones (modelos mentales) que tienen los maestros en formación respecto a la cognición* [online]. Retrieved from: http://ayura.udea.edu.co:8080/jspui/bitstream/123456789/676/1/2008%20-%20Estudio%20sobre%20las%20representaciones%20%28modelos%20mentales%29%20que%20tiene%20los%20maestros%20en%20formacion%20respecto%20a%20la%20cognici%C3%B3n.pdf

Rodriguez, M. L., Marrero, J., & Moreira, M. A. (2001). La teoría de los modelos mentales de johnson-laird y sus principios: una aplicación con modelos mentales de célula en estudiantes del curso de orientación universitaria. *Investigações em Ensino de Ciências, 6*(3), 243-268.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131-164.

Staats, A. W. (1996). *Behavior and personality: Psychological behaviorism*. New York, NY: Springer.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2014, Janaury). Computational thinking benefits society [online] *Social Issues in Computing* [blog]. Retrieved from: http://socialissues.cs.toronto.edu/2014/01/computational-thinking/

CURRICULUM VITAE

René Fabián Zúñiga Muñoz Candidate to Master in Computer, and Telematics Specialist from the Universidad del Cauca (Colombia); Educational Management Specialist from the Universidad Católica de Manizales (Colombia); Systems Engineer from the Universidad Cooperativa de Colombia; member of IDIS research group at the Universidad del Cauca. He currently serves as Coordinator at the Institución Educativa Técnico Industrial of Popayan (Colombia) and is chair professor at the Universidad del Cauca; his research works are in software engineering and computer education / Candidato a Magister en Computación y Especialista en Telemática de la Universidad del Cauca (Colombia); Especialista en Gerencia Educativa de la Universidad Católica de Manizales (Colombia); Ingeniero de Sistemas de la Universidad Cooperativa de Colombia; integrante del grupo de Investigación IDIS de la Universidad del Cauca. Actualmente se desempeña como Coordinador en la Institución Educativa Técnico Industrial de Popayán (Colombia) y como docente de cátedra de la Universidad del Cauca; las líneas de investigación sobre las cuales trabaja son: ingeniería de software e informática educativa.

Julio Ariel Hurtado Alegría Electronics and Telecommunications Engineer and Specialist in Networks and Telematics Services (Universidad del Cauca, Colombia); Specialist in Processes Software Development (Universidad de San Buenaventura, Colombia) and Ph.D. in Computer Sciences from the Universidad de Chile. Researcher in computer science, especially in the areas of software engineering and software engineering processes. Teacher dedicated to the formation of systems engineers and related professions in methodologies, techniques and practices for software development by integrating three perspectives: technical, management and quality. IDIS member of the group of research at the Universidad del Cauca / Ingeniero en Electrónica y Telecomunicaciones y Especialista en Redes y Servicios Telemáticos de la Universidad del Cauca (Colombia); Especialista en Procesos para el Desarrollo de Software de la Universidad San Buenaventura de Cali (Colombia); y Doctor en Ciencias de la Computación de la Universidad de Chile. Investigador de la Ciencia de la Computación, especialmente en las áreas de ingeniería de software e ingeniería del proceso software. Docente dedicado a la formación de ingenieros de sistemas y profesiones afines en metodologías, técnicas y prácticas para el desarrollo de software, integrando tres perspectivas: técnica, gestión y calidad. Integrante del grupo de Investigación IDIS de la Universidad del Cauca.

Patricia Paderewsky Rodríguez. Ph.D. In Computing from the Universidad de Granada (Spain) and full time professor since 1990, at the same university. Her research interests include software architecture based on cooperative agent systems, HCI and educational video games. She belongs to the GEDES research group at the Universidad de Granada. She has participated as an evaluator and speaker at various international events / Doctora en Computación de la Universidad de Granada (España) y docente a tiempo completo de la misma universidad, desde 1990. Sus campos de investigación son: arquitectura de software, sistemas basados en agentes cooperativos, HCI y videojuegos educativos. Pertenece al grupo de investigación GEDES de la Universidad de Granada. Ha participado como evaluadora y ponente en diversos eventos internacionales.