

# Estudiando el modelo ChildProgramming-G para encontrar elementos que permitan desarrollar un sistema de memoria transactiva

Studying the ChildProgramming-G model to find elements that allow the development of a transactive memory system

Julio A. Hurtado Alegría<sup>1</sup>, Viviana Y. Gómez Calvache<sup>1</sup>, Andrea C. Zambrano Lasso<sup>1</sup>

<sup>1</sup> Universidad del Cauca, Colombia

ahurtado@unicauca.edu.co , vygomez@unicauca.edu.co , aczambrano@unicauca.edu.co

**RESUMEN.** Partiendo de los grandes retos y dificultades que trae consigo la industria del software hoy en día, y la manera abrupta en que cambia de una generación a otra, es necesario que los desarrolladores de software adquieran ciertas habilidades y conocimientos desde edades tempranas, por tal motivo se llevó a cabo dos experiencias con niños en edad escolar. En la primera experiencia se aplicó el modelo de desarrollo para enseñar a los niños a programar denominado ChildProgramming, en la segunda experiencia se aplicó el modelo ChildProgramming-G, el cual es una extensión que incluye técnicas de gamificación. Durante las experiencias se pudo observar que los modelos desarrollan un conocimiento homogéneo en los niños, además de que se presentan algunos inconvenientes a la hora de la planificación y asignación de las tareas dentro de los equipos de trabajo. Por tal motivo y para dar solución a los inconvenientes presentados se pretende incorporar al modelo ChildProgramming-G un conjunto de prácticas basadas en la formación de expertos temáticos que desarrollen un sistema de memoria transactiva (SMT), garantizando la diferenciación de conocimiento, además de la coordinación y la credibilidad dentro del equipo, que conlleven a un mayor rendimiento. En este artículo se describen las experiencias llevadas a cabo y los conceptos relevantes acerca de la memoria transactiva, que serán tomados en cuenta para la realización de las prácticas, las cuales hacen parte de un trabajo de investigación en curso.

**ABSTRACT.** Based on the great challenges and difficulties that come with the software industry today, and the abrupt way it changes from one generation to another, it is necessary that software developers acquire certain skills and knowledge from an early age, for that reason He held two experiences with school-age children. In the first experiment the development model was used to teach children how to program called ChildProgramming, in the second experiment the ChildProgramming-G model, which is an extension that includes gamification techniques applied. During the experiments it was observed that a homogeneous models develop knowledge in children, plus some disadvantages when planning and allocation of tasks within the teams are presented. For that reason and to solve the problems presented is intended to incorporate the model ChildProgramming-G a set of solutions based on the formation of thematic experts to develop a system transactive memory (SMT), ensuring differentiation of knowledge as well as practical the coordination and credibility within the team, that lead to higher performance. This article describes the experiments carried out and relevant about the memory transactivate concepts that will be taken into account for the implementation of practices, which are part of an ongoing research are described.

**PALABRAS CLAVE:** ChildProgramming, ChildProgramming-G, Sistemas de memoria transactiva, Conocimiento compartido.

**KEYWORDS:** ChildProgramming, ChildProgramming-G, Transactive memory system, Shared knowledge.

## 1. Introducción

En la actualidad, la educación debe estar al ritmo de las exigencias de una sociedad cambiante, que cada día busca descubrir alternativas tecnológicas, propuestas desde los mismos planes de estudio de las instituciones educativas. Sin embargo, son pocas las instituciones que han implantado estos métodos de enseñanza, debido a que no reconocen la importancia de fortalecer el desarrollo cognitivo de los estudiantes, mediante la exploración de herramientas tecnológicas que permitan adquirir nuevos conocimientos y experiencias como parte integral de su formación (Hurtado et al., 2012; Flannery, 2014). Según estudios realizados, muchos estudiantes carecen de habilidades como el pensamiento crítico, el pensamiento analítico y habilidades de resolución de problemas. De acuerdo a la organización de NESTA del Reino Unido "se enfrenta a una escasez de personas relacionadas con la ciencia, la tecnología, la ingeniería y las matemáticas (STEM)" (Jobs, 2011). Teniendo en cuenta estos aspectos, se llevó a cabo una primera experiencia donde se aplicó el modelo de proceso ChildProgramming (ChP) para enseñar a los niños a programar, propuesto por Hurtado (Hurtado et al., 2012; Hurtado, 2012), aplicado para este caso a niños entre los 5 y 6 años de edad. En el transcurso de la primera experiencia, García y Collazos (2012) propusieron un modelo denominado ChildProgramming-G (ChP-G) como una extensión del modelo inicial ChP. Esta propuesta incorpora un componente lúdico mediante técnicas de gamificación como lo son las dinámicas y mecánicas de juego, involucrando la herramienta Scratch (Entorno de aprendizaje gráfico ideal para la enseñanza de la programación).

En los resultados que obtuvo García y Collazos (2012) se describe que los niños tenían inconvenientes a la hora de planear y asignar las tareas dentro de los equipos. En las experiencias llevadas a cabo para esta investigación con los modelos ChP y ChP-G se pudo observar que se presentan los inconvenientes mencionados por el autor. Además, teniendo en cuenta que estos modelos no consideran el desarrollo de experticias, podemos suponer como hipótesis que el conocimiento que se genera es homogéneo, originando una sobrecarga de información sin que se identifiquen y referencien focos de conocimiento especializado aunque en la práctica existan. Hecho por el cual se cree se presentan los inconvenientes antes nombrados. Investigaciones muestran que la coordinación y la diferenciación del conocimiento son especialmente importantes para el desarrollo de software. El desconocer la experticia que cada miembro del equipo va ganando, hace que se desaproveche la capacidad de éste para ser más efectivo. Por lo tanto, se plantea la incorporación de un sistema de memoria transactiva (SMT) al modelo ChildProgramming-G, que permita la formación de expertos temáticos dentro del equipo, lo cual hará coincidir las tareas, los conocimientos y las personas, promoviendo así una mayor efectividad del talento humano, además de promover en los niños el desarrollo de la creatividad y la lógica como habilidades enfocadas a formar personas competentes capaces de afrontar nuevos y mayores retos. Asimismo, fomentar el interés de los niños por carreras de la ciencia y la tecnología, dado que es necesario que los profesionales en esta área adquieran ciertas habilidades y conocimientos desde edades muy tempranas (Hurtado et al., 2012; Hurtado, 2012)

## 2. Marco conceptual y trabajos relacionados

### 2.1. ChildProgramming y ChildProgramming-G

ChildProgramming es un modelo de desarrollo para enseñar a los niños a programar planteado por Hurtado (Hurtado et al., 2012; Hurtado, 2012). El modelo conceptual se basó en tres principios fundamentales: la Lúdica, la Colaboración y la Agilidad (Hurtado et al., 2012). Sin embargo, el proceso inicial del modelo se desarrolló teniendo en cuenta un componente cognitivo en lugar de lúdico. Los autores realizaron un estudio exploratorio del modelo el cual fue aplicado a un grupo de estudiantes entre 8 y 10 años de edad pertenecientes a los grados cuarto y quinto de básica primaria, seleccionando las prácticas que mejor resultado obtuvieron en los equipos, en términos de comportamiento, productividad y calidad del trabajo, las cuales fueron incorporadas en el proceso ChP, el cual se basa en el marco de desarrollo Scrum, que comprende las fases de: pre-juego, juego y post-juego (ver Figura 1), definidas por ChP según los términos de los niños para su mayor entendimiento. Se desarrolló la evaluación del modelo propuesto con una actividad de desarrollo de software utilizando la herramienta Scratch, logrando obtener como resultado un incremento



tanto en la calidad de los productos como en la efectividad de los equipos. Más adelante, García y Collazos (2012) argumentando la importancia del componente lúdico dentro del aprendizaje y la enseñanza de la programación, y debido a que no había sido incorporado explícitamente en el modelo, plantearon ChildProgramming-G (ver Figura 2), una versión extendida del modelo inicial ChP, mediante técnicas de gamificación que incluye mecánicas y dinámicas de juego, además de los tipos de jugadores, esto con el objetivo principal de incrementar la motivación hacia el aprendizaje, el compromiso y el trabajo colaborativo.

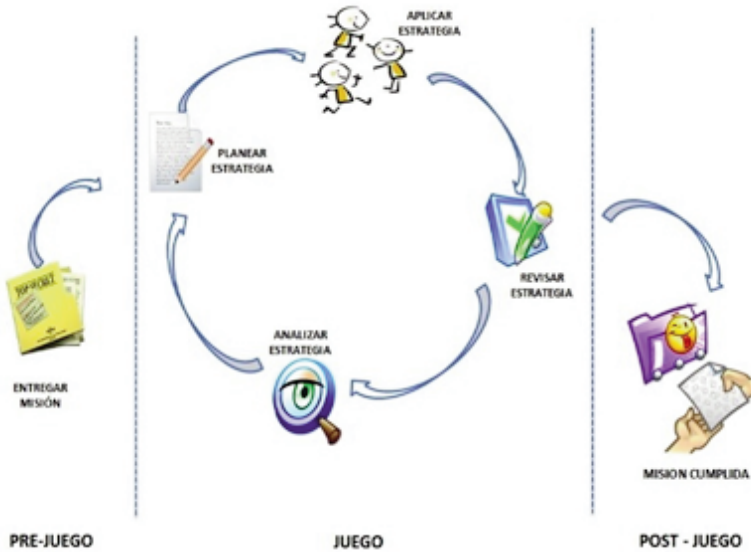


Figura 1. Ciclo de vida del proceso ChildProgramming. Fuente: Elaboración propia.

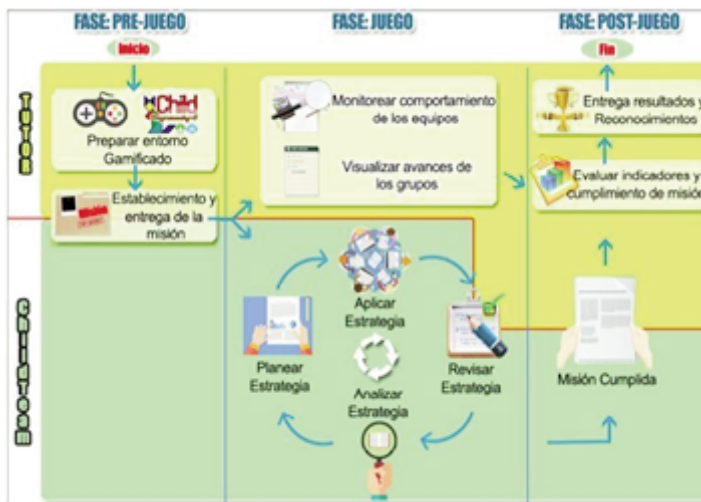


Figura 2. Ciclo de vida del proceso ChildProgramming-G. Fuente: Elaboración propia.

## 2.2. Sistema de memoria transactiva

Wegner (Sánchez-Manzanares et al., 2006) plantea que las personas se utilizan mutuamente como ayudas externas de memoria, lo cual facilita el desarrollo de “un sistema compartido para adquirir, almacenar y recuperar la información”. La teoría del sistema de memoria transactiva, indica que para que exista un intercambio efectivo del conocimiento, los miembros de este deben ser capaces de identificar quien posee que

conocimiento, es decir “quien sabe qué” (Hsu et al. 2012), esto permitirá al equipo compartir de manera efectiva el conocimiento que se encuentra distribuido entre ellos (Taglieri, 2010). Por consiguiente el sistema de memoria transactiva, reduce la carga de información de los miembros del equipo, ya que se divide el trabajo cognitivo. Dado esto, se dice que el sistema de memoria transactiva se desarrolla cuando los miembros de un equipo poseen diferentes conocimientos, son precisos o exactos en el reconocimiento de la experiencia de los demás y pueden comunicarse fácilmente para combinar sus conocimientos cuando sea necesario (Kitaygorodskaya, 2006). Una ventaja de la memoria compartida es promover el uso más eficiente de los recursos humanos, puesto que a nivel de equipo se puede conseguir un mayor rendimiento en las actividades que se estén llevando a cabo, haciendo coincidir las tareas, los conocimientos y las personas (Hsu et al. 2012).

## 2.2.1. Elementos para la formación de los sistemas de memoria transactiva

Revisando la literatura acerca de los sistemas de memoria transactiva se encontró que las investigaciones han adoptado dos enfoques para su estudio, ya sea a través de sus manifestaciones cognitivas o a través de los comportamientos, tomando estos dos enfoques como base para extraer los elementos necesarios para el desarrollo de los sistemas de memoria transactiva. A continuación se describen los elementos que hacen parte de estos enfoques.

### 2.2.1.1. Enfoque de comportamiento

Uno de los enfoques en el que se centran los sistemas de memoria transactiva es en el comportamiento de las personas, en cómo los miembros de equipos comparten la información de sus áreas de especialización. Wegner sugirió que la memoria transactiva se compone de tres comportamientos como son: la actualización de directorios, asignación de información y la coordinación de recuperación.

#### Directorios de actualización

La actualización de directorio es el proceso por medio del cual los integrantes de un equipo crean y revisan sus percepciones de “quien sabe qué” o mediante el intercambio de solicitar información a los compañeros de equipo. Este proceso consta de dos directorios los cuales se denominan “auto-directorio” y “directorio de los demás”. Por consiguiente el directorio de actualización, permite que las personas sean capaces de reconocer donde reside la experiencia dentro de la red, el experto será la persona o las personas que son percibidas por tener un mayor conocimiento sobre algún tema teniendo como base estos dos directorios (Pearsall, 2008; Palazzolo et al., 2006).

- Auto-directorio: se utiliza para mantener un registro de los propios niveles de conocimiento.
- Directorio de los demás: en este directorio las personas mantiene las percepciones de lo que saben los demás miembros de la red.

#### Asignación de la información

La asignación de información es un elemento importante debido a que este comportamiento puede señalar la identificación de expertos en el equipo, además de proporcionar apoyo informativo a los expertos. Significa que cuando las personas reciben nueva información, en lugar de recepcionar esa información en su memoria personal, esa información puede ser asignada a otra persona en el equipo que tiene una mejor y más desarrollada estructura de directorios, es decir que la información se asigna a la persona que es percibida por el equipo como el experto en el tema (Palazzolo et al., 2006).

#### Coordinación de recuperación

A través de este comportamiento las personas utilizan su directorio de actualización para acceder o solicitar información o conocimiento a áreas de especialización de los compañeros del equipo, lo que muestra una comprensión de cómo se distribuye el conocimiento dentro del equipo.



### 2.2.1.2. Enfoque cognitivo

Las investigaciones han mostrado que los sistemas de memoria transactiva también se centran en manifestaciones cognitivas en el equipo. Moreland fue uno de los primeros en proponer dimensiones cognitivas de la memoria transactiva, en su investigación determinó que los equipos que son entrenados de manera conjunta desarrollan más plenamente la memoria transactiva, esto debido a que hay una mayor conciencia de la distribución del conocimiento individual dentro del equipo. Este reconocimiento de la experiencia permite a los integrantes del equipo buscar la información en el compañero correcto y mejorar la coordinación de la experiencia, lo que implicaría un mejor rendimiento. Identificaron tres factores que representan los componentes cognitivos de los sistemas de memoria transactiva: la diferenciación de memoria, la credibilidad de tareas y la coordinación de tareas. Más adelante Lewis continuó esta línea de investigación, cambiando el nombre de estos tres factores por: especialización, credibilidad y coordinación (Pearsall, 2008; Palazzolo et al., 2006).

#### Especialización

Al igual que la diferenciación de la memoria, la especialización hace referencia a la experiencia que se encuentra distribuida dentro del equipo.

#### Credibilidad

La credibilidad se refiere a la confianza en la fiabilidad de las experticias de los integrantes del equipo.

#### Coordinación

La coordinación se refiere a la capacidad del equipo para integrar sin problemas sus esfuerzos y conocimientos cuando los requieran o los necesiten.

## 2.3. Trabajos relacionados

Los equipos de desarrollo de software que se encuentran en entornos distribuidos, enfrentan retos particulares para intercambiar sus conocimientos. Desde la perspectiva de los sistemas de memoria transactiva Chen [2] buscan explorar los comportamientos y relaciones entre los SMT, el intercambio de conocimiento y la calidad de la comunicación. Este estudio utilizó las medidas de Lewis para medir la diferenciación y la credibilidad del conocimiento, y la medida propuesta por Faraj y Sproull (Sproull, 2014) para medir la ubicación del conocimiento. En el estudio participaron 72 desarrolladores de software libre. Los resultados indicaron que los SMT contribuyeron en gran medida al desempeño de estos equipos, puesto que se evidenció que saber dónde se localiza el conocimiento, ayuda a que los desarrolladores se acerquen a los expertos cuando sea necesario y de esta manera se ayuda al aprendizaje del equipo. Por otro lado la credibilidad del conocimiento impacta de manera favorable la comunicación y el intercambio de conocimiento entre los desarrolladores de software, pues la confianza entre los miembros del equipo permite entablar una comunicación más abierta y precisa. Estas contribuciones también son válidas para equipos de software no distribuidos.

Otras investigaciones han justificado la relación positiva que existe entre el intercambio de conocimiento y el grado en que un equipo completa la tarea. Por ejemplo Faraj y Sproull (Sproull, 2014), argumentan que debido a que el desarrollo de software es un trabajo que requiere de conocimiento, uno de sus recursos más importantes es la experiencia. Aunque, la sola experiencia no garantiza que se obtendrán productos de alta calidad, al igual que otros recursos la experiencia también debe ser coordinada. La difícil coordinación, las dificultades en el intercambio e integración del conocimiento fueron factores que dificultaron los resultados de un proyecto de un estudio observacional de un equipo de desarrollo de software. Por esto, los autores proponen la coordinación de la experiencia como un elemento importante en la coordinación del trabajo en equipo. Plantean que esta coordinación se basa en procesos cognitivos, socialmente compartidos, que se

desarrollan y evolucionan para cumplir con tareas que requieren de habilidades y conocimientos, además sugieren que los integrantes del equipo deben saber diferenciar los conocimientos de los demás cuando lo necesiten. Este estudio explora la coordinación de la experiencia a través de 69 equipos pertenecientes a la división de desarrollo de aplicaciones de una empresa de alta tecnología. En el estudio se desarrollaron medidas para la coordinación de la experiencia, la coordinación administrativa, experiencia profesional, métodos de desarrollo de software, presencia de experiencia, eficacia y eficiencia en un equipo. Se concluyó que la coordinación de la experiencia afecta positivamente el rendimiento de un equipo, esto por encima de factores tradicionales como la coordinación administrativa y la presencia de la experiencia. Estos resultados apoyan los argumentos de otras investigaciones, donde se afirma que las mejoras en los procesos sociales en el desarrollo de software, es un área donde se pueden obtener grandes beneficios para aumentar el rendimiento del desarrollo de software.

Estos trabajos encuentran que el intercambio de conocimiento y la coordinación de la experiencia pueden ser una forma en la que los equipos de desarrollo de software obtendrán un mejor rendimiento en sus actividades, puesto que el desarrollo de software es un proceso donde los desarrolladores comparten e integran sus conocimientos especializados. En nuestra investigación la experticia será el elemento central con el cual se trabajara el modelo ChildProgramming-G, donde los expertos que se formen a través de las prácticas, deberán aportar sus conocimientos al equipo para que en conjunto coordinen e intercambien sus conocimientos y de esta manera planeen estrategias que los lleve a cumplir el objetivo de la tarea de una manera más organizada y precisa. A diferencia de las investigaciones anteriores, este trabajo se aplica en el ámbito educativo con niños en edad escolar.

### 3. Aplicación de los modelos de proceso ChP y ChP-G en niños de edad escolar

#### 3.1. Primera Experiencia

Esta experiencia se llevó a cabo con 25 niños entre los 5 y 6 años de edad del colegio “Instituto Nacional Mixto de Piendamó” (Cauca, Colombia) del grado primero de básica primaria. Se realizaron tres visitas al centro educativo, en las que se contó con 2 horas para realizar cada una de las actividades diseñadas para trabajar con la herramienta Scratch. Se creó un ambiente de trabajo como ChildProgramming lo sugiere: un proyector, se instaló Scratch en cada uno de los computadores a utilizar, se hizo uso de carteleras, papeles de colores y marcadores para el desarrollo de las actividades. Las dos primeras actividades fueron totalmente guiadas y desarrolladas en grupos de 2 estudiantes, en la tercera actividad se aplicó el proceso Ch-P donde se pidió a los estudiantes desarrollar un requerimiento en grupos conformados por seis estudiantes.

##### 3.1.1. Descripción de las actividades

Primera Actividad: “el gato y el ratón”, esta actividad buscó que el niño ubicará de forma correcta en un escenario un fondo tropical y dos objetos que son el gato y el ratón, posteriormente el estudiante debía dar movimiento y sonido al gato dando la ilusión de que este quiere cazar al ratón. Sin embargo, el ratón no debería dejarse cazar y por esto se incluyeron eventos y sensores para que este último detectara cuando el gato está cerca y se pudiera esconder. Esta actividad busca que el estudiante haga un reconocimiento general de la herramienta así como de los principales elementos con los que esta cuenta.

Segunda Actividad: “El vuelo de un pájaro”, esta actividad buscó darle vida a un pájaro, permitiendo que éste volase para alcanzar una fruta que sería su alimento, haciendo uso de los mismos elementos de programación que la primera actividad. Sin embargo, aumentó la complejidad al hacer que el pájaro se moviera en dos direcciones y la animación del vuelo del pájaro se vea más real al trabajar con los disfraces del objeto. Con esta actividad se buscó reforzar los conocimientos aprendidos en la primera actividad, aparte de incluir nuevos conocimientos logrando una mayor familiarización con la herramienta por parte del estudiante.

Tercera Actividad: en esta actividad se aplicó el proceso Ch-P donde se presentó a los estudiantes un requerimiento considerado como “la misión”, con características similares a las desarrolladas en las anteriores actividades. Se buscó que en grupos conformados por 6 estudiantes que a su vez trabajan en parejas den solución al problema planteado, bajo la adopción de algunas prácticas ágiles de desarrollo como lo son la definición de roles y el trabajo en pareja (ver Figura 3.). Esta actividad estuvo enfocada en introducir a los estudiantes en los principios de la estrategia de aprendizaje de Ch-P (pre-juego, juego, post-juego).



Figura 3. Equipo de trabajo aplicando ChildProgramming en un Proyecto con Scratch. Fuente: Elaboración propia.

### 3.2. Segunda Experiencia

Esta experiencia se llevó a cabo con 30 niños entre los 9 y 12 años de edad del colegio “Colombo Francés de Popayán” (Cauca, Colombia) del grado quinto de básica primaria (ver Figura 4). Se realizó una introducción a la herramienta scratch y a los conceptos básicos de programación, enseguida se desarrollaron dos actividades en las que se aplicó el modelo ChildProgramming-G. Esta experiencia se realizó con el propósito de observar y entender mejor el modelo, además de ver si los niños eran capaces de reconocer el conocimiento que sus compañeros de equipo iban adquiriendo en el transcurso del tiempo. Asimismo, poder deducir de qué manera y con qué elementos se puede desarrollar un sistema de memoria transactiva dentro del modelo aplicado.



Figura 4. Equipo de trabajo aplicando ChProgramming-G en un Proyecto con Scratch. Fuente: Elaboración propia.

### 3.2.1. Desarrollo de la experiencia

- Se dio a los niños una introducción en términos de fácil comprensión para ellos acerca de la programación y sus conceptos básicos como son las estructuras de control de secuencia, de decisión y de iteración. Además, Se explicó el funcionamiento de los principales componentes de scratch: área de bloques (movimiento, apariencia, sonido, lápiz, control, sensores, operadores y variables), área de programa, área de ejecución y área de objetos. esta introducción se llevó a cabo en 4 sesiones de clase, que corresponden a 6 horas de tiempo.
- Se realizaron dos actividades, en cada una de las cuales se presentó a los estudiantes un requerimiento o misión al que debían dar solución aplicando la dinámica del modelo ChildProgramming-G. La misión se explicó y entregó a cada equipo, contando con dos sesiones de clase para completar cada una de ellas, lo que corresponde a 6 horas de tiempo.

## 4. Resultados

En la primera experiencia Se observó que los niños realizaron de manera completa las actividades dirigidas en el tiempo designado, unos con mayor dificultad que otros, esto tal vez se presentó por la diferencia en su experiencia en el manejo de computadores. Se destacó el interés y la motivación de los niños por el aprendizaje de la herramienta Scratch. Sin embargo, algunos niños eran muy individualistas y esto dificultó un poco las actividades en equipo. La última actividad donde se aplicó el modelo ChP no contó con la participación de todo el curso, por lo tanto se pudo conformar 3 equipos. En ésta última actividad sobresalió un niño de un equipo el cual se convirtió en un protagonista del conocimiento, puesto que se observó que cada vez que los niños del equipo presentaban inconvenientes con la tarea delegada, acudían siempre a él para que los guiara y así poder completar dicha tarea. Aunque él no tenía la respuesta a todas sus dudas, la credibilidad del estudiante por parte del equipo fue alta. Adicionalmente se presentaron dificultades a la hora de planear y asignar las tareas dentro del equipo.

En la segunda experiencia realizada con el modelo ChP-G se pudo confirmar a partir de observaciones que los niños presentaban las mismas dificultades a la hora de planear y asignar las tareas dentro del equipo. En esta experiencia se inculco a los niños, el compartir el conocimiento dentro del equipo, ya que si algún integrante del equipo no se sentía seguro o no lograba realizar la tarea que se le había asignado, este podía acudir y solicitar ayuda a la persona del equipo que el creyera que tenía los conocimientos necesarios para dar solución al problema. Esto se realizó con el objetivo de que los niños empezaran a reconocer las fortalezas y conocimientos de sus compañeros de equipo

## 5. Propuesta

De acuerdo a la literatura y a este primer caso exploratorio nace ChP-GTM, la cual agrega un conjunto de prácticas orientadas a la formación de expertos temáticos, que permitan construir un sistema de memoria transactiva dentro del modelo ChildProgramming-G, fortaleciendo al equipo en términos de credibilidad, especialización y coordinación. Este sistema será un modelo descriptivo que representa las transacciones que se requieren para que dos o más personas puedan de forma cooperativa almacenar, recuperar y comunicar información y conocimiento. Este modelo se piensa como un modelo en red totalmente conectada en donde cada conocimiento se cultiva desde la especialización, inicia con una topología centralizada por cada foco de especialidad, pero que se entreteje hasta formar un clické de tamaño mayor o igual a  $n/2 + 1$ , donde n es el tamaño del equipo (ver Figura 5). La idea es lograr en el tiempo de nuevo un conocimiento homogéneo como lo propone ChildProgramming, pero en un equipo que reconoce las experticias individuales de los miembros en el equipo.





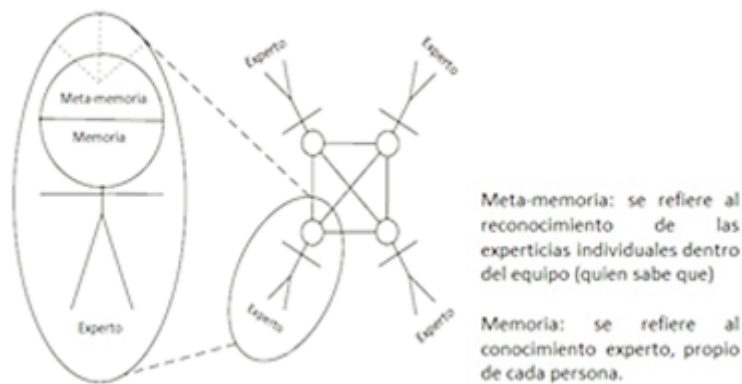


Figura 5. Red de conocimiento con un equipo de cuatro personas. Fuente: Elaboración propia.

## 6. Conclusiones

Al aplicar el modelo ChildProgramming y ChildProgramming-G, se logró observar a través de los resultados obtenidos, que existió un foco de conocimiento especializado, pero sin ser identificado por los miembros del equipo bajo el tema de su dominio sino como aquel que todo lo sabía, sin que esto fuera cierto. Esto de alguna manera permite observar que el proceso ChP al igual que ChP-G no facilitan el desarrollo de experticias, que permitan llevar a cabo una mejor coordinación y planificación de las tareas dentro de los equipos, por lo que puede ser una de las causas de este problema, teniendo en cuenta la importancia de estos dos conceptos en los equipos de desarrollo de software. A partir de estos hallazgos se decidió llevar a cabo una investigación relacionada con los sistemas de memoria transactiva para mejorar el modelo ChP-G. Para tal fin se está diseñando un estudio de caso con niños de edad escolar entre los 9 y los 10 años del grado quinto de la Institución Educativa Técnico Industrial de la ciudad de Popayán, con el objetivo de realizar un análisis de la memoria transactiva en Childprogramming-G con el fin de identificar qué elementos podemos tomar como base para el desarrollo de las prácticas que ayudarían a la conformación de un sistema de memoria transactiva efectivo.

Cómo citar este artículo / How to cite this paper

Hurtado, J. A.; Gómez, V. Y.; Zambrano, A. C. (2017). Estudiando el modelo ChildProgramming-G para encontrar elementos que permitan desarrollar un sistema de memoria transactiva. *Campus Virtuales*, 6(1), 99-108. ([www.revistacampusvirtuales.es](http://www.revistacampusvirtuales.es))

## Referencias

- Flannery, L. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Informática y Educación*, 72, 145-157
- García, A. A.; Collazos, A. (2012). ChildProgramming-G: Extendiendo ChildProgramming con Técnicas de Gamificación.
- Hurtado, J. A. (2012). Proceso ChildProgramming: Un Modelo de Desarrollo de Software Para Niños.
- Hurtado, J. A.; Collazos, C. A.; Cruz, S. T.; Rojas, O. E. (2012). ChildProgramming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Ludica, la Colaboración y la Agilidad. *Rev. Univ. RUTIC*, 1(1), 9-14.
- Jobs, C. S. (2011). STEM Sector Employer Skills Survey Report. ([http://www.ctstemjobs.org/resources/CT\\_STEM\\_Employer\\_skills\\_survey\\_summary\\_report.pdf](http://www.ctstemjobs.org/resources/CT_STEM_Employer_skills_survey_summary_report.pdf))
- Hsu, J. S. -C.; Shih, S. -P.; Chiang, J. C.; Liu, J. Y. -C. (2012). "The impact of transactive memory systems on IS development teams' coordinate communication and performance. *International Journal Project Manager*, 30(3), 329-340.
- Kitaygorodskaya, N. (2006). Measurement of Team Knowledge: Transactive Memory System and Team Mental Models Menu. *PrevMenu*.
- Palazzolo, E. T.; a Serb, D.; She, Y.; Su, C.; Contractor, N. S. (2006). Coevolution of Communication and Knowledge Networks in

- Transactive Memory Systems: Using Computational Models for Theoretical Development. 16, 223–250.
- Pearsall, M. J. (2008). A model of transactive memory development in teams. ProQuest Diss. Theses.
- Sánchez-Manzanares, M.; Rico, R.; Gil, F.; Martín, R. S. (2006). Memoria transactiva en equipos de toma de decisiones: Implicaciones para la efectividad de equipo. *Psicothema*, 18(4), 750-756.
- Sproull, L.; Smith, R. H.; Park, C. (2014). Coordinating Expertise in Software Development Teams. *Manage. Sci.*, (December), 1554–1568.
- Taglieri, B. L. (2010). Transactive Memory Systems in Undergraduate Information Systems Student Project Groups. *Structure*, (May).

