

ACTIVIDADES DE AULA CON SCRATCH QUE FAVORECEN EL USO DEL
PENSAMIENTO ALGORÍTMICO.

EL CASO DEL GRADO 3° EN EL INSA

TRABAJO DE GRADO

ESTUDIANTE:

JUAN CARLOS LÓPEZ GARCÍA

TUTORA:

SANDRA PEÑA

UNIVERSIDAD ICESI

CREA

MAESTRÍA EN EDUCACIÓN

CALI

2014

Tabla de contenido

1. Introducción.....	4
2. Hipótesis de investigación	5
2.1. Pregunta de investigación.....	6
2.2. Justificación.....	6
2.3. Objetivos.....	7
3. Marco teórico	8
3.1. Pensamiento computacional	9
3.2. Pensamiento Algorítmico.....	11
3.3. Aprendizaje Basado en Problemas (ABP)	13
3.4. Scratch	14
3.5. Interacciones.....	17
4. Metodología.....	18
4.1. Contexto empírico de la investigación	18
4.2. Descripción de los sujetos de la investigación	20
4.3. Instrumentos utilizados en la recolección de la información	21
4.3.1. Instrumento de medición del pensamiento computacional.....	21
4.3.3. Rejilla ICOT de ISTE para observar clases.....	29
4.3.4. Imágenes de plantillas de análisis de problemas	30
4.3.5. Grabaciones en video de algunas de las clases	30
4.4. Tipo de investigación y procedimiento.....	31
4.4.1. Caracterización de la investigación.....	31
4.4.2. Procedimiento de recolección de la información.....	31
5. Resultados	34
5.1. Datos cuantitativos.....	34
5.2. Datos cualitativos	37
5.2.1. Conceptos de pensamiento algorítmico e interacciones relacionados con Analizar un problema.....	41

5.2.2. Conceptos de pensamiento algorítmico e interacciones relacionados con Diseñar una solución.....	45
5.2.3. Conceptos de pensamiento algorítmico e interacciones relacionados con Implementar una solución.....	47
5.2.4. Conceptos de pensamiento algorítmico e interacciones relacionados con Depurar una solución.....	50
6. Análisis y discusión de resultados	52
6.1 Análisis cuantitativo	52
6.2 Análisis cualitativo.....	55
6.2.1 Conceptos del Pensamiento Algorítmico.....	57
6.2.2 Interacciones docente-estudiante(s).....	62
6.2.3 Interacciones entre pares	64
7. Conclusiones y recomendaciones	65
8. Bibliografía.....	70
ANEXOS	76

1. Introducción

La programación de computadores se viene implementando en ambientes escolares desde la década de 1970; sin embargo, el pensamiento computacional es una noción que aparece hace pocos años. En el año 2006 Jannette Wing acuñó el término para referirse a la actividad mental de formular y resolver problemas que admitan soluciones computacionales. Esta nueva noción, unida a nuevos lenguajes de programación, tales como Scratch, hace parte de un movimiento que aboga actualmente porque cada estudiante, de cada escuela, tenga la oportunidad de aprender a programar.

En el Instituto Nuestra Señora de la Asunción (INSA) de Cali, Colombia, se viene implementando desde el año 2004 una propuesta para enseñar a programar a los estudiantes de educación básica primaria. En el año lectivo 2011 se reportó que varios docentes de la institución observaron mejoras en los desempeños académicos de estudiantes de secundaria, los cuales habían aprendido a programar en la primaria. Al respecto, en el año 2012, se llevó a cabo en el INSA una investigación que concluyó que “los resultados del análisis de tareas muestran en detalle la forma cómo el uso del entorno gráfico de programación Scratch, junto con las actividades educativas propuestas en el aula, promueven el desarrollo del pensamiento computacional, la adquisición de conocimiento conceptual académico y habilidades de planificación cognitiva” (Taborda & Medina, 2012; p 17).

Posteriormente, a finales de 2012 se presentó al comité de investigaciones de la Universidad Icesi otro proyecto, para ser ejecutado durante el año lectivo 2013, cuyo título fue: “Impacto de Scratch en el desarrollo del Pensamiento Computacional”. El objetivo general de ese proyecto consistió en determinar si elaborar programas de computador con el entorno gráfico de programación Scratch favorece el desarrollo del pensamiento computacional en los estudiantes de 2° y 3° grado de básica primaria del INSA. Para lograr este fin, en dicha

investigación se propuso inicialmente identificar y definir operacionalmente las habilidades cognitivas involucradas en el pensamiento computacional, así como diseñar y validar instrumentos para evaluar su desarrollo. El método involucró la recolección de información tanto cualitativa como cuantitativa del desempeño de los estudiantes ante situaciones de resolución de problemas y del desarrollo de actividades de clase con y sin el uso del computador.

En el presente trabajo de grado de Maestría se utilizó el conjunto de datos recolectado en el marco de esta investigación de la Universidad Icesi con el fin de profundizar en una de las dimensiones del pensamiento computacional, como lo es el pensamiento algorítmico. Por tanto, el objetivo de este trabajo de grado consistió en establecer la relación que pudiera existir entre el uso, por parte de los estudiantes de grado 3° del INSA, de conceptos del pensamiento algorítmico y las actividades de aula fundadas en el uso de Aprendizaje Basado en Problemas (ABP) como estrategia didáctica, que a su vez, utiliza el entorno de programación Scratch como herramienta.

Por su parte, uno de los objetivos de la Maestría en Educación de la Universidad Icesi establece que “se propiciará el uso de las TIC como apoyo a los procesos pedagógicos y se promoverá una actitud de investigación a través de la intervención en el aula de clase” (CREA, 2012; p 6). En consecuencia, la presente investigación es absolutamente pertinente para la Maestría, por cuanto indaga lo que sucede en el aula de clase cuando se utiliza una herramienta de las Tecnologías de la Información y las Comunicaciones (TIC), como Scratch, con el fin de que los estudiantes desarrollen habilidades de pensamiento.

2. Hipótesis de investigación

A partir del trabajo empírico que se ha venido realizando en programación de computadores, desde el año 2004 en el Instituto Nuestra Señora de la Asunción

(INSA), se plantea la siguiente hipótesis: Enseñar a programar computadores favorece el uso de conceptos del pensamiento algorítmico de los estudiantes de grado 3° de básica primaria del INSA.

2.1. Pregunta de investigación

El objeto de investigación en el presente trabajo de grado se expresó mediante la siguiente pregunta:

¿Cuáles son las características de las actividades de aula bajo el modelo de resolución de problemas, del entorno de programación Scratch y de las interacciones en el aula, que favorecen el uso de conceptos del pensamiento algorítmico por parte de los estudiantes de grado 3° de educación básica primaria del INSA?

2.2. Justificación

En la actualidad, Scratch es el entorno gráfico de programación de computadores más ampliamente utilizado por docentes de todo el mundo. Tanto Scratch, como la definición de pensamiento computacional son relativamente recientes, razón por la cual no se encontraron muchas investigaciones disponibles que relacionen este entorno de programación con el desarrollo del pensamiento computacional y, más concretamente, con el desarrollo del pensamiento algorítmico. Por ejemplo, en el programa del congreso sobre Scratch “Connecting Worlds”, llevado a cabo en Barcelona, España, en julio de 2013, solo hay dos ponencias con investigaciones que relacionan Scratch con el pensamiento computacional y, solo una de ellas, realizada en un entorno escolar (Citilab, 2013).

Por lo anterior, la importancia de esta investigación radica en que busca aportar luces sobre las características de la intervención docente con Scratch en el INSA

que más influyen en el uso de conceptos del pensamiento algorítmico por parte de los estudiantes de grado 3°. Se espera que los resultados de esta investigación ayuden a afinar las propuestas de uso de entornos de programación de computadores en procesos educativos. Además, que contribuyan a formular cualificaciones docentes más pertinentes en el uso de programación de computadores en educación escolar como medio para ayudar a los estudiantes a desarrollar su pensamiento algorítmico. Asociado a este último propósito, este trabajo de grado promoverá la reflexión en torno a ¿cómo implementar estrategias didácticas más efectivas de Aprendizaje Basado en Problemas (ABP)?

2.3. Objetivos

El objetivo general de esta investigación consiste en establecer la relación entre el uso de conceptos del pensamiento algorítmico y la intervención educativa fundada en el uso del Aprendizaje Basado en Problemas (ABP), como estrategia didáctica, en clases de informática en las cuales se usa como herramienta el entorno de programación Scratch.

Como objetivos específicos, se formulan los siguientes cuatro:

1. Caracterizar el uso del Aprendizaje Basado en Problemas (ABP) como estrategia didáctica en el INSA.
2. Identificar qué elementos de la intervención educativa en INSA podrían tener relación con el uso del pensamiento algorítmico.
3. Identificar los elementos de Scratch que inciden en el uso de conceptos del Pensamiento Algorítmico.
4. Establecer la relación de los elementos identificados en los puntos 2 y 3 con el desarrollo del Pensamiento Algorítmico.

3. Marco teórico

La programación de computadores en educación escolar tuvo su primer auge en la década de 1980, cuando los computadores personales empezaron a llegar a las escuelas. Esto permitió que se extendiera el uso de Logo, un entorno de programación diseñado en 1967 por un grupo de investigadores del MIT liderado por Seymour Papert. Este lenguaje se difundió rápidamente entre los educadores de todo el mundo que contaban con computadores en sus escuelas y buena parte de su éxito se explica porque venía acompañado de la teoría constructivista del aprendizaje.

En la década de 1990 y la mitad de la primera década del 2000, ese interés por la programación de computadores en la educación escolar decayó. Según López (2014), decayó en buena parte debido a: (1) las equivocaciones que cometían los estudiantes al escribir las instrucciones, lo que causaba errores de sintaxis en la ejecución de los programas; (2) la interfaz poco atractiva de Logo; (3) la falta de una comunidad global de docentes que trabajaban con Logo; y, (4) la enseñanza de las suites de oficina en las clases de informática desplazó a la enseñanza de la programación.

No obstante, a partir del año 2005, el interés renació. Por una parte, se diseñaron varios entornos de programación totalmente gráficos cuyas instrucciones funcionan mediante bloques que encajan unos con otros (Scratch, Alice, Kodu, AppInventor, etc). Estos entornos tienen una curva de aprendizaje muy empinada; en muy corto tiempo y con muy poca instrucción los estudiantes pueden utilizarlos con éxito para elaborar sus propios programas.

Por otra parte, a partir del año 2010, surgió un movimiento que aboga porque cada estudiante, en cada escuela, tenga la oportunidad de aprender a programar (code.org, 2013). El sitio code.org, que lidera el movimiento, recibió el apoyo de educadores, políticos, científicos, ingenieros, artistas, etc. En últimas, este movimiento busca que la programación esté presente en todos los currículos

escolares y se apoya en una serie de nuevos entornos gráficos de programación de computadores y especialmente diseñados para su uso en la educación primaria y secundaria. Tal ha sido el “boom” de la programación, que el Departamento de Educación (2013) del gobierno británico estableció como obligatoria, a partir de septiembre de 2014, la enseñanza de programación de computadores en los grados primero a once en todas las escuelas subvencionadas de Inglaterra.

Esta introducción, además de contextualizar el papel de la programación de computadores en la educación escolar actual, define la postura pedagógica que enmarca lo observado en la investigación. Este trabajo se aborda desde la teoría sociocultural de Vigotsky; pues según Ivic (1994), esta proporciona un marco conceptual ideal para investigaciones en las que se utilizan las TIC en la enseñanza: “¿qué instrumento podría ser más adecuado y útil para investigar las repercusiones de estos nuevos instrumentos culturales en el hombre, que una teoría como la de Vigotsky, que pone precisamente en el centro de sus preocupaciones la función de los instrumentos de la cultura en el desarrollo psicológico histórico y ontogenético?” (Ivic, 1994; p 9).

3.1. Pensamiento computacional

El pensamiento computacional es un concepto acuñado por Jeannette Wing en un artículo publicado en el año 2006 en la revista “Communications of the ACM”. Este pensamiento implica “la solución de problemas, el diseño de sistemas y la comprensión de la conducta humana, haciendo uso de conceptos fundamentales de la informática. El pensamiento computacional incluye una serie de herramientas mentales que reflejan la amplitud del campo de la informática” (Wing, 2006; p 33).

En torno a este nuevo concepto, la Sociedad Internacional para la Tecnología en Educación (ISTE) y la Asociación de Docentes en Ciencias de la Computación de los Estados Unidos (CSTA) se unieron con líderes de la educación superior, de la

industria y de la educación escolar para generar una definición operativa del pensamiento computacional que pudieran utilizar los docentes de educación escolar. En esta tarea, ISTE y CSTA encuestaron a casi 700 docentes de ciencias de la computación, investigadores y profesionales en ejercicio con el fin de recopilar sus respuestas sobre varias definiciones operativas de pensamiento computacional. La siguiente definición fue la que obtuvo el mayor apoyo:

[El pensamiento computacional es] “un proceso de solución de problemas que incluye, pero no se limita a, las siguientes dimensiones: a) Formular problemas de manera que permitan usar computadores y otras herramientas para solucionarlos; b) Organizar datos de manera lógica y analizarlos; c) Representar datos mediante abstracciones, como modelos y simulaciones; d) Automatizar soluciones mediante pensamiento algorítmico [una serie de pasos ordenados]; e) Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos más eficiente y efectiva; y f) Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos” (ISTE, 2011; p 1).

Por otra parte, el grupo desarrollador de Scratch también propuso un marco de referencia para el pensamiento computacional, que incluye tres dimensiones:

“[1] conceptos computacionales (los conceptos de los que deben ocuparse los diseñadores a medida que programan; tales como iteración, secuencia, bucles, paralelismo, eventos, condiciones, operadores, datos); [2] prácticas computacionales (las prácticas que los diseñadores desarrollan a medida que se ocupan de los conceptos; tales como, trabajar iterativa e incrementalmente, reutilizar y mezclar, abstraer y modularizar, probar y depurar proyectos o remezclas basadas en el trabajo de otros); y, [3] perspectivas computacionales (las perspectivas que los diseñadores construyen sobre el mundo que los rodea y sobre ellos mismos, tales como expresar, conectar, cuestionar)” (Brennan & Resnick, 2012; p 1).

Dado que “actualmente hay consenso general respecto a que el foco de la educación escolar no debe estar en los contenidos temáticos sino en el desarrollo de habilidades, inteligencias y/o competencias básicas” (López, 2012; p 67), el pensamiento computacional debería hacer parte del currículo escolar. Principalmente, porque éste pensamiento contribuye a desarrollar la habilidad para

solucionar problemas, habilidad que aparece en todos los conjuntos de habilidades indispensables para el siglo XXI.

3.2. Pensamiento Algorítmico

Tal como se observa en la definición operativa de ISTE y CSTA, el pensamiento algorítmico es una de las dimensiones del pensamiento computacional. Habría que empezar por definir qué es un algoritmo y una definición comúnmente aceptada es: “método para resolver un problema que consiste en instrucciones exactamente definidas” (Futschek, 2006; p 160). Otra definición de algoritmo, específicamente en el campo de la computación, dice que este es “una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible” (López, 2009; p 21).

El pensamiento algorítmico es considerado como uno de los conceptos clave de la tecnología de información que permite a las personas tener fluidez en el uso de las Tecnologías de la Información y la Comunicación (TIC). En tal sentido, los miembros de la comisión NRC lo operacionalizan de la siguiente forma: “...conceptos generales de pensamiento algorítmico, incluyen descomposición funcional, repetición (iteración y/o recursión), organización de datos básicos (registro, matriz, lista), generalización y parametrización, algoritmo vs programa, diseño de arriba hacia abajo y refinamiento” (National Research Council, 1999; p 31).

El pensamiento algorítmico es, de alguna manera, un conjunto de habilidades que están conectadas a la construcción y comprensión de algoritmos. Según Futschek (2006), este pensamiento incluye las siguientes capacidades: a) analizar problemas dados; b) especificar un problema de manera precisa; c) encontrar las acciones básicas que son adecuadas para resolver el problema dado; d) construir

un algoritmo correcto para resolver un problema determinado, utilizando las acciones básicas; e) pensar en todos los posibles casos tanto especiales como normales de un problema; y, f) mejorar la eficiencia de un algoritmo. “El pensamiento algorítmico posee un elemento creativo fuerte: la construcción de nuevos algoritmos que resuelvan problemas dados. Si alguien quiere hacer esto, necesita pensar algorítmicamente” (Futschek, 2006; p 160). Vale la pena aclarar que el pensamiento algorítmico no es un componente simple del pensamiento computacional, es una dimensión compleja que se entrelaza con otros de los componentes. De hecho, las primeras tres capacidades que Futschek (2006) asigna al pensamiento algorítmico, se podrían enmarcar claramente en la dimensión “formular problemas” de la definición operativa de pensamiento computacional propuesta por ISTE (2011).

La propuesta de programación de computadores implementada en el INSA se enfocó en la solución de problemas a partir del año 2005. Para tal efecto, se utiliza una plantilla para analizar problemas que atiende los cuatro primeros puntos propuestos por Futschek (2006). Desde ese entonces, hasta hoy, esta plantilla ha tenido como finalidad contribuir a que los estudiantes desarrollen tanto la habilidad para solucionar problemas, como su pensamiento algorítmico. A partir del enunciado de un problema (en diferentes dominios), se pide al estudiante que identifique cinco elementos: formulación del problema, datos disponibles, restricciones, resultados esperados y procesos necesarios (ver Anexo 3). Este análisis, previo al trabajo con el entorno de programación Scratch, implica una actividad cognitiva que involucra planificación, formulación de hipótesis, abstracción, comprensión lingüística; es decir, las mismas habilidades identificadas en los estudios sobre pensamiento computacional (Wing, 2006; ISTE, 2011).

3.3. Aprendizaje Basado en Problemas (ABP)

El Aprendizaje Basado en Problemas (PBL, por su sigla en inglés), es una estrategia “centrada en el estudiante, en la que este aprende a través de la experiencia de resolución de problemas” (Triantafyllou & Timcenk, 2013; p 2). La estrategia fue propuesta a finales de la década de 1960 por Howard Barrows y sus colegas en la escuela de medicina de la Universidad McMaster de Ontario, Canadá.

La estrategia de ABP fue diseñada con varios objetivos; por ejemplo, ayudar a los estudiantes a desarrollar “conocimientos flexibles, habilidades para resolver problemas eficazmente, aprendizaje auto-dirigido, habilidades de colaboración eficaces y motivación intrínseca” (Hmelo-Silver, 2004; p 235). En un ambiente de aprendizaje en el cual se implementa el ABP, el papel del docente cambia respecto a entornos educativos tradicionales. En éste, el docente hace las veces de tutor y facilitador de los aprendizajes mediante instrucción, apoyo, guía y acompañamiento a los estudiantes en el proceso de aprendizaje. Por lo tanto, “el papel del docente es el de guiar y desafiar el proceso de aprendizaje en lugar de únicamente proporcionar conocimientos” (Triantafyllou & Timcenk, 2013; p 3).

Cuando se combina un enfoque socio-cultural del aprendizaje con estrategias de PBL, los estudiantes deben ser “considerados como agentes activos que participan en la construcción de conocimiento social, con actividades construccionistas, en las que construyen conceptos en diferentes áreas a través del desarrollo de aplicaciones informáticas” (Triantafyllou & Timcenk, 2013; p 3).

Existen varias estrategias para solucionar problemas. Pero en INSA desde el año 2004 se ha venido utilizando en las clases de informática la metodología formulada por Polya (1957) en su libro “How Solve It”. Esta metodología tiene cuatro pasos que coinciden con la estrategia para elaborar programas de computador. En la gráfica 1 se aprecia a la izquierda, en azul, los pasos propuestos por Polya y a la derecha, en amarillo, el ciclo de programación de

computadores que siguen los estudiantes de INSA.



Gráfica 1: Paralelo entre las etapas planteadas por Polya y el modelo que se sigue en INSA para resolver problemas con Scratch (López, 2013b).

Según Polya (1957), cuando se resuelven problemas, intervienen cuatro operaciones mentales: 1. Entender el problema 2. Trazar un plan 3. Ejecutar el plan (resolver) 4. Revisar.

3.4. Scratch

Scratch es un entorno gráfico de programación de computadores disponible desde el año 2007. Su desarrollo estuvo a cargo del grupo “Lifelong Kindergarten” del Laboratorio de Medios del MIT, bajo la dirección y liderazgo del Dr. Mitchel Resnick. Según sus diseñadores, Scratch es “un entorno de programación que permite a niños y jóvenes crear sus propias historias interactivas, juegos y simulaciones y, a continuación, compartir esas creaciones en una comunidad en línea con otros jóvenes programadores de todo el mundo” (Brennan & Resnick, 2012; p 1). Con este entorno gráfico de programación, millones de estudiantes de todo el mundo “están creando una gama amplia de proyectos que incluyen historias animadas, programas de noticias en línea, reseñas de libros, tarjetas de felicitación, videos musicales, proyectos de ciencia, tutoriales, simulaciones y proyectos de arte y música conducidos mediante sensores” (Resnick, 2010; p 1).

Entre los nuevos entornos gráficos de programación, Scratch tomó la delantera

por encima de otros lenguajes de programación dirigidos a ambientes escolares, tales como Alice, Kodu, App Inventor, etc. En el sitio Web <http://scratch.mit.edu>, la comunidad de usuarios ha compartido en línea, desde el año 2007, 5.563.960 proyectos (cifra a Mayo 20, 2014).

Un objetivo clave de este entorno consiste en iniciar en la programación a los que no tienen experiencia previa en este campo. “Dicho objetivo está presente en muchos aspectos del diseño Scratch [...] tales como la elección de un lenguaje de bloques visuales, el diseño de la interfaz de usuario de una sola ventana y un número mínimo de comandos” (Resnick, 2010; p 3). Adicionalmente, el sistema de bloques visuales, que encajan unos con otros, “ayuda a eliminar los errores de sintaxis, lo que permite a los usuarios centrarse en los problemas que están tratando de resolver, en lugar de luchar para conseguir que sus programas compilen” (Resnick, 2010; p 14).

Según Resnick (2010), Scratch se basa en las ideas construccionistas de Logo y Etoys. Además, asegura que aprender a programar ofrece importantes beneficios. En particular, “programar apoya el ‘pensamiento computacional’, que ayuda a las personas a aprender estrategias importantes de solución de problemas y de diseño (tales como, modularización y diseño iterativo) que conducen a dominios externos a la programación” (Resnick et al, 2009; p 62).

Entre las razones por las cuales se debe aprender a programar, Resnick (2013) argumenta las siguientes:

“en el proceso de aprender a programar, las personas aprenden muchas otras cosas. No están simplemente aprendiendo a programar, están programando para aprender; pues además de comprender ideas matemáticas y computacionales, tales como variables y condicionales, simultáneamente están aprendiendo estrategias para solucionar problemas, diseñar proyectos y comunicar ideas. Esas habilidades son útiles no solo para los científicos de la computación sino para todas las personas sin distinción de edad, proveniencia, intereses u ocupación” (Resnick, 2013; párrafo 5).

Por otra parte, “la programación implica la creación de representaciones externas

de sus procesos de solución de problemas, [por lo tanto] la programación le proporciona [al estudiante] oportunidades para reflexionar sobre su propio pensamiento, incluso para pensar en el pensamiento mismo” (Resnick et al, 2009; p 62).

Desde el punto de vista académico, Scratch se soporta en la teoría construccionista del aprendizaje que a su vez se inspiró en la teoría constructivista. Esta sostiene que los alumnos construyen individualmente modelos mentales para comprender el mundo que les rodea: “de acuerdo con los principios del constructivismo, los ambientes de aprendizaje deben soportar múltiples perspectivas o interpretaciones de la realidad, de la construcción de conocimientos y de las actividades basadas en experiencia, ricas en contexto” (Triantafyllou & Timcenk, 2013; p 2).

Para Papert (1986), el aprendizaje es una reconstrucción del conocimiento más que una transmisión de este. Según Papert, citado por Stager & Libow (2013), la idea de materiales manipulables puede extenderse a la idea de que el aprendizaje es más efectivo cuando, como parte de una actividad, el aprendiz tiene la experiencia de construir un producto significativo. Esta es la base del construccionismo planteado por Papert y precisamente es eso lo que sucede cuando los estudiantes elaboran programas en un entorno como Scratch, no solo construyen productos significativos (animaciones, historias interactivas o juegos), sino que tienen la oportunidad de experimentar durante el proceso de construcción. Así, para Stager & Libow (2013), el computador ofrece un material flexible que los estudiantes pueden entretelar con sus propias ideas y dominar para sus propios fines.

Para Stager & Libow (2013), el construccionismo va más allá de la teoría constructivista: aunque el aprendizaje se sucede en la cabeza del estudiante, esto ocurre más posiblemente cuando dicho estudiante está comprometido con una actividad personalmente significativa, que sucede fuera de su cabeza y que

convierte el aprendizaje en real y compartible (aquello que se construye y que se comparte puede ser un ensayo, una manualidad, una idea o un programa de computador).

3.5. Interacciones

En cuanto a las interacciones tanto docente-estudiante(s) como entre pares, estas se abordan desde la concepción sociocultural del desarrollo. Según Ivic (1994), esta concepción considera que los vínculos con los demás forman parte de la naturaleza del niño y sus vínculos sociales deben tenerse en cuenta al analizar su desarrollo y diagnosticar sus aptitudes. “El concepto de zona de desarrollo próximo ilustra precisamente este punto de vista. Tal zona se define como la diferencia (expresada en unidades de tiempo) entre las actividades del niño limitado a sus propias fuerzas y las actividades del mismo niño cuando actúa en colaboración y con la asistencia del adulto” (Ivic, 1994; p 11) o con la asistencia de un par.

En este trabajo de grado se acoge la definición de zona de desarrollo próximo que para Vigotsky no es otra cosa que:

“la distancia entre el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema, y el nivel de desarrollo potencial, determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración con otro compañero más capaz” (Vigotsky, 1995; p 133)

También se utiliza en este trabajo de grado el concepto de Andamiaje propuesto por Bruner en 1975, el cual deriva de la obra de Vigotsky y es asimilable a la zona de desarrollo próximo. Según Larkin (2002), el andamiaje es un proceso en el cual se ofrece apoyo a los estudiantes, hasta tanto estos puedan aplicar nuevas habilidades y estrategias de forma independiente. A medida que ellos comienzan a demostrar dominio en la realización de cierta tarea, la asistencia o apoyo se disminuye gradualmente con el fin de desplazar, del docente al estudiante, la

responsabilidad por el aprendizaje. Entre las interacciones se consideran tanto las que se llevan a cabo entre docente-estudiante(s), como las que se dan entre pares (estudiante-estudiante).

En cuanto al papel del docente en entornos educativos enriquecidos con TIC, el uso de herramientas como Scratch debería ir acompañado de buenas actividades de clase para que efectivamente se promueva el desarrollo del pensamiento algorítmico. Una forma de implementar en las aulas tanto la teoría sociocultural de Vigotsky como el construccionismo de Papert, consiste en formular a los estudiantes problemas en forma de preguntas que les generen retos y oportunidades de aprendizaje. Así, los estudiantes deben inventar diversas maneras de solucionar los problemas propuestos. Aunque los estudiantes pueden utilizar diversos materiales para solucionar un problema, según Stager & Libow (2013), utilizar el computador como uno de esos materiales les ofrece una experiencia más fructífera y relevante en términos de experimentación con ideas matemáticas y científicas.

4. Metodología

4.1. Contexto empírico de la investigación

Esta investigación se llevó a cabo en el Instituto Nuestra Señora de la Asunción (INSA), Institución Educativa fundada por la comunidad católica de Padres Basilianos¹ en el año de 1992. En su sitio Web, el INSA manifiesta que su principal objetivo consiste en proveer una educación de calidad a sus estudiantes y a sus familias en un contexto de justicia y paz (INSA, 2013). Este centro educativo es de carácter privado y está localizado en el barrio Andrés Sanín (Ángel del Hogar) de

¹ Ver <http://www.basilian.org/>

Cali, Colombia. Atiende una población de 720 estudiantes “provenientes de estratos socioeconómicos 1 y 2, del barrio mismo y de otros barrios aledaños: Siete de Agosto, Puerto Mallarino, La Playita y El Vivero” (INSA, 2013).

La institución cuenta con una infraestructura física compuesta por un edificio con aulas de clase, laboratorios de física, química e idiomas; por un teatro que funciona también como auditorio y por dos salas de informática. Cada sala de informática cuenta con 40 computadores y acceso a Internet. Adicionalmente, el INSA pone al servicio de la comunidad educativa un Centro Cultural con biblioteca, sala de proyección y un espacio para realizar actividades después de la jornada escolar.

En el año 2004, el Instituto de Nuestra Señora de la Asunción (INSA), con el acompañamiento de la Fundación Gabriel Piedrahita Uribe (FGPU)², empezó a implementar una propuesta para trabajar en algoritmos y programación con estudiantes de grado 5°. Según López (2013a), esta iniciativa buscaba especialmente el desarrollo, por parte de los estudiantes, de capacidades intelectuales de orden superior, incluyendo pensamiento algorítmico, solución de problemas y creatividad. En años lectivos posteriores se implementó esta propuesta con estudiantes de otros grados y al momento de realizar esta investigación se trabajaba con estudiantes de grados 2° a 6°.

En la implementación de la propuesta para trabajar en algoritmos y programación con estudiantes de básica primaria del INSA, se utilizó el entorno de programación “MicroMundos”³ hasta el año lectivo 2009. Pero en vista de que éste tiene costo, lo que dificultaba su actualización, se decidió empezar a utilizar el entorno de programación Scratch a partir del año lectivo 2010. Como ya se mencionó en el marco teórico, Scratch⁴ es un entorno de programación de computadores que se puede descargar gratuitamente de Internet y fue desarrollado por un grupo de

² Ver <http://www.eduteka.org/quienes.php3>

³ Ver <http://www.micromundos.com/>

⁴ Ver <http://scratch.mit.edu/>

investigadores del Laboratorio de Medios del MIT.

4.2. Descripción de los sujetos de la investigación

En el presente trabajo de grado participaron 66 estudiantes de los dos grupos del grado 3° de básica primaria del INSA. Ambos grupos estaban compuestos por niños (n=29) y niñas (n=37), con edades entre 7 y 10 años (7 años, n=2; 8 años, n=26; 9 años, n=32; 10 años, n=6). Estos estudiantes provenían de familias pertenecientes a estratos socioeconómicos 1 y 2, de los barrios Andrés Sanín, Siete de Agosto, Puerto Mallarino, La Playita y El Vivero, de la ciudad de Cali.

Debido a la imposibilidad para realizar esta investigación con estudiantes de solo uno de los grupos de grado 3°, se decidió trabajar con los 66 estudiantes de ambos grupos, distribuidos así: 34 estudiantes del grupo 3°A (51,5%); y, 32 estudiantes del grupo 3°B (48,5%). De la muestra se excluyeron 5 estudiantes del grupo 3°B que no completaron las respuestas en alguna de las dos oportunidades en que se aplicó del instrumento de medición del pensamiento computacional. Por lo tanto, la composición final de estudiantes fue la siguiente: 61 estudiantes en total; 34 estudiantes del grupo 3°A (55,7%); y, 27 estudiantes del grupo 3°B (44,3%).

El INSA envió a los padres de familia de todos los estudiantes de grado 3° un “formato de consentimiento informado” para que lo diligenciaran y autorizaran la participación en esta investigación. Como resultado, todos los padres de familia aceptaron que sus hijos y/o hijas participaran en la investigación, incluso que fueran grabados en video. Dichos formatos diligenciados reposan en la secretaría académica del INSA.

4.3. Instrumentos utilizados en la recolección de la información

En el presente estudio se utilizaron cinco instrumentos para la recolección de datos: 1) Instrumento de medición del pensamiento computacional; 2) Grabaciones en video de la pantalla de computador con el trabajo en Scratch de dos estudiantes por grupo; 3) Rejilla ICOT de ISTE para observar clases en las que se usan las TIC; 4) Imágenes de las plantillas de análisis de problemas diligenciadas por los estudiantes; 5) Grabaciones en video de algunas de las clases. Mediante estos cinco instrumentos se recolectó la información en la investigación “Impacto de Scratch en el desarrollo del Pensamiento Computacional” que la Universidad Icesi llevó a cabo en INSA. Dicha investigación se realizó entre febrero y octubre de 2013. En el presente trabajo de grado se utilizan los datos de esta investigación, correspondientes a los estudiantes de grado 3°. Vale la pena aclarar que el autor de este trabajo de grado participó como parte del equipo investigador en todas las fases de la investigación “Impacto de Scratch en el desarrollo del Pensamiento Computacional”.

4.3.1. Instrumento de medición del pensamiento computacional

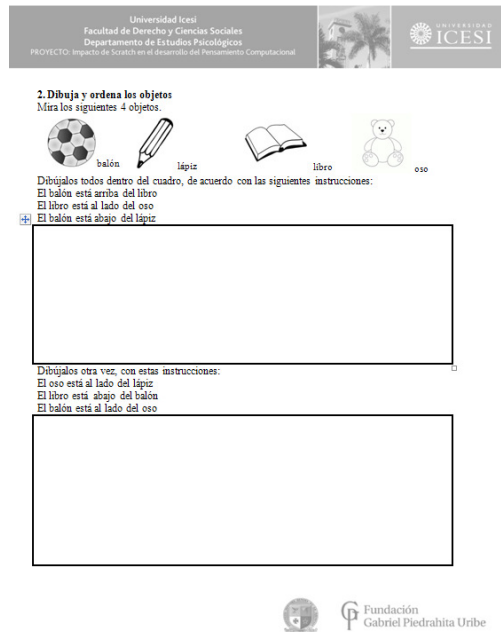
El primero es un instrumento de medición del pensamiento computacional, compuesto por 5 tareas presentadas al estudiante en forma impresa y que demanda para su solución poner en marcha las cuatro principales habilidades relacionadas con el pensamiento computacional: pensamiento lógico, abstracción y modelamiento, pensamiento algorítmico y planificación cognitiva (Ver Anexo 1). En la investigación “Impacto de Scratch en el desarrollo del Pensamiento Computacional”, se determinó tanto la validez del instrumento por medio de un análisis de tareas, como la confiabilidad del mismo a través de un análisis de concordancia inter-jueces. Este análisis de concordancia (coeficiente Kappa) dio 0.85 para la tarea 1 “conociendo el tamaño del ratón”, 0.84 para la tarea 2 “dibuja y ordena los objetos”; 0.82 para la tarea 3 “organiza los animales”; 0.71 para la

tarea 4 “gánate los puntos”; 0.76 para la tarea 5 “completa los códigos”. Adicionalmente, a dicho instrumento se le hizo un proceso de validación mediante una prueba piloto con 12 estudiantes del INSA, llevada a cabo los días 14 y 15 de febrero de 2013: 6 estudiantes de grado 2° y 6 de grado 3°.

Para el presente trabajo de grado solo se utilizaron los resultados de los estudiantes de grado 3° en las tareas 2 (dibuja y ordena los objetos) y 4 (gánate los puntos); debido a que dichas tareas son las que demandan en mayor grado el uso del pensamiento algorítmico.

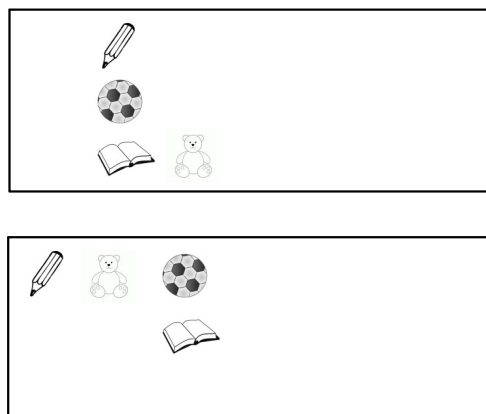
4.3.1.1. Tarea “dibuja y ordena los objetos”

La tarea “dibuja y ordena los objetos” es una adaptación de la tarea propuesta por Morra (2001) en la que plantea la operacionalización de la abstracción y el modelamiento de información. La tarea se presentó impresa sobre papel a los estudiantes. En ésta aparecían los dibujos de cuatro objetos en el siguiente orden: balón, lápiz, libro y oso. Luego se solicitó a los estudiantes que dibujaran estos cuatro objetos, dentro de dos rectángulos, siguiendo las descripciones en forma de instrucciones que aparecían en la consigna (ver gráfica 2).



Gráfica 2: Tarea “dibuja y ordena los objetos”, tomada del instrumento de medición del pensamiento computacional.

Esta tarea demandó que los estudiantes dibujaran dentro de dos rectángulos los mismos cuatro objetos (balón, lápiz, libro y oso) siguiendo dos conjuntos de instrucciones diferentes que dieran como resultado dos dibujos (modelos) con los objetos en distinto orden (ver la gráfica 3).



Gráfica 3: Una posible solución de los dos modelos propuestos en la tarea “dibuja y ordena los objetos”

Para elaborar cada modelo, los estudiantes debieron procesar las instrucciones de manera secuencial para establecer la relación espacial entre tres pares de objetos, siguiendo las tres instrucciones dadas que permitían organizar en el espacio los objetos. Con la lectura de cada instrucción, se integra nueva información al modelo; información que modifica el modelo y que debe permanecer disponible en la memoria. Las instrucciones se basan en el “Test de Descripción Espacial” propuesto por Ehrlich & Johnson-Laird (1982) que consiste en presentar la relación espacial entre varios objetos mediante tres instrucciones secuenciales de la forma: “A está debajo de B”; “B está a la izquierda de C” y “D está a la derecha de C”. Para Ehrlich & Johnson-Laird (1982), se trata de una descripción sencilla de una relación espacial de dos dimensiones (arriba-abajo, izquierda-derecha) entre cuatro objetos que puede ilustrar el uso de un modelo mental.

En esta tarea, se usó para el primer modelo un conjunto de instrucciones semicontinuas de la forma A-C; C-D; A-B; y, para el segundo modelo un conjunto discontinuo de la forma D-B; C-A; A-D. En el segundo modelo, se presentaron en la primera instrucción dos objetos, otros dos objetos diferentes en la segunda instrucción y en la tercera, se repitió un objeto de la primera instrucción y otro de la segunda. Según Ehrlich & Johnson-Laird (1982), los niños menores de 10 años tienen dificultades con las instrucciones discontinuas debido a la demanda de memoria requerida para resolverlas correctamente.

Para resolver esta tarea, los estudiantes debieron seguir mentalmente y en paralelo la secuencia de instrucciones que les permita armar el modelo resultante de manera que cupiera en el espacio asignado en forma de rectángulo (planificación, paralelismo). Luego debieron dibujar los objetos en el orden que indicaba el modelo mental. En el caso del primer modelo, si se empezaba a dibujar, de arriba-abajo, izquierda-derecha, siguiendo la primera instrucción, se corría el riesgo de llegar a la tercera instrucción sin el espacio suficiente para dibujar, dentro del rectángulo, el lápiz encima del balón. Con el segundo modelo, se esperaba establecer si los estudiantes transferían la estrategia utilizada en el

primer modelo al segundo o si mejoraban la estrategia.

Para evaluar los desempeños alcanzados por los estudiantes en esta tarea en las dos aplicaciones, se tuvieron en cuenta tanto la cantidad de instrucciones incluidas correctamente dentro del modelo, como la transferencia del procedimiento para lograr el éxito entre el primer modelo y el segundo. Por lo tanto, se plantearon tres niveles de desempeño:

Nivel de desempeño básico: El estudiante omitió información o confundió las relaciones entre objetos al elaborar cada modelo. Combinó distintos tipos de errores y correcciones en los dos modelos. Hizo correcciones, pero sin lograr elaborar el modelo correcto luego de las correcciones. Omitió objetos o los repitió en posiciones incorrectas respecto a las instrucciones. No hubo diferencia procedimental entre el primer modelo y el segundo.

Nivel de desempeño intermedio: El estudiante comprendió los componentes del modelo e hizo transferencia procedimental entre el primer modelo y el segundo. En los modelos aparecieron las relaciones entre objetos de manera correcta, con correcciones tales como borrar un elemento y dibujarlo nuevamente o dibujarlo al final, en el espacio que le quedaba disponible, pero cumpliendo con la instrucción. Repitió objetos, pero se conservaron las relaciones espaciales propuestas en las instrucciones. Hubo menos errores en el segundo modelo, respecto de los cometidos en el primer modelo.

Nivel de desempeño avanzado: El estudiante demostró que comprendió toda la información incluida en las instrucciones y elaboró correctamente los dos modelos. Pudieron presentarse correcciones tales como borrar un objeto y volverlo a dibujar en el primer modelo. Esto último indicaba que el estudiante en el primer modelo dibujó los objetos a medida que iba leyendo cada instrucción, pero en el segundo leyó primero las tres instrucciones y procedió a integrar toda la información en un solo modelo mental.

4.3.1.2. Tarea “gánate los puntos”

Esta tarea pretendió determinar la habilidad de los estudiantes para comprender y usar estructuras de control, especialmente estructuras condicionales (ver la gráfica 4). Al realizarla, los estudiantes debieron obtener un resultado basado en el cumplimiento de dos condiciones: encontrar palabras que tengan las últimas tres letras iguales a otra palabra y encontrar palabras que inician con la letra p y que terminan con a.

4. Gánate los puntos

A continuación, hay un fragmento del poema “Pastorcita” de Rafael Pombo. Si buscas en el texto puedes ganar puntos así:

- Por cada palabra que tenga las últimas tres letras iguales a otra palabra, ganas 5 puntos.
- Por cada palabra que comienza con P y que termina con a, ganas 1 punto.

Subraya las palabras con las que ganaste puntos y anota al frente el número de puntos ganados, tal como está en el ejemplo.

Pastorcita	Puntos
<u>P</u> astorcita <u>p</u> erdió sus <u>o</u> vejas	<u>1 + 5</u>
<u>¡</u> x <u>q</u> uién sabe por dónde <u>a</u> ndarán!	<u> </u>
-No te enfades, que oyeron tus <u>q</u> uejas	<u>5</u>
<u>y</u> ellas mismas bien pronto vendrán.	<u> </u>
Y no vendrán solas, que traerán sus colas,	<u> </u>
Y ovejas y colas gran fiesta darán.	<u> </u>
Pastorcita se queda dormida,	<u> </u>
Y soñando las oye balar.	<u> </u>
Se despierta y las llama enseguida,	<u> </u>
Y engañada se tiende a llorar.	<u> </u>
No llores, pastora, que niña que llora	<u> </u>
Bien pronto la oímos reír y cantar.	<u> </u>
Total Puntos:	<u> </u>



Fundación
Gabriel Piedrahita Uribe

Gráfica 4: Tarea “gánate los puntos”, tomada del instrumento de medición del pensamiento computacional.

Se presentó un bloque de información consistente en tres estrofas del poema “Pastorcita”, escrito por Rafael Pombo. El estudiante debía localizar en el texto las partes que cumplían las condiciones dadas. Al igual que la tarea “dibuja y ordena los objetos”, ésta también se presentó en formato impreso en papel. Se dieron dos

instrucciones para ganar puntos y una instrucción de asignación de puntos.

Esta tarea demandó para su resolución que el estudiante utilizara reglas condicionales y que las mantuviera activas durante la lectura del poema para que pudiera identificar las palabras que cumplieran con esas condiciones. Además, debió gestionar el sistema de puntos que se asignaban cada vez que se cumpliera con cada una de las dos condiciones dadas (ver la gráfica 5). Esta tarea también demandó del estudiante capacidad para comprender las instrucciones, en relación con el tipo de palabras que se buscaban en el texto y el mantenimiento de las condiciones durante toda la búsqueda.

Pastorcita	Puntos
<u>Pastorcita</u> <u>perdió</u> <u>sus</u> <u>ovejas</u>	<u>1+5+5+5</u>
¡y <u>quién</u> sabe por <u>dónde</u> <u>andarán</u> !	<u>5+5+5</u>
- <u>No</u> te enfades, <u>que</u> oyeron tus <u>quejas</u>	<u>5+5</u>
y <u>ellas</u> mismas <u>bien</u> <u>pronto</u> <u>vendrán</u> .	<u>5+5+5+5</u>
Y no <u>vendrán</u> <u>solas</u> , <u>que</u> <u>traerán</u> <u>sus</u> <u>colas</u> ,	<u>5+5+5+5+5+5</u>
Y <u>ovejas</u> y <u>colas</u> gran fiesta <u>darán</u> .	<u>5+5+5</u>
<u>Pastorcita</u> se queda dormida,	<u>1+5+5</u>
Y soñando <u>las</u> oye balar.	<u>5</u>
Se despierta y <u>las</u> llama enseguida,	<u>5+5</u>
Y engañada se <u>tiende</u> a llorar.	<u>5</u>
No llores, <u>pastora</u> , que niña <u>que</u> <u>llora</u>	<u>1+5+5+5</u>
<u>Bien</u> <u>pronto</u> la oímos reír y cantar.	<u>5+5</u>
Total Puntos:	<u>163</u>

Gráfica 5: Una posible solución de la tarea “gánate los puntos”

Para evaluar los desempeños alcanzados por los estudiantes en esta tarea en las aplicaciones inicial y final, se plantearon tres niveles de desempeño:

Nivel de desempeño básico: El estudiante no usó condicionales en las

respuestas dadas. No se observó una búsqueda sistemática de información. Aparentemente no comprendió las instrucciones o no identificó, dentro del texto, palabras que cumplieran con las dos condiciones dadas.

Nivel de desempeño intermedio: El estudiante demostró comprensión básica de al menos una de las dos condiciones dadas; sin embargo, no se evidenció una búsqueda sistemática de información, ni un seguimiento de la condición a lo largo de todo el texto. Por ejemplo, el estudiante señaló las palabras que cumplían solo con una de las condiciones o señaló únicamente algunas palabras dentro del texto, especialmente aquellas que se encontraban en la primera estrofa o que guardaban semejanza con las palabras señaladas en el ejemplo.

Nivel de desempeño avanzado: El estudiante demostró comprensión de las instrucciones y las siguió rigurosamente a lo largo de todo el poema. Se consideró que los estudiantes que exhiben este tipo de desempeño, podían manejar estructuras de control y mantener activa la información sobre las condiciones dadas, a lo largo de la lectura de todo el texto, de manera que podían identificar las palabras que cumplían con ellas, señalarlas y seguir las instrucciones dadas. En este grupo estaban las respuestas en las que se señalaban todas las palabras dentro del texto que cumplían con las condiciones dadas y que asignaban la puntuación correspondiente.

4.3.2. Grabaciones en video de la pantalla del computador

Uno de los mayores retos de las investigaciones que incluyen actividades de programación consiste en determinar cuáles son los procedimientos seguidos por los sujetos para llegar a un código que funcione correctamente. Esto es particularmente crítico en el caso del pensamiento algorítmico, dado que es fundamental el orden de los pasos para llegar a una solución. En otras palabras, observar proyectos elaborados en Scratch culminados y funcionales no permite

determinar cómo se llegó a ese punto. En tal sentido, capturar en video lo que sucede en la pantalla permite hacer seguimiento a las acciones y procesos que los estudiantes llevan a cabo en Scratch.

En este trabajo de grado se utilizaron las grabaciones realizadas con “CamStudio”, un programa de código abierto que se descargó gratuitamente del sitio <http://camstudio.org>. Se solicitó al docente de informática de grado tercero que seleccionara un estudiante fijo de cada grupo (3ºA y 3ºB). De esta manera se grabó lo que sucedió en la pantalla de un estudiante de cada grupo, a medida que cada uno de ellos trabajó con Scratch. Esto con el fin de poder observar si hubo progresos en desarrollo de pensamiento algorítmico a medida que transcurrían las clases. Del grupo 3ºA se grabó la pantalla del estudiante JC y del grupo 3ºB a la estudiante NG.

4.3.3. Rejilla ICOT de ISTE para observar clases

El instrumento de observación de clases de ISTE (ICOT, por su sigla en inglés), proporciona una “plataforma para el registro de observaciones de uso de las TIC en las aulas. Las casillas de verificación y los menús desplegables permiten [...] registrar fácilmente una variedad de situaciones y de características de las actividades que se presentan en una clase, cuando en ella se hace uso de las TIC” (ISTE, 2012; párrafo 1). La traducción al español de este instrumento fue realizada por Willy Figueroa Celis y Juan Carlos López García, de la Fundación Gabriel Piedrahita Uribe.

Este instrumento fue elaborado en Excel y agrega automáticamente los datos en una tabla para facilitar el análisis y presentación de información. Permite registrar información sobre el observador, el docente, el número de estudiantes, duración de la observación, el papel del docente, los tipos de uso de la tecnología y el cumplimiento de los estándares en TIC de ISTE. Aunque este instrumento permite

registrar datos cuantitativos y cualitativos, para el presente trabajo de grado solo se tuvieron en cuenta los datos cualitativos: interacciones docente-estudiantes y/o entre pares, actividades realizadas durante la clase, compromiso que demostraron los estudiantes con la actividad y recursos utilizados por parte de docentes y estudiantes (Ver Anexo 2). Todos estos datos se registraron en los seis campos de “Notas” que contiene este instrumento.

4.3.4. Imágenes de plantillas de análisis de problemas

Según López (2013b), la primera fase para solucionar problemas mediante programas de computador consiste en formularlos claramente (comprensión lingüística), especificar los resultados que se desean obtener (metas), identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles en los resultados requeridos (metas).

En el INSA, este proceso de análisis de problemas se lleva a cabo mediante el uso de una Plantilla de Análisis diseñada especialmente para promover el Aprendizaje Basado en Problemas (ver Anexo 3). Se hizo un registro fotográfico al azar de algunas plantillas de análisis diligenciadas por los estudiantes.

4.3.5. Grabaciones en video de algunas de las clases

Se realizaron grabaciones en video de algunas clases de informática de los dos grupos correspondientes al grado 3° de primaria del INSA. Estas grabaciones se realizaron entre abril y septiembre de 2013. El aporte del presente trabajo de grado consistió analizar los datos en términos tanto de pensamiento algorítmico como de los elementos que caracterizan la intervención en programación de computadores que se realiza con Scratch en el INSA: Aprendizaje Basado en

Problemas (ABP), interacciones docente-estudiante e interacciones estudiante-estudiante.

La grabación en video aportó confiabilidad porque no se limita a lo que los observadores de las clases recordaron; por el contrario, en momentos en que surgían dudas se pudo volver sobre los registros una y otra vez. Por otra parte, el hecho de contar con varios observadores le dio confiabilidad a este instrumento.

4.4. Tipo de investigación y procedimiento

4.4.1. Caracterización de la investigación

Para cumplir con los objetivos propuestos se planteó realizar una investigación de tipo descriptivo, con un diseño no experimental, sin grupo control. No se trabajó con muestras aleatorias.

4.4.2. Procedimiento de recolección de la información

Los datos cuantitativos se recogieron mediante la aplicación en dos ocasiones, a lo largo del año lectivo, del instrumento de medición del pensamiento computacional (ver Anexo 1) diseñado para la investigación “Impacto de Scratch en el desarrollo del Pensamiento Computacional”, realizada por la Universidad Icesi. En febrero 14 y 15 de 2013 se aplicó una prueba piloto de este instrumento con el fin de evaluar la pertinencia de las tareas en términos del tiempo que los niños se tomaban en responderlas y la comprensión que tuvieron sobre las consignas e instrucciones presentadas. Esto además, permitió revisar las tareas propuestas y realizar algunas modificaciones a algunas tareas.

La aplicación inicial se realizó a los estudiantes de grado 3° del INSA los días 6 y

12 de marzo de 2013. Por su parte, la aplicación final se llevó a cabo con los mismos estudiantes el día 8 de octubre de 2013. Se decidió utilizar únicamente los datos de grado 3° debido a que los estudiantes de grado 2° apenas se estaban familiarizando con la metodología para solucionar problemas y con Scratch. En ese grado los estudiantes trabajaron principalmente con el editor de pinturas, lo cual no demandó un conjunto amplio de habilidades de pensamiento algorítmico.

Por otra parte, entre abril 16 y septiembre 10 de 2013, se realizaron 27 observaciones en las clases de informática de grado 3°, utilizando la rejilla ICOT de ISTE (ver Anexo 2). Además, se hicieron 15 grabaciones en video de capturas de pantalla con el trabajo en Scratch de dos estudiantes fijos de grado 3°; uno por cada grupo. También se recolectaron imágenes de plantillas de análisis y se realizaron 27 grabaciones en video de clases de informática.

Las observaciones de clase corrieron a cargo de una investigadora auxiliar y cuatro monitores (dos en el primer semestre de 2013 y otros dos en el segundo semestre). La investigadora auxiliar es psicóloga y en ese momento estaba cursando una maestría en psicología y los monitores eran estudiantes de pregrado del programa de psicología de la Universidad Icesi. Ninguno tenía experiencia previa en programación de computadores, lo cual evitó que se presentaran posibles sesgos en las observaciones.

Con respecto a la rejilla ICOT de ISTE, se puede afirmar que resultó dispendioso diligenciarla por la cantidad de información cuantitativa que recoge, para lo cual se requiere de una persona exclusiva para esta tarea. Sin embargo, la información cualitativa del trabajo con Scratch y de las interacciones que se llevan a cabo en el aula se recolectó mediante este instrumento de forma fácil y práctica. Para una futura aplicación de este instrumento, se recomienda hacer primero un piloto que permita, desde la práctica, precisar varios de los conceptos para los cuales recoge información cuantitativa.

Las grabaciones en video de las clases de informática estaba previsto hacerlas

con un plano amplio que permitiera captar la mayor parte del aula. Esto se cambió a partir de las primeras dos observaciones y se pasó a grabar situaciones puntuales de la clase que pudieran aportar con mayor detalle al objetivo de la investigación. De estas grabaciones emergieron con fuerza las interacciones entre pares.

El primer reto en la recolección de la información consistió en familiarizarse con la cámara de video; dominar lo técnico. En ocasiones resultó más práctico hacer las grabaciones con la cámara de un teléfono celular. El segundo reto fue la observación no participante con respecto a la grabación del trabajo del profesor. Se evidenció que los estudiantes no cambiaron su comportamiento o forma de trabajo por el hecho de que los estuvieran grabando en video. Sin embargo, la preocupación por sentirse observado pudo influir en la forma de trabajo del docente.

En cuanto a las imágenes de las plantillas de análisis de problemas (ver Anexo 3) que los estudiantes diligenciaron, no se tenía planeado registrarlas. Pero, dado que constituían una fuente de información valiosa para analizar la forma como los estudiantes planifican las tareas, se decidió tomar fotografías al azar de algunas de estas plantillas diligenciadas por los estudiantes.

Por último, en la grabación de las pantallas de computador para registrar el trabajo de los estudiantes con Scratch, se presentaron varios problemas. Primero, el plan era grabar las pantallas de dos estudiantes por grupo, uno de ellos de manera fija en todas las observaciones; este estudiante debería tener un desempeño promedio en el uso de Scratch. El otro estudiante se seleccionaba al azar en cada observación. Sin embargo, luego de varias sesiones de observación se encontró que los estudiantes seleccionados por el docente eran consistentemente los mejores de cada grupo. Después de algún tiempo, los observadores ya tuvieron criterio para determinar cuáles estudiantes eran los mejores, los con desempeño promedio, y los de bajo desempeño.

Dado que la grabación de pantalla no generaba audio, el segundo problema consistió en que no quedó registrado lo que los estudiantes decían cuando trabajaban con Scratch. Como tercer problema, CamStudio no estaba instalado en todos los computadores, por lo que hacer coincidir a los estudiantes fijos con los equipos que contaban con el software de grabación, no siempre fue posible. El estudiante fijo entraba al salón y se sentaba en un computador cualquiera y abría su trabajo en Scratch, entonces había que pedirle que se cambiara a otro equipo que tuviera CamStudio, lo cual generaba malestar. También había que controlar que los estudiantes no cerraran el programa de captura de pantalla porque se perdía toda la grabación.

5. Resultados

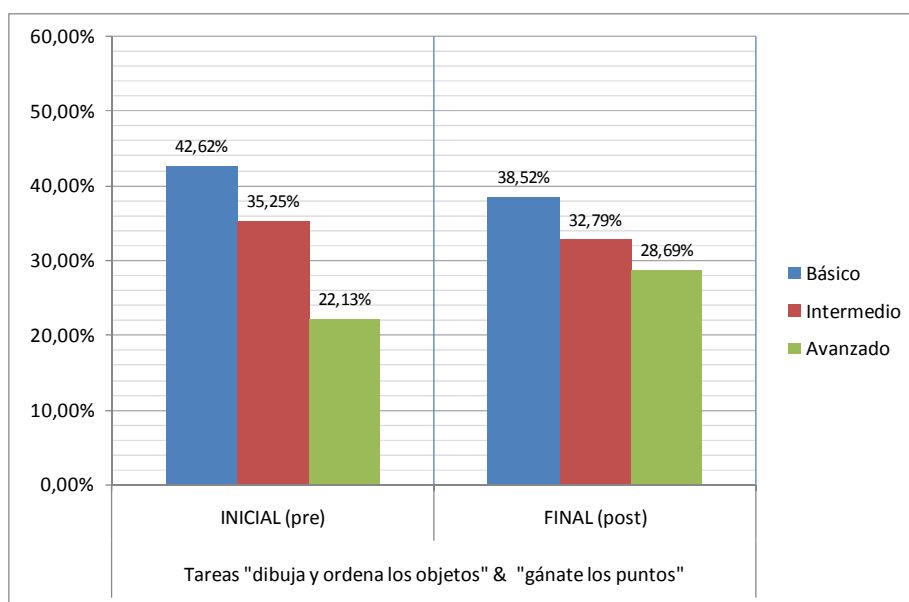
En esta investigación se obtuvo información tanto cuantitativa como cualitativa. Inicialmente se presentan los resultados cuantitativos que muestran un panorama general del desarrollo de habilidades relacionadas con pensamiento algorítmico por parte de los estudiantes participantes. Posteriormente, a partir de los datos cualitativos, se presentan resultados más detallados en los que se caracteriza la manera como se evidencia, en los desempeños de los estudiantes y en las interacciones, cada uno de los cuatro ejes identificados (analizar, diseñar, implementar y depurar).

5.1. Datos cuantitativos

A continuación se presentan los datos que se obtuvieron a partir de los desempeños de los estudiantes ante dos de las tareas que componen el instrumento de medición del pensamiento computacional (ver apartado 4.3.1). Las tareas correspondieron a: “dibuja y ordena los objetos” y “gánate los puntos”.

Como ya se mencionó, las tareas se aplicaron a los estudiantes de grado 3° del INSA, al inicio y al final del año lectivo 2013; y se utilizó la comparación entre los puntajes de las dos aplicaciones para obtener un índice del cambio en habilidades relacionadas con el pensamiento algorítmico.

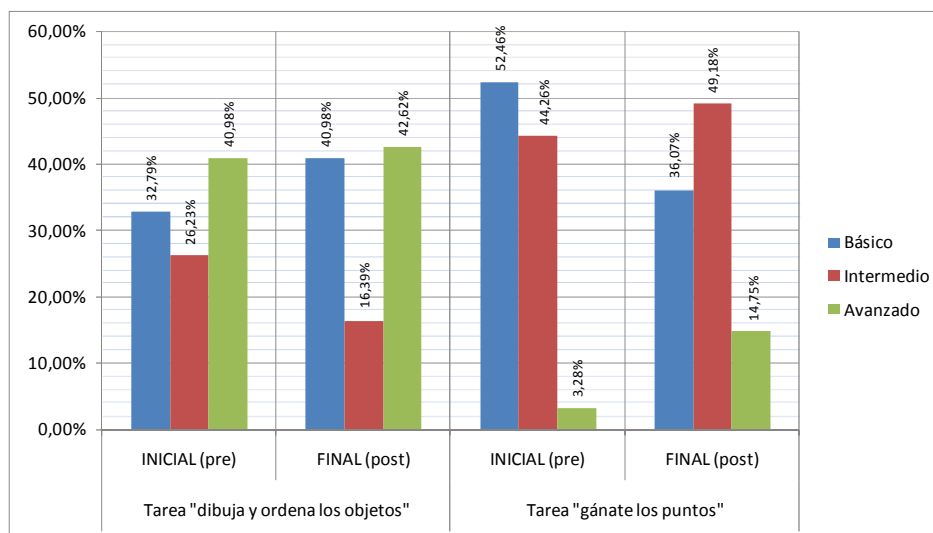
Tal como lo muestra la gráfica 6, al promediar el número de estudiantes en cada uno de los niveles de desempeño para las dos tareas, las mayores diferencias se observaron en los porcentajes de los niveles básico y avanzado. Entre la aplicación inicial y la aplicación final de la prueba se apreció una disminución de 4.10% en el nivel básico, un descenso de 2.46% en el nivel intermedio y un aumento de 6.56% en el nivel avanzado.



Gráfica 6: Número promedio de estudiantes con desempeños básico, intermedio y avanzado en las dos tareas

Al contrastar los datos de los niveles de desempeño observados en las aplicaciones inicial y final para cada una de las tareas, se encontró una diferencia de comportamiento interesante: en la tarea “Dibuja y ordena los objetos”, el mayor cambio de frecuencias ocurrió en el nivel intermedio con una disminución de 9.84%; el nivel básico aumentó 8.19% y el avanzado apenas aumentó 1.64%.

Mientras tanto, en la tarea “gánate los puntos” el mayor cambio se presentó en el nivel básico con una disminución de 16.39%. Los niveles intermedio y avanzado aumentaron 4.92% y 11.47%, respectivamente. Los datos se resumen en la siguiente gráfica:



Gráfica 7: Desempeños de los estudiantes de grado 3° en las tareas 2 y 4

Se observó variabilidad en los desempeños entre las dos tareas lo cual plantea, por una parte, modos de comprensión y de desempeño diferenciados en los estudiantes que serán descritos y analizados con mayor detalle a partir de los datos cualitativos; y, por la otra, un cuestionamiento al instrumento de medición cuantitativo. Esto último amerita un análisis metodológico del instrumento que supera los objetivos del presente trabajo de grado.

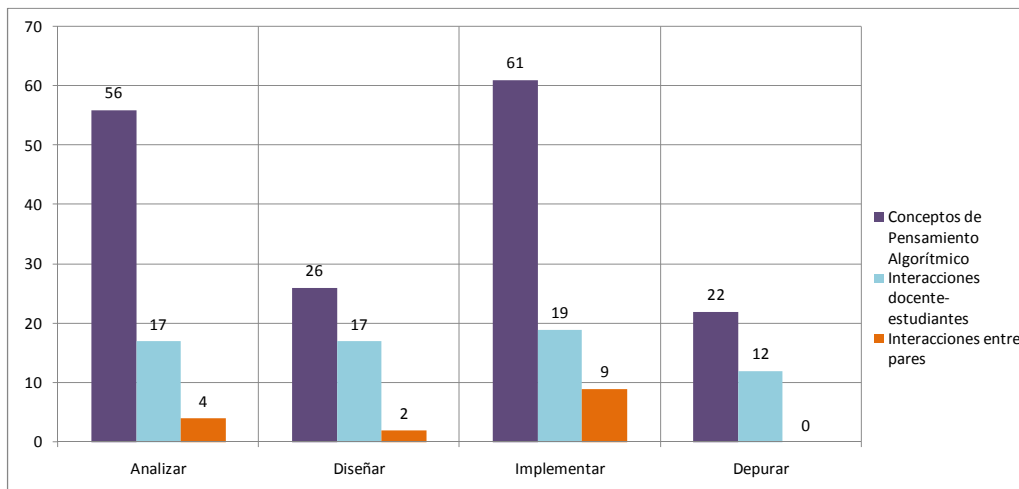
Antes de continuar con la presentación de los hallazgos, vale la pena advertir que es un mismo profesor quien imparte la asignatura en los dos grupos de grado 3°; y se hace evidente en las observaciones de clases que las actividades se realizan en condiciones y tiempos similares en ambos grupos, a lo largo del año lectivo. Además, en los videos de clases analizados, se observó que el docente actuó consistentemente en ambos grupos. Lo que hace suponer que la variabilidad en los desempeños de los niños tiene que ver más con su propia comprensión de la

tarea y en general de la actividad pedagógica, que con las condiciones de enseñanza.

5.2. Datos cualitativos

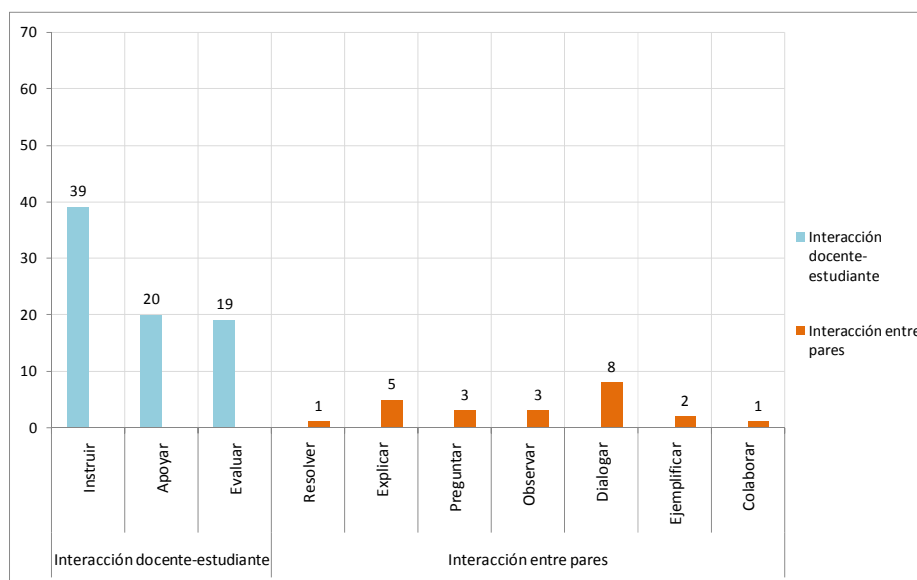
A continuación se presentan los datos que se obtuvieron mediante cuatro instrumentos de observación y registro de clases: 1) Grabaciones en video de la pantalla del computador con el trabajo en Scratch de un estudiante por grupo; 2) Rejilla ICOT de ISTE para observar clases; 3) Imágenes de las plantillas de análisis de problemas diligenciadas por los estudiantes y, 4) Grabaciones en video de algunas de las clases (ver los apartados 4.3.2 a 4.3.5). Estos instrumentos se utilizaron en las clases de informática de grado 3° del INSA entre abril y septiembre de 2013.

La información cualitativa recolectada (videos, registros e imágenes) se procesó con la versión 7.1.6 del software atlas.ti. Los registros fueron clasificados de acuerdo a tres dimensiones de las actividades de aula: si correspondían a una aproximación autónoma del estudiante a los conceptos del pensamiento algorítmico, si ocurrían en medio de interacciones con el docente o en interacciones con sus compañeros. Por su parte, las categorías estuvieron relacionadas con cuatro ejes: Analizar, Diseñar, Implementar y Depurar (ver gráfica 11). Estos corresponden a los elementos adoptados en el INSA para articular las actividades de los estudiantes bajo el modelo de Aprendizaje Basado en Problemas (ABP). La gráfica 8 muestra la frecuencia de registros para cada uno de estos ejes.



Gráfica 8: Frecuencia de aparición de los cuatro ejes en los datos cualitativos

Como se puede apreciar en la gráfica 8, Analizar e Implementar fueron los ejes que con mayor frecuencia aparecieron, con 31.4% y 36.3% respectivamente, en relación tanto con conceptos de pensamiento algorítmico, como con los dos tipos de interacciones. Por su parte, Diseñar y Depurar, con 18.4% y 13.9% respectivamente, fueron los ejes con menor ocurrencia.



Gráfica 9: Frecuencia de aparición de las categorías de análisis

En la gráfica 9 se muestran las frecuencias de ocurrencia de los conceptos correspondientes a las categorías Interacciones docente-estudiante(s) e Interacciones entre pares. En cuanto a la labor del docente, se observó que, en el 50% de las interacciones, éste imparte algún tipo de instrucción, lo que duplica tanto las de apoyo (25.6%), como las de evaluación (24.4%). Por otra parte, las interacciones entre pares que predominan son explicar y dialogar con 21.74% y 34.78%, respectivamente.

Ahora bien, si se revisan los datos correspondientes a cada una de estos conceptos, en relación con los ejes de la propuesta didáctica adoptada en INSA, se encontró que en las interacciones docente-estudiante(s) el concepto “instruir” predomina para los ejes analizar, diseñar e implementar; mientras que para el eje depurar, la interacción se dirigió con mayor frecuencia a “evaluar”; tal como lo muestra la gráfica 10.

Se observó además que las interacciones entre pares fueron muy inferiores a las interacciones docente-estudiantes. Esto pudo deberse a que solo se tuvieron en cuenta aquellas interacciones significativas en relación a los conceptos y los ejes propuestos en el libro de categorías (ver Anexo 4). Resultan interesantes dos cosas: primero, que la actividad de depuración de los programas está muy orientada a la evaluación de los productos finales de los estudiantes; segundo, que al sumar todas las interacciones entre pares con las interacciones de apoyo y de evaluación del docente, el resultado es muy superior a las interacciones de instrucción.

La gráfica 10 presenta una matriz de doble entrada que compendia las frecuencias de ocurrencia de los conceptos de cada una de las tres dimensiones de las actividades de aula (pensamiento algorítmico y los dos tipos de interacciones) y las cruza con los momentos o ejes de la actividad de aprendizaje en los que ellas ocurren, de acuerdo con el enfoque de solución de problemas adoptado en el INSA (analizar, diseñar, implementar y depurar).

			APRENDIZAJE BASADO EN PROBLEMAS (Polya)			
			Analizar	Diseñar	Implementar	Depurar
ACTIVIDADES DE AULA	Conceptos del pensamiento algorítmico	1.1.2 Metas	9			
		1.1.3 Organización de datos	9			
		1.1.4 Restricción	10			
		1.1.5 Procesos	9			
		1.1.6 Comprensión lingüística	18			
		1.1.7 Planificación cognitiva	1	11		
		1.2.2 Procesos		5		
		1.2.4 Fondos		0		
		1.2.5 Objetos		9		
		1.3.1 Instrucciones repetitivas			17	
		1.3.3 Instrucciones secuenciales		1	16	
		1.3.4 Eventos			1	
		1.3.5 Automatización			2	
		1.3.6 Paralelismo			4	
		1.3.7 Variables			1	
		1.3.8 Comandos			7	
		1.3.9 Reuso			3	
		1.3.10 Interfaz			10	
		1.4.1 Depuración				7
		1.4.3 Optimización				1
		1.4.4 Prueba y validación				14
	Interacciones Docente-Estudiantes	2.1.1 Instruir	11	16	12	
		2.1.2 Apoyar	5	2	8	5
		2.1.3 Evaluar	3	1	3	12
	Interacciones entre pares	2.2.1 Resolver			1	
		2.2.2 Explicar			5	
		2.2.3 Preguntar	3			
		2.2.4 Observar		1	2	
		2.2.5 Dialogar	2	2	4	
		2.2.6 Ejemplificar		1	1	
		2.2.7 Colaborar	1			

Gráfica 10: Cruce de frecuencias entre dimensiones y ejes.

A continuación se presenta, para cada una de los cuatro ejes del Aprendizaje Basado en Problemas, los hallazgos agrupados en tres dimensiones de las actividades de aula: 1) conceptos del pensamiento algorítmico; 2) interacciones docente-estudiante(s); e, 3) interacciones entre pares.

5.2.1. Conceptos de pensamiento algorítmico e interacciones relacionados con Analizar un problema

Analizar un problema es la actividad que permite aproximarse a la comprensión que elaboran los estudiantes sobre una situación que deben resolver. En esta vía, en la asignatura observada, se inició con un ejercicio de análisis, como primer momento para la resolución del problema. En el análisis se identifican cinco componentes:

- a) Formular el problema (comprensión lingüística),
- b) Especificar los resultados que se desea obtener (identificación de metas),
- c) Identificar la información disponible (organización de datos),
- d) Establecer las condiciones particulares que impone el problema (consideración y cumplimiento de las restricciones), y
- e) Definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (secuencia de operaciones; elaboración del plan).

Para apoyar y orientar el análisis de problemas, en INSA se adoptó un instrumento didáctico denominado “plantilla de análisis de problemas” (ver apartado 4.3.2); este consiste en un documento en el que se presenta el problema por medio de un enunciado y se pide a los estudiantes diligenciar cada uno de los siguientes campos: Formular el problema, Resultados esperados, Datos disponibles, Restricciones y Procesos necesarios (ver Anexo 3).

Por ejemplo, uno de los enunciados especificaba:

“Crear un diálogo entre dos personajes. El diálogo debe ser animado; es decir que los personajes deben moverse e interactuar durante el diálogo. Los objetos y fondos utilizados en el desarrollo de esta actividad pueden ser importados o dibujados. El diálogo debe ser fluido y coherente. Guardar el proyecto con el nombre de Dialogo

01”⁵ (Plantilla_Sep03de2013_3B_NG1).

Frente a esta tarea, los desempeños de los estudiantes en “formular el problema” fueron variados:

“debo hacer un dialogo con dos objetos que interactuen entre si”
(Plantilla_Sep03de2013_3A_AL1)

“tengo que hacer un diálogo entre dos personajes que tienen que interactuar durante el diálogo y pueden ser importados o dibujados y el fondo también y se debe guardar como Dialogo 2” (Plantilla_Sep03de2013_3A_JC1).

También se observó que algunos estudiantes no comprendieron cómo debían diligenciar cada uno de los campos de la plantilla. Por ejemplo, un estudiante entendió por "resultados esperados" cómo le iba a ir en la materia y no cómo debía lucir y funcionar el proyecto al final:

“creo que me va ir muy bien que soy” (Plantilla_Sep03de2013_3B_AL).

Con el campo “datos disponibles” se observó que la mayoría de los estudiantes experimentaron dificultades para diligenciarlo. Tal vez los dos campos que menos confusión generaron fueron “restricciones” y “procesos necesarios”. En este último, se evidenció la habilidad de los estudiantes para plantear y seguir conjuntos secuenciales de pasos. A continuación dos ejemplos de esto:

“1. primero voy a importar el fondo 2. segundo voy a importar los objetos que son una bruja y un fantasma 3. Voy a hacer que mis objetos se muevan y queden juntos 4. Despues que empiezen a hablar 5. Se vayan y desaparezcan”
(Plantilla_Sep03de2013_3A_JC2).

“1. Niña (objeto) 2. Niña (objeto) 3. Fondo (el patio de un colegio) 4. Crear el dialogo 5. Guardar mi proyecto con el nombre dialogo2” (Plantilla_Ago27de2013_3B_MCR).

El campo “procesos necesarios” se notó que revistió especial importancia, pues se constituyó en puerta entrada a la fase de diseño. Por lo general, en este campo los

⁵ Aunque en el enunciado se especifica que el proyecto se debe guardar con el nombre “Dialogo01”, el docente cambió esta instrucción para que lo guardaran con el nombre “Dialogo2”.

estudiantes empezaron a expresarse en términos de características del escenario y de los objetos. Características que usaron en la fase de diseño para especificar “comportamientos” de los elementos del proyecto en Scratch.

En relación con las interacciones docente-estudiantes orientadas a la realización de actividades de analizar problemas se encontró en la información cualitativa que aparecen tres tipos de interacción: instrucción, apoyo y evaluación. Al respecto se encontró que el docente además de presentar el enunciado de los problemas en el encabezado de la plantilla de análisis de problemas, instruyó a los estudiantes sobre cómo diligenciar la plantilla y al final de las clases las recogía para revisarlas. Luego inició cada clase devolviendo las plantillas que habían sido diligenciadas en sesiones previas:

“El profesor se dirige a todos los estudiantes y les dice que les va a entregar las plantillas para que las terminen de diligenciar. Les dice que él va a revisarlas antes de que inicien el trabajo en Scratch. Les dice que ‘son mínimo siete líneas de dialogo’ las que deben escribir en las plantillas” (P16:ICOT_ObservacionesGrado3.docx - 16:58).

Otra interacción de tipo instrucción que realizó el docente consistió en presentar las consignas en el tablero. Se observó que las instrucciones de cada actividad quedaban claramente especificadas en el tablero paso por paso (Video_Ago13de2013_Grado3B_Clase1, 3:47-3:48; Video_Ago27de2013_Grado3A_Clase1, 0:04-0:14). Estas instrucciones en ocasiones no solo contribuyeron clarificar el diseño de la tarea sino a recordar las “restricciones” solicitadas:

“1) no importar el fondo; 2) importar sólo dos objetos”; 3) dibujar 3 objetos; y 4) la bomba de aire y las algas hacen parte del fondo” (P16:ICOT_ObservacionesGrado3.docx - 16:13).

También se evidenciaron interacciones de apoyo. Como por ejemplo, cuando los estudiantes le hicieron preguntas al docente sobre algunos ítems de la plantilla:

“¿qué debían escribir en formular el problema, en datos disponibles y en

restricciones” (P16:ICOT_ObservacionesGrado3.docx - 16:52).

Las interacciones de evaluación no se limitaron a la calificación de los trabajos terminados (evaluación sumativa). También se notó que el docente hizo evaluación formativa cuando pasó por los puestos observando lo que los estudiantes habían realizado hasta el momento.

*[El docente] “pasa por cada puesto revisando que los estudiantes lo estén haciendo y a su vez respondiendo preguntas relacionadas con la actividad”
(P16:ICOT_ObservacionesGrado3.docx - 16:3).*

“al finalizar dicha explicación, indica que seguidamente pasará por el puesto de trabajo de cada estudiante revisando aquellos trabajos que ya hayan terminado lo que acababa de explicar. El profesor también pasa por cada puesto resolviendo dudas que tienen los estudiantes, aclarando que no deben profundizar en el diseño de los objetos sino en el movimiento de los mismos y que deben utilizar la función “repetir” en vez de “por siempre”. También pasa por cada puesto asignando una calificación al trabajo de los estudiantes y al mismo tiempo haciendo llamados de atención relacionados con la disciplina” (P16:ICOT_ObservacionesGrado3.docx - 16:46)

En torno a las interacciones entre pares que favorecen la actividad de analizar problemas se identificaron seis tipos de interacción: explicaciones, preguntas, observaciones, diálogos, ejemplificaciones y colaboración. En el eje de análisis, las interacciones que se hallaron con mayor frecuencia fueron preguntar, colaborar y dialogar:

*“algunos estudiantes le preguntaron a sus compañeros qué debían escribir en algunos ítems de la plantilla. Se observó que por lo menos 5 pares de niños trabajaron en grupo, dialogaron y discutieron las instrucciones del docente”
(P16:ICOT_ObservacionesGrado3.docx - 16:53).*

“se observaron estudiantes que avanzaron en su plantilla de análisis de problemas y les faltó realizar el último punto de la misma, otros estaban iniciándola y en ocasiones establecían un diálogo con sus compañeros y realizaban preguntas relacionadas con la plantilla” (P16:ICOT_ObservacionesGrado3.docx - 16:56).

“una pareja de niños se preguntaron entre sí qué significaba datos disponibles y al no

saberlo, empezaron a ingresar datos del usuario y contraseña al computador para ingresar al sistema; luego dialogaron un momento y continuaron revisando la plantilla (P16:ICOT_ObservacionesGrado3.docx - 16:59).

5.2.2. Conceptos de pensamiento algorítmico e interacciones relacionados con Diseñar una solución

Diseñar una solución se entiende como la representación mental o gráfica, mediante símbolos geométricos o dibujos, de la secuencia lógica de los procesos necesarios para llegar a tal solución. En esta categoría del aprendizaje basado en problemas se observó con mucha frecuencia el concepto de pensamiento algorítmico relacionado con la planificación cognitiva:

*“la estudiante regresó sobre el diseño que tenía planificado inicialmente e hizo que dos de los objetos tuvieran parlamentos coherentes”
(Pantalla_Abril23de2013_3B_NG; 09:05-10:32).*

“se evidenció planificación cognitiva regulada por el orden en que ejecutó las acciones. Primero el fondo. Luego importó los objetos. Posteriormente deshabilitó el giro de los tres objetos y procede a crear los disfraces para cumplir con la consigna del profesor. El estudiante se refirió constantemente a las restricciones dadas para realizar el ejercicio” (Pantalla_Junio11de2013_3A_JC; 12:14-13:05).

“se notó que [la estudiante] se basó en la plantilla de análisis de problemas tanto para los pasos, como para los parlamentos. Fue muy asertiva ahora que utiliza la plantilla de análisis” (Pantalla_Sept03de2013_3B_NG; 35:38-42:48).

Una situación que llamó la atención fue que los estudiantes destinaron buena parte del tiempo de clase a mejorar los elementos gráficos de sus proyectos.

Con respecto a las interacciones en las que el docente promueve la realización de actividades de diseño de soluciones, éste alentó permanentemente a los estudiantes para que diseñaran fondos y objetos y los programaran para darles un comportamiento:

“un número mayor de estudiantes trabajaron en la creación del fondo, el diseño de figuras, dibujaron formas y colores, agregaron disfraces a los objetos, etc”
(P16:ICOT_ObservacionesGrado3.docx - 16:10).

“iniciando la sesión, el profesor anunció que ‘vamos a trabajar movimientos, para que nuestras animaciones se vean mejor’. Agregó que se debe: 1) hacer un fondo que incluya: aire, agua y tierra. 2) agregar accesorios: sol o estrellas, etc. 3) importar o agregar tres objetos que puedan estar en cada uno de los espacios creados. Después de esas instrucciones el profesor dijo: ‘así aprenderemos a darle posición al objeto y dirección, así vamos a saber en dónde se va ubicar el objeto y hacía dónde se va a desplazar’” (P16:ICOT_ObservacionesGrado3.docx - 16:27).

La forma en que el docente impartió las instrucciones estuvo muy orientada por un conjunto secuencial de pasos. También se observaron interacciones de apoyo; por ejemplo, cuando el docente se sentó con algunos estudiantes y les explicó sobre el movimiento de los objetos y el momento de espera de los mismos (P16:ICOT_ObservacionesGrado3.docx - 16:62). Las interacciones tipo evaluación se evidenciaron cuando el docente pasó por los puestos y solicitó a los estudiantes que revisaran aspectos relacionados con la sincronización de los diálogos, el movimiento de los objetos, la ortografía en los diálogos, etc. (P16:ICOT_ObservacionesGrado3.docx - 16:62).

Por su parte, las interacciones entre pares que favorecieron el diseño de soluciones se observaron poco. Básicamente, las interacciones en lo relativo al eje de diseño estuvieron marcadas por diálogos y ejemplificaciones:

“algunos estudiantes conversaron con sus compañeros sobre aspectos relacionados con la actividad de clase, explicando y/o ejemplificando entre sí cómo podían realizar los diálogos, otros conversaban sobre aspectos estéticos relacionados con los objetos, tales como tamaño, disfraces, colores y/o sonidos”
(P16:ICOT_ObservacionesGrado3.docx - 16:41).

5.2.3. Conceptos de pensamiento algorítmico e interacciones relacionados con Implementar una solución

La implementación se entiende como la traducción del diseño de la solución a un lenguaje de programación como Scratch. En cuanto a los conceptos de pensamiento algorítmico relacionados con implementar una solución, se observó que el uso de instrucciones secuenciales y repetitivas fue lo que más apareció en la información cualitativa. Por ejemplo:

“la estudiante ubicó la secuencia correcta de instrucciones en los tres objetos de su proyecto” (Pantalla_Junio4de2013_3B_NG; 22:43 – 23:36).

“el programa de este estudiante de 3ºB tiene los comandos adecuados y en la secuencia precisa. Aunque incluye una estructura repetitiva, ésta es básicamente una estructura secuencial de pasos” (Pantalla_Agosto27de2013_3B_NG; 01:18-01:57).

También se observó que algunos estudiantes programaron el movimiento de un objeto entre cinco posiciones de la pantalla, procediendo en forma secuencial. Para ello:

“la estudiante ubicó el objeto en el primer punto de la pantalla e ingresó las coordenadas x, y en la instrucción [deslizar en _ segundos a x: y:]. Luego movió el objeto a la segunda posición y volvió a ingresar en la instrucción [deslizar en _ segundos a x: y:] las coordenadas que aparecen en la información del objeto. Siguió el mismo procedimiento para las otras tres instrucciones de movimiento (Pantalla_Julio30de2013_3B_NG; 03:39-05:48).

Por otra parte se observó que, a medida que los estudiantes programaron fondos y objetos en Scratch, tendieron a utilizar los comandos ya utilizados en proyectos anteriores, bajo el entendido que en la nueva situación también debían funcionar. Para una muestra, tres ejemplos:

“se observó que la estudiante utilizó el comando [por siempre] en todos los objetos, aun cuando no fuera el comando adecuado para solucionar este problema (Pantalla_Agosto06de2013_3B_NG; 23:28-24:18).

“se notó que la estudiante del grupo 3 B hizo transferencia de los trabajos realizados con anterioridad, en los cuales incluyó una instrucción [por siempre]. Ahora creó un hilo para cada parlamento, en lugar de utilizar una estructura secuencial con los parlamentos y los comandos de espera” (Pantalla_Agosto06de2013_3B_NG; 04:51-07:06)

“se observó que la estudiante trató de lograr que los objetos mostraran los mensajes (hablaran), pero no funcionó la secuencia de comandos implementada por ella. Solo apareció uno de los dos mensajes que había incluido en el parlamento del objeto. Siguió utilizando una estructura repetitiva. Ensayó, modificando el tiempo a 0.65 segundos. En este caso se vio claramente que la estudiante utilizó estrategias que le dieron resultado en ejercicios anteriores. Por último, eliminó la estructura repetitiva y volvió a asignar el tiempo de 2 segundos al comando esperar” (Pantalla_Agosto06de2013_3B_NG; 11:12-12:44).

En cuanto al manejo de la interfaz del entorno de programación, se observó que los estudiantes utilizaron con propiedad tanto el editor de pinturas, como las opciones para guardar los trabajos antes de salir del programa. También asignaron nombres significativos a cada objeto:

“se apreció que el estudiante cambió los nombres predeterminados de sus tres objetos (objeto1, objeto3, objeto 4) por otros que representan a cada objeto (Fantasma, LORO, Perro)” (Pantalla_Junio11de2013_3A_JC; 29:05-30:09).

Con respecto a las interacciones en las que el docente promovió la realización de actividades para implementar soluciones, se evidenció que los estudiantes siguieron las instrucciones dadas por el docente en cuanto a los comandos que debían implementar para resolver los problemas. Además, el docente recordó permanentemente cuáles eran los pasos a seguir y los objetivos de la actividad:

“respecto al uso del Scratch, el docente señaló a los estudiantes que tenía como objetivo que ellos profundizaran en el movimiento de los objetos (P16:ICOT_ObservacionesGrado3.docx - 16:18).

“el docente instruyó: finalizando el fondo, le deben agregar movimiento a los objetos siguiendo los siguientes pasos: 1. al presionar, 2. apuntar en dirección (estos pasos sirven para delimitar los movimientos y direcciones de los objetos, por ejemplo, que

los terrestres no se vayan al cielo). El paso dos, tiene cuatro opciones numéricas para establecer direcciones, la primera de ellas es de noventa que significa a la derecha, la siguiente es -90 que se refiere a la izquierda, está el 0 indicando que el objeto se moverá hacia arriba, y finalmente se ubica el 180 indicando una dirección hacia abajo” (P16:ICOT_ObservacionesGrado3.docx - 16:23).

“iniciando la sesión, el profesor anunció: ‘Vamos a trabajar movimientos, para que nuestras animaciones se vean mejor’. Agregó que deben: ‘1) hacer un fondo que incluya: aire, agua y tierra. 2) agregar accesorios: sol o estrellas, etc. 3) importar o agregar tres objetos que puedan estar en cada uno de los espacios creados’. Después de esas instrucciones el profesor dijo: ‘así aprenderemos a darle posición al objeto y dirección, así vamos a saber en dónde se va ubicar el objeto y hacía dónde se va a desplazar’. Para lograr todo lo anterior deben agregar los siguientes bloques: 1) al presionar bandera verde, 2) apuntar en dirección” (P16:ICOT_ObservacionesGrado3.docx - 16:27).

“el profesor solicitó a los estudiantes que dirigieran su atención al tablero para explicar la actividad de clase que debían desarrollar. La actividad consistió en la creación de diálogos en Scratch. Los estudiantes debían distinguir entre los comandos ‘decir’ y ‘pensar’. Seguidamente, el profesor preguntó a los estudiantes sobre qué aparecería si presionan la bandera verde, ante lo cual ellos generaron distintas respuestas que podían ser o no las indicadas en relación con el problema planteado (P16:ICOT_ObservacionesGrado3.docx - 16:38).

En relación con las interacciones de apoyo, el docente actuó como facilitador la mayor parte del tiempo. Respondió preguntas de los estudiantes y les indicó, paso a paso, qué debían hacer para resolver los problemas.

“unos pocos estudiantes no habían terminado la actividad y el docente se dedicó a pasar por sus puestos para ofrecerles ayuda en la culminación del trabajo” (P16:ICOT_ObservacionesGrado3.docx - 16:21).

“el profesor explicó a los estudiantes cómo debían realizar el examen práctico: ‘1) marcar el examen; 2) poner el número de lista; 3) leer el enunciado del problema’. Luego, el profesor preguntó a los estudiantes si tenían preguntas. Las respondió y les dijo que el tiempo era corto y que debían empezar a trabajar” (P16:ICOT_ObservacionesGrado3.docx - 16:29).

En torno a las interacciones entre pares que favorecen la actividad de implementar soluciones se apreció que los estudiantes que iban terminando una actividad, tenían como tarea ayudar a sus compañeros. Otras interacciones que se observaron con frecuencia fueron el diálogo y la explicación:

“aunque el trabajo fue propuesto de manera individual, algunos estudiantes dialogaron entre ellos sobre aspectos relacionados con la tarea, sobre el uso de Scratch y/o se ayudaron durante el trabajo de programación”
(P16:ICOT_ObservacionesGrado3.docx - 16:8).

“todos los estudiantes trabajaron en la actividad propuesta por el profesor. Sin embargo, quienes iban terminando le mostraron a sus compañeros lo que hicieron”
(P16:ICOT_ObservacionesGrado3.docx - 16:37).

“algunos estudiantes conversaron con sus compañeros sobre aspectos relacionados con la actividad de clase, explicando y/o ejemplificando entre sí cómo podían realizar los diálogos, otros conversaron sobre aspectos estéticos relacionados con los objetos, tales como tamaño, disfraces, colores y/o sonidos”
(P16:ICOT_ObservacionesGrado3.docx - 16:41).

“otros estudiantes, entre los que se encontraban los que ya habían terminando, se acercaron a sus compañeros a observar su trabajo o a decirles qué tipo de algoritmos podían agregar o quitar de la pantalla de programación”
(P16:ICOT_ObservacionesGrado3.docx - 16:64).

5.2.4. Conceptos de pensamiento algorítmico e interacciones relacionados con Depurar una solución

Depurar hace referencia a probar, validar y refinar una solución. Los conceptos de pensamiento algorítmico relacionados con depurar una solución se evidenciaron básicamente con “probar y validar” y “depurar”. Se observó que los estudiantes probaron y validaron permanentemente sus programas:

“el estudiante trató, mediante la instrucción [ir a x, y], de darle una posición inicial a cada uno de los tres objetos que ubicó en el escenario. Sin embargo, cada vez que

ejecutaba el programa, el objeto 'tiburón' quedaba en una posición fuera del agua (Pantalla_Junio4de2013_3A_JC; 18:46-21:20).

"la estudiante ubicó manualmente en una posición al objeto 'jemela' y cuando hacía clic sobre la bandera verde [ejecutar el programa], el objeto cambió de posición, por lo tanto, lo ubicó nuevamente y cambió las coordenadas de manera correcta. Ambas coordenadas eran positivas, estaban en el cuadrante I. (Pantalla_Junio4de2013_3B_NG; 27:33-29:12).

"al programar el objeto ave, el estudiante utilizó los mismos comandos del objeto perro, pero no en el mismo orden. Además, se observó que al probar el funcionamiento del ave, se dio cuenta que al perro le había puesto dentro del ciclo repetitivo la instrucción [decir hola por 2 segundos]. Fue al programa del perro y sacó esa instrucción del ciclo y la puso al comienzo (Pantalla_Junio11de2013_3A_JC; 22:14-24:33).

En cuanto a depurar, los estudiantes tendieron constantemente a mejorar sus programas o a corregir aquello que no funcionó de acuerdo con el plan que se trazaron en las fases de análisis y diseño. Este tipo de acción requiere motivación e interés para lograr trabajos destacados, novedosos, diferentes, que vayan más allá de lo solicitado por el docente:

"el estudiante logró realizar la tarea asignada por el profesor. Entonces, se dedicó a mejorar el proyecto agregando un objeto sombrero que ubicó sobre el objeto perro" (Pantalla_Junio4de2013_3A_JC; 24:01-24:51).

"un estudiante del grupo 3ºA refinó el algoritmo incluyendo al comienzo de la secuencia de comandos la instrucción [mostrar], la cual estaba en otro hilo. Posteriormente eliminó el hilo (Pantalla_Sept10de2013_3A_JC; 44:43-45:06)

Con respecto a las interacciones en las que el docente promueve la realización de actividades que demanden depurar soluciones se observó que éste les formuló preguntas que conducían a mejorar los programas:

"algunas de las preguntas que el profesor hizo a sus estudiantes fueron: ¿cómo haría para que un objeto utilice el comando 'decir' y el otro objeto el comando 'pensar', en un tiempo determinado" (P16:ICOT_ObservacionesGrado3.docx - 16:39).

Por último, en la información cualitativa analizada, no se observó ningún tipo de interacción entre pares que favoreciera la depuración de una solución.

6. Análisis y discusión de resultados

Este trabajo de grado tiene como objetivo central responder la pregunta: ¿Cuáles son las características de las actividades de aula bajo el modelo de resolución de problemas, del entorno de programación Scratch y de las interacciones en el aula, que favorecen el uso de los conceptos del pensamiento algorítmico por parte de los estudiantes de grado 3° de básica primaria del INSA?

Para tal efecto se recolectó y procesó información cuantitativa mediante la aplicación en dos ocasiones (al iniciar y al finalizar el año lectivo) de una prueba cuyo objeto fue describir niveles de desempeño de los estudiantes en tareas que demandan el uso de conceptos del pensamiento algorítmico (ver apartado 4.3.1). Así mismo, se recolectó y procesó información cualitativa mediante cuatro instrumentos (ver apartados 4.3.2 a 4.3.5).

6.1 Análisis cuantitativo

El criterio para determinar si hubo cambio en las habilidades de pensamiento algorítmico fue el siguiente: disminución en el número de estudiantes con nivel de desempeño básico y aumento en la cantidad con nivel avanzado.

La información cuantitativa evidencia que sí hubo un cambio positivo en las tareas “dibuja y ordena los objetos” y “gánate los puntos”, vistas de manera combinada. Tal como se observa en la gráfica 6, en general, hubo una disminución de 4.10% en el número de estudiantes con nivel de desempeño básico y un aumento de 6.56% en el nivel de desempeño avanzado para las dos tareas.

Cabe recordar que la tarea “dibuja y ordena los objetos” demandaba de los estudiantes seguir mentalmente y en paralelo la secuencia de instrucciones para poder armar el modelo resultante de manera que cupiera en el espacio asignado en forma de rectángulo (planificación, paralelismo). Luego debían dibujar los objetos en el orden que indica el modelo mental. Por su parte, la tarea “gánate los puntos” demandaba utilizar reglas condicionales y mantenerlas activas durante la lectura del poema para poder identificar las palabras que cumplieran con las condiciones dadas. Ambas tareas demandaron del estudiante comprensión lingüística para entender las instrucciones dadas en el enunciado.

Aunque en los datos globales aparece un cambio que indicaría un avance en el desarrollo de habilidades de pensamiento algorítmico, al revisar detenidamente los desempeños en cada una de las tareas, se encuentra cambio significativo solamente en la tarea “gánate los puntos”: el nivel básico disminuyó y el nivel avanzado aumentó.

Por su parte, la tarea “dibuja y ordena los objetos” presentó cambio en sentido contrario a lo esperado, indicando puntajes mayores en la aplicación inicial que en la final para el nivel básico y un aumento muy leve en el nivel avanzado. Tal como está planteada, esta tarea busca establecer la cantidad de información que el estudiante puede tener disponible, al mismo tiempo, para generar un modelo (ver apartado 4.3.1.1). En esa medida, los desempeños con más alta calificación, son aquellos que muestren una representación mental del modelo completo, antes de iniciar el dibujo. En este punto, vale la pena preguntarse ¿qué información fue más relevante para los niños al realizar esta tarea?: ¿seguir minuciosamente unas instrucciones dadas en forma secuencial; o elaborar un modelo mental, que incluya todos los elementos y relaciones previos a la instrucción explícita de dibujar?

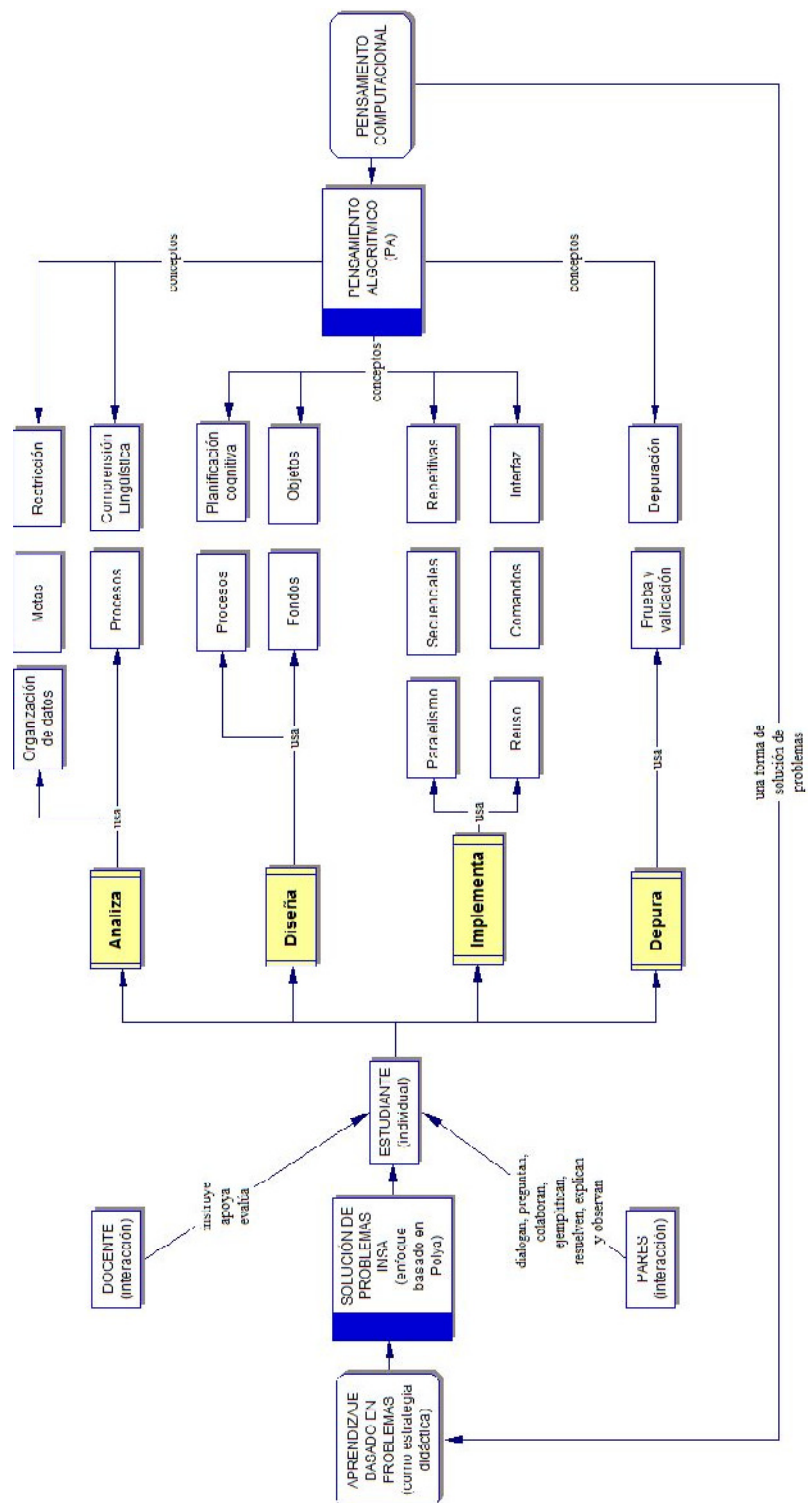
Al revisar las calificaciones de esta tarea se observa que los estudiantes que disminuyeron su nivel de desempeño, en la aplicación inicial integraron toda la

información en un solo modelo, pero en la aplicación final dibujaron parejas de objetos. Dado que para armar el modelo se debe seguir mentalmente y en paralelo la secuencia de instrucciones, los estudiantes muy seguramente activaron en la aplicación final el procesamiento secuencial y dejaron de lado el procesamiento paralelo. Esto se podría explicar por el tipo de trabajo secuencial que se realiza en clase. Los datos cualitativos apoyan esta hipótesis, pues se observa que en clase se hace mucho énfasis en los procedimientos secuenciales; luego, los datos y los desempeños muestran que los niños son consistentes con las actividades que realizan en las clases y tienden a utilizar los recursos vistos previamente. Este es el caso, solo por citar un ejemplo, cuando los estudiantes utilizaron en la programación de sus objetos el comando [por siempre], usado en proyectos anteriores; aunque en la situación presente se requería el comando [repetir 10], ellos asumieron que el comando [por siempre] debía funcionar en la nueva situación.

Al enfrentar las dos tareas con lo que dice la teoría respecto del pensamiento algorítmico, se aprecia que éstas solo cubren el 20% de los conceptos observados del pensamiento algorítmico: planificación cognitiva, comprensión lingüística, paralelismo y condicionales. Por lo tanto, las dos tareas usadas en la prueba no aportan suficientes elementos de validez para confirmar una relación entre el desarrollo del pensamiento algorítmico y la intervención educativa que se adelanta en el INSA. Además, el instrumento utilizado no garantiza que los estudiantes lo entiendan de manera consistente. Pues tal como se evidencia en los resultados de la tarea “dibuja y ordena los objetos”, los estudiantes demostraron aproximaciones diferentes frente a la misma tarea entre una y otra aplicación. De manera que, aunque aparece un claro indicador de cambio en los desempeños de los mismos, estos datos no explican suficientemente el uso de conceptos del pensamiento algorítmico y se hace necesario complementarlos con datos cualitativos.

6.2 Análisis cualitativo

Del análisis de la información cualitativa (ver gráfica 10), a la luz de las categorías de análisis propuestas (ver Anexo 4), emergió un modelo categorial que relaciona la intervención educativa que se lleva a cabo en las clases de informática en el INSA con el uso de conceptos del pensamiento algorítmico por parte de los estudiantes (ver gráfica 11). Concretamente, de la intervención educativa se analizaron tres dimensiones fundamentales de las actividades de aula observadas: a) los conceptos del pensamiento algorítmico que los estudiantes utilizan; b) las interacciones entre docente y estudiantes, en términos de acciones que promueven el enfoque de solución de problemas; y, c) las interacciones entre pares que favorecen la solución de problemas con el entorno de programación Scratch.



Gráfica 11: Modelo categorial empírico

Este modelo categorial esquematiza la forma cómo el enfoque basado en Polya para solución de problemas actúa como articulador de las actividades de aula en las clases de informática en INSA. Dicho enfoque se operacionaliza mediante cuatro ejes, fases o momentos de la actividad planteada por el docente (analizar, diseñar, implementar y depurar). A medida que los estudiantes van recorriendo cada uno de estos ejes articuladores, por un lado se ven inmersos en una serie de interacciones con el docente y con sus pares y, por la otra, se apoyan en diferentes conceptos del pensamiento algorítmico para avanzar en la solución del problema con Scratch.

A continuación se procede a analizar y discutir los resultados cualitativos, agrupados de acuerdo a cada una de las tres dimensiones de las actividades de aula que emergieron de los datos (ver gráficas 10 y 11).

6.2.1 Conceptos del Pensamiento Algorítmico

Los conceptos de pensamiento algorítmico relacionados con Implementar fueron los que más ocurrencias presentaron con 37.0% del total. En este eje los estudiantes trabajan todo el tiempo con Scratch, lo cual amerita hacer énfasis en que hay diferencia entre programar en Scratch y solucionar problemas con Scratch. En el primer caso, los estudiantes exploran el entorno de programación y tratan de usar su creatividad para elaborar programas interesantes para ellos, lo cual ya tiene valor por sí mismo; sin embargo, según López (2009), cuando se solucionan problemas con Scratch, los estudiantes deben, necesariamente y de forma consciente, tanto activar estrategias cognitivas como usar recursos y conceptos del pensamiento computacional para poder resolverlos.

En el INSA esta habilidad general para resolver problemas se apoya en una herramienta como Scratch que constituye “un andamiaje para nuevas formas de pensar y razonar en la zona de desarrollo que se encuentra entre las capacidades

que ya tienen los estudiantes y sus capacidades potenciales” (Jonassen, 2000; p 13). Esta perspectiva vigotskiana enfatiza “la reorganización funcional de la cognición con el uso de tecnologías simbólicas” (Jonassen, 2000; p 13). En el caso de la propuesta educativa del INSA, el uso de Scratch no solo se enmarca en la categoría de “herramienta de la mente” propuesta por Jonassen (2000) o en la de “auxiliares exteriores” que según Ivic (1994) plantea Vigotsky, sino que se erige en mediadora tanto para implementar la estrategia de solución de problemas basada en el enfoque de Polya como para posibilitar que los estudiantes tengan la oportunidad de utilizar conceptos del pensamiento algorítmico.

Adicionalmente, en el eje de Implementación los estudiantes utilizaron intensivamente solo tres de los seis conceptos del pensamiento algorítmico: estructura de control secuencial, estructura repetitiva y la interfaz de Scratch. La resolución de problemas observada en este eje no demandó el uso de la estructura condicional, por lo tanto esta no se vio reflejada en los trabajos de los estudiantes; lo cual es importante dado que la tarea “gánate los puntos” del instrumento cuantitativo (ver apartado 4.3.1) se enfoca en medir los desempeños en el uso de condicionales. Todo esto lo explica el hecho que los estudiantes de grado 3° que participaron en este estudio apenas se estaban iniciando en el uso de Scratch y los problemas que planteó el docente fueron relativamente sencillos de resolver. Sin embargo, cuando en grados posteriores estos estudiantes aborden problemas más complejos, se espera que utilicen, para solucionarlos, mayor cantidad de conceptos del pensamiento algorítmico relacionados con el eje Implementar.

Por su parte, los conceptos relacionados con Analizar representaron el 34.0% de las ocurrencias, solo tres puntos porcentuales menos que el eje Implementar. Las tareas que los estudiantes realizan en este eje se derivan de la metodología de solución de problemas basada en Polya y los estudiantes utilizan una plantilla impresa para analizar el problema planteado por el docente. Es el único eje en el cual los estudiantes no utilizan el entorno de programación Scratch, pues “es tonto

tratar de responder una pregunta que usted no comprende” (Polya, 1957, pág. 6). A pesar de las dificultades que entraña para algunos estudiantes de grado 3° diligenciar la plantilla de análisis, aquellos que la resolvieron correctamente fueron más asertivos en el uso de los comandos de Scratch cuando pasaron a la fase de Implementación. Esta plantilla demanda de los estudiantes pensar en diferentes aspectos de los problemas; recordemos que según Schunk (1997), analizar un problema se entiende como especificar los resultados que se desean obtener (metas), identificar la información disponible (organización de datos), establecer las restricciones que impone el problema y definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (operaciones). Además, la plantilla les facilita el diseño de una solución dado que el campo “Procesos necesarios” es crucial para dar un orden algorítmico a cualquier solución que el estudiante proponga; mientras que los campos “Metas” y “Restricciones” ayudan a configurar los comportamientos tanto del escenario como de los objetos.

De acuerdo con el modelo categorial empírico, a cada uno de los cuatro ejes del enfoque de solución de problemas le corresponde un número determinado de conceptos del pensamiento algorítmico (ver gráfica 11). Los resultados muestran que en los ejes Diseñar, Implementar y Depurar las frecuencias de ocurrencia se concentran en uno o dos de los conceptos correspondientes. Mientras que para el eje analizar, cinco de sus seis conceptos tienen frecuencia de ocurrencia alta y similar (ver gráfica 10). En términos del desarrollo de la habilidad de resolución de problemas esto demuestra que en cada actividad de aula los estudiantes de grado 3° del INSA realizan sistemáticamente todos los pasos propuestos en la metodología de análisis de problemas.

En cuanto a los conceptos relacionados con Diseñar, estos representan solo el 15.7% de las ocurrencias, 21.3 puntos porcentuales menos que el eje Implementar. Aquí, los conceptos que presentaron mayor ocurrencia fueron “Planificación cognitiva” y “Objetos”. El concepto “Procesos” fue uno de los menos utilizados, lo cual llama la atención ya que tiene una fuerte relación con la

definición de algoritmo: “herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible” (López, 2009; p 21).

Dado que diseñar una solución “se entiende como la representación mental o gráfica, mediante símbolos geométricos o dibujos, de la secuencia lógica de los procesos para implementar una solución” (López, 2009; p 21), toda la carga del eje Diseño corrió principalmente por cuenta del concepto “Planificación Cognitiva”. Esto si entendemos “Planificación cognitiva” como a) considerar los pasos previstos en la sección “Procesos” de la plantilla de análisis; b) seleccionar y decidir las operaciones a efectuar; y, c) planificar, formular hipótesis y anticipar qué sucederá. Hilando más delgado, de estos tres elementos de la planificación, el que más ocurrencia presentó fue el b.

El otro concepto con ocurrencia alta dentro de este eje fue “Objetos”. Aquí se esperaba que los estudiantes además de la apariencia de los objetos, trabajaran en los comportamientos que estos deberían tener. Sin embargo, lo que se observó fue que ellos se concentraron en la apariencia (disfraces, nombres, colores, etc). Solo en la actividad del diálogo, que el docente incluyó en la plantilla de análisis un paso adicional para que los estudiantes escribieran el parlamento de dos objetos, fue que se apreció un atisbo de formalizar el comportamiento de los objetos. Aparte de este paso en la plantilla de análisis, los estudiantes no realizaron procedimiento alguno que permitiera plantear formalmente el diseño de la solución. Dado que ese paso adicional ayudó a los estudiantes a pensar en términos del comportamiento de los objetos, se plantea como hipótesis que se puede facilitar a los estudiantes la transición entre la fase de análisis y de implementación si en la de diseño se utiliza una herramienta como la propuesta por López (2011) que les permita pensar los escenarios y objetos en términos de apariencia, comportamiento, comandos a utilizar y secuencias de instrucciones (ver Anexo 6).

Por último, los conceptos relacionados con Depurar representan el 13.3% de las ocurrencias, cifra inferior en 23.7 puntos porcentuales al eje Implementar. La mayoría de las ocurrencias en este eje correspondieron a “Prueba y validación” del funcionamiento de los programas, lo cual se da de manera muy natural dado que los estudiantes se acostumbran a probar permanentemente sus programas. Con respecto a “Depuración”, se observaron varias iniciativas de estudiantes por mejorar y corregir sus programas, algunas de ellas, promovidas por el docente.

La fase de depuración de programas de computador demanda un nivel alto de reflexión. Para Stager (2003; párrafo 28), “depurar programas contribuye a mejorar la capacidad para resolver problemas. Nada funciona de manera correcta la primera vez. La depuración basada en la retroalimentación es una habilidad útil para toda la vida”.

En teoría, el pensamiento computacional es una forma de solución de problemas que incluye al pensamiento algorítmico y en la práctica se evidenció que el enfoque de Polya constituye un eje articulador y metodológico para llevar a cabo actividades de aula orientadas a promover el desarrollo de la habilidad para resolver problemas. Con base en esto, se propone en este trabajo como hipótesis que al utilizar una herramienta de la mente como Scratch para resolver problemas, haciendo uso de conceptos del pensamiento algorítmico, necesariamente los estudiantes deberían desarrollar dicho pensamiento. Esta hipótesis se sustenta en la afirmación de Jonassen (2000) de que las herramientas de la mente tienen efectos sobre el aprendizaje cuando los estudiantes establecen una asociación intelectual con el computador; cuando ellos trabajan con computadores, mejoran las capacidades de pensamiento y aprendizaje.

No se pretende afirmar aquí que esta sea la única forma en que este pensamiento se desarrolla, pues siguiendo la corriente vigotskiana “el desarrollo consiste en la formación de funciones compuestas, de sistemas de funciones, de funciones sistemáticas y de sistemas funcionales” (Ivic, 1994; p 5). Sin embargo, al utilizar

Scratch estaríamos frente a una situación de “desarrollo artificial”, situación que según Ivic (1994) se enmarca en el desarrollo sociocultural de las funciones cognitivas mediante el apoyo de auxiliares exteriores.

6.2.2 Interacciones docente-estudiante(s)

Respecto a las interacciones docente-estudiante(s), de los datos emergieron básicamente tres tipos de interacción: instruir, apoyar y evaluar. Cuando se analizan los resultados desde la perspectiva del Aprendizaje Basado en Problemas, la mayoría de las ocurrencias se dieron en el cruce con el eje Implementar (29,5%). Los ejes Analizar, Diseñar y Depurar tuvieron una frecuencia de ocurrencias muy similar entre sí: 24,4%, 24,4% y 21.7%, respectivamente.

Cuando se analizan los resultados desde la dimensión “Interacción docente-estudiante(s)”, primó la de tipo instrucción (50%) sobre la de apoyo (26%) y de evaluación (24%). No obstante, cabe aclarar que no todas las interacciones en las que el docente impartió instrucción a todo el grupo fueron directivas (paso a paso). En varios casos se observó cómo el docente, al tiempo que instruía, realizaba preguntas a los estudiantes y los alentaba a formular hipótesis sobre lo que sucedería en Scratch si hacían lo que él estaba planteando en el tablero.

Según Baquero (2004), los intercambios discursivos entre docente y estudiante(s) guardan relación con las categorías de zona de desarrollo próximo y de andamiaje. En este sentido, Baquero (2004) advierte que se han atribuido efectos cognitivos importantes a la formulación de preguntas por parte del docente, especialmente en la estimulación del pensamiento de los alumnos. Esta forma de interacción docente-estudiante(s), basada en preguntas, se observó constantemente en las clases analizadas.

Sin embargo, habría que tener en cuenta la variable “manejo de la clase” para

explicar la ocurrencia tan alta de interacciones tipo instrucción. Dado que mantener el orden en un aula con estudiantes de grado 3° es una tarea retadora para cualquier docente, utilizar una secuencia paso a paso puede ser una estrategia usada por el docente para que la clase no se salga de control.

Por otra parte, en los cuatro ejes se evidenció permanentemente el papel de guía y apoyo que cumplió el docente, caminando por entre los puestos y respondiendo todo tipo de preguntas. Cuando encontraba alguna situación que se repetía con varios estudiantes, se dirigía al tablero y la aclaraba para todo el grupo. En este sentido, se identificó una “asistencia andamiada” del docente en diversas formas: aclaraciones generales a todo el grupo, sugerencias, recomendaciones puntuales, ejemplificaciones y explicaciones.

Desde la perspectiva vigotskiana de la zona de desarrollo próximo, el apoyo del docente, como sujeto con mayor dominio sobre los problemas planteados, resulta fundamental en cualquier propuesta que utilice la solución de problemas como estrategia didáctica. En este sentido, se formula aquí como hipótesis que emplear elementos propios del andamiaje en las clases de informática contribuye a que los estudiantes desarrollen la habilidad para solucionar problemas.

En cuanto a las interacciones de tipo evaluación, éstas se concentraron en un alto porcentaje en el eje depuración; eje para el cual no se observó ninguna interacción de tipo instrucción. Cabe destacar, que la evaluación no se limitó a producir una calificación, buena parte de esta actividad el docente la orientó a dar retroalimentación a los estudiantes; retroalimentación que varias veces se presentó en forma de pregunta que demandaba del estudiante elaborar nuevas formas de plantear la solución.

Esta evaluación en forma de retroalimentación es crucial en un entorno de andamiaje: “el tutor ofrece apoyo y retroalimentación, y aplica evaluaciones frecuentes para determinar cuándo disminuir la asistencia” (Jiménez, 2008; p 225).

6.2.3 Interacciones entre pares

La interacción entre pares representa solo un 23% del total, mientras que las interacciones docente-estudiante(s) corresponden al 77% de las ocurrencias. En las interacciones entre pares se identificaron siete tipos de interacción: resolver, explicar, preguntar, observar, dialogar, ejemplificar y colaborar. La mayoría de estas interacciones se dieron en relación al eje Implementar con 57% de las ocurrencias. En tanto, los ejes Analizar y Diseñar presentaron 26% y 17% de las ocurrencias, respectivamente. En el eje depurar no se observó ninguna ocurrencia.

Entre los tipos de interacción entre pares, sobresalen dos en particular: Dialogar (35%) y Explicar (22%), seguidas de Preguntar y Observar, con 13% cada una. Estos tipos de interacción permiten que los estudiantes asuman roles complementarios al papel del docente; por ejemplo, entre pares se facilita formular preguntas y obtener respuestas, dar explicaciones o pedir las, observar cómo se realiza un procedimiento o mostrarlo, establecer diálogos exploratorios entre compañeros, etc. Según Baquero (2004), las interacciones entre pares, en el contexto de actividades de aula, guardan un efecto positivo en el desarrollo de capacidades de los sujetos involucrados.

Según Cazden, citado por Baquero (2004), el valor intelectual de las interacciones entre iguales en el aula de clase, se incrementa si el maestro modela algún tipo de interacción que los niños puedan aplicar más tarde entre ellos. Esto apoya la hipótesis que utilizar una herramienta como la espiral de la creatividad (Resnick, 2007), especialmente en el paso de compartir, contribuye efectivamente a las tareas de los ejes de diseño, implementación y depuración.

A pesar de algunos reparos que se plantean a los beneficios positivos de la zona de desarrollo próximo cuando se dan interacciones entre pares (Baquero, 2004), lo

que se observó con los estudiantes de grado 3° del INSA es que cuando interactuaron entre ellos, la ayuda del par “más capaz” fue determinante para el avance en la actividad del par “menos capaz”. Pues aunque las actividades de aula inician con un trabajo individual por parte de los estudiantes, tan pronto se encuentran en una situación que no saben resolver, inmediatamente acuden al docente o a sus compañeros para recibir el apoyo necesario para poder continuar con el trabajo. Por lo tanto, se plantea como hipótesis que Scratch, en un contexto de solución de problemas, demanda del estudiante que interactúe con el docente o con sus compañeros para resolver situaciones que individualmente o en solitario, no podría.

7. Conclusiones y recomendaciones

El análisis de los resultados permite concluir que, aunque en los datos cuantitativos aparece un claro indicador de cambio en los desempeños de los estudiantes, estos desempeños no explican suficientemente el uso de conceptos del pensamiento algorítmico. Por otra parte, los datos cualitativos si permitieron establecer un modelo categorial empírico que explica cómo los estudiantes de grado 3° del INSA usan y se apropian de los conceptos del pensamiento algorítmico en un entorno educativo que tiene como eje articulador la metodología de solución de problemas propuesta por Polya.

En dicho entorno, los conceptos del pensamiento algorítmico constituyen, junto a las interacciones docente-estudiante(s) y entre pares, dimensiones de las actividades de aula diseñadas por el docente de informática. La dimensión conceptos del pensamiento algorítmico explica la forma cómo están inmersos en mayor o menor grado estos conceptos en cada actividad de aula que plantea el docente. La dimensión interacción docente-estudiante(s) explica cómo algunas de las actividades realizadas por el docente favorecen la estrategia didáctica de

solución de problemas. Por último, la dimensión interacción entre pares da cuenta de las formas en que unos estudiantes trabajan con otros con el fin de lograr resolver problemas con Scratch.

En tanto, dadas las características del entorno de programación Scratch, unidas a las dimensiones y eje articulador de las actividades de aula, se encontró que los estudiantes, con mayor o menor frecuencia, utilizan grupos de conceptos en cada uno de los cuatro ejes que conforman la metodología de solución de problemas adoptada en INSA. Por ejemplo, los estudiantes se apoyaron en conceptos tales como “comprensión lingüística”, “organización de datos”, “metas”, “restricciones” y “procesos” para analizar los problemas formulados por el docente. Las soluciones a los problemas planteados las diseñaron utilizando, principalmente, los conceptos “planificación cognitiva” y “objetos” y al implementar la solución en Scratch se apoyaron en los conceptos “estructuras secuenciales”, “estructura repetitivas” e “interfaz”. En relación a la depuración de los programas, básicamente utilizaron el concepto “prueba y validación”. Vale la pena poner de presente que estos ejes no necesariamente se recorren de manera secuencial; es un proceso dinámico y flexible en el que los estudiantes pueden avanzar de un eje a otro y luego regresar al anterior (ver gráfica 1).

Por su parte, las interacciones en el aula, que promueven y favorecen la solución de problemas, se presentan de dos formas: docente-estudiante(s) y entre pares. En la primera se identificaron tres tipos de interacción del docente: instrucción, apoyo y evaluación. Aunque la instrucción fue el tipo de interacción que más se presentó, resulta interesante observar que si se suman todas las interacciones entre pares con las interacciones docente-estudiante(s) relacionadas con apoyo y evaluación, se encuentra que el 64% del total de las interacciones son diferentes a instrucción. Se evidencia mucho apoyo, entre docente-estudiantes y entre pares; lo cual desde una perspectiva vigotskiana de la zona de desarrollo próximo, resulta fundamental en cualquier propuesta que utilice el ABP como estrategia didáctica.

En cuanto a los objetivos específicos de la investigación, se puede concluir que se cumplieron en virtud de que:

1. Se caracterizaron en detalle los ejes, fases o momentos que conforman la propuesta de Aprendizaje Basada en Problemas adoptada por el INSA como estrategia didáctica. En este punto, no solo se identificaron los ejes analizar, diseñar, implementar y depurar, sino que se determinó que el enfoque para solución de problemas de Polya constituye un eje central y articulador en las actividades de aula que se llevan a cabo en las clases de informática con Scratch de grado 3° en el INSA.
2. Se identificaron tres dimensiones de las actividades de aula que inciden directa o indirectamente en el desarrollo del pensamiento algorítmico: a) conceptos del pensamiento algorítmico que al usarlos frecuente y sistemáticamente se espera potencien su desarrollo; b) interacciones docente-estudiante(s) que en un entorno de solución de problemas promueven el uso de los conceptos del pensamiento algorítmico; y, c) interacciones entre pares que favorecen la solución de problemas con Scratch.
3. Se identificó que programar con Scratch no es lo mismo que resolver problemas con Scratch. En este último caso, los estudiantes deben activar estrategias cognitivas así como usar recursos y conceptos del pensamiento computacional para poder resolverlos. Además, se determinó que Scratch se enmarca en la categoría de “herramientas de la mente” o en la de “auxiliares exteriores”, por cuanto contribuye no solo al uso y apropiación del pensamiento algorítmico sino que se erige como mediadora tanto para implementar la estrategia de solución de problemas basada en el enfoque de Polya, como para posibilitar que los estudiantes tengan la oportunidad de utilizar conceptos del pensamiento algorítmico.
4. Se plantea como hipótesis que el desarrollo del pensamiento algorítmico,

en el contexto analizado, se da en función del uso de sus conceptos en un entorno educativo que tiene como eje articulador un enfoque de solución de problemas como estrategia didáctica, pero que cuenta además con una herramienta de la mente como Scratch y con unas interacciones que proveen el andamiaje necesario para operacionalizar la metodología propuesta. Recordemos que para Vigotsky, “lo fundamental en el desarrollo no estriba en el progreso de cada función considerada por separado sino en el cambio de las relaciones entre las distintas funciones, tales como la memoria lógica, el pensamiento verbal, etc.” (Ivic, 1994; pp 4-5).

A partir del cumplimiento de los objetivos específicos, se puede concluir que también se cumplió el objetivo general de esta investigación dado que para las actividades de aula que se realizan en la clase de informática de grado 3° del INSA, se identificaron unas características particulares que en conjunción con el uso de Scratch como herramienta de la mente, ejercen una influencia en el uso y apropiación de conceptos del pensamiento algorítmico.

De las conclusiones de este trabajo de grado se desprenden varias recomendaciones, tanto para futuras investigaciones, como para los docentes interesados en diseñar mejores actividades de aula basadas en solución de problemas con Scratch.

Para futuras investigaciones se recomienda:

- Tener en cuenta los conceptos del pensamiento algorítmico representados en el modelo categorial (ver gráfica 11) al diseñar instrumentos de recolección de datos cuantitativos.
- Recolectar datos correspondientes a distintos grados escolares para poder monitorear el desarrollo del pensamiento algorítmico. Esto, debido a que año tras año, los problemas planteados a los estudiantes son más complejos y demandan el uso de mayor cantidad de conceptos del

pensamiento algorítmico lo que permite observar un rango más amplio de desempeños.

- Plantear diseños de investigación más controlados si se desea proponer hipótesis causales en relaciones predictivas. Por ejemplo, usar grupos control con otros tipos de intervención en el aula de otras instituciones educativas.

Para diseñar mejores actividades de aula basadas en solución de problemas con Scratch, se recomienda:

- Fortalecer la capacidad de los docentes para formular buenos problemas.
- Adoptar una perspectiva teórica, como el modelo de aprendizaje sociocultural de Vigotsky, que involucre el reconocimiento de instrumentos mediadores, procesos de andamiaje y la existencia de una 'zona de desarrollo próximo' en la actividad cognitiva de los estudiantes; lo cual tiene implicaciones directas en la práctica docente.
- Diseñar actividades de aula que en las que se promueva una mayor interacción entre pares; pero utilizando, por ejemplo, la espiral del pensamiento creativo propuesta por Resnick (2007) para propiciar espacios de reflexión sobre el trabajo que los estudiantes realizan con Scratch.
- Utilizar más tiempo para realizar cada actividad, pues apropiarse de los conceptos del pensamiento algorítmico toma tiempo, requiere exponerse varias veces a problemas similares y experimentar con comandos nuevos.
- Asegurarse de vincular los conceptos del pensamiento algorítmico representados en el modelo categorial (ver gráfica 11) al diseñar actividades de aula basadas en solución de problemas con Scratch. Por ejemplo, planificación cognitiva, metas, organización de datos, restricciones, procesos, estructuras de control, depuración, etc.

- En el eje correspondiente al diseño de una solución, utilizar una herramienta diferente a Scratch que facilite a los estudiantes construir un puente entre el análisis y la implementación. Dicha herramienta puede ser una plantilla como la propuesta por López (2011) que permita prever los escenarios y objetos en términos de apariencia, comportamiento, comandos a utilizar y secuencias de instrucciones (ver Anexo 6). También puede complementarse con la elaboración de diagramas de flujo para representar las secuencias de instrucciones en pseudocódigo.

Para terminar, se cita a Ivic (1994) refiriéndose al modelo de desarrollo de Vigotsky en relación al papel del aprendizaje en la adquisición del lenguaje. Este afirma que “la contribución del aprendizaje consiste en que pone a disposición del individuo un poderoso instrumento: la lengua. En el proceso de adquisición, este instrumento se convierte en parte integrante de las estructuras psíquicas del individuo (la evolución del lenguaje)” (Ivic, 1994; p 4). Siguiendo esta línea de pensamiento, se plantean algunas preguntas que podrían ser abordadas en trabajos de grado de estudiantes de maestría en psicología: ¿Es posible que aprender a programar computadores sea asimilable, en alguna medida, a la adquisición del lenguaje? ¿Será que en el proceso de aprender un lenguaje de programación, los conceptos del pensamiento algorítmico se convierten en parte integrante de las estructuras psíquicas de los estudiantes?

8. Bibliografía

Baquero, R. (2004). *Vigotsky y el aprendizaje escolar* (4ta ed.). Buenos Aires: Aique Grupo Editor.

Brennan, K. & Resnick, M. (2012). *New Frameworks for Studying and Assessing the Development of Computational Thinking*. American Educational Research Association meeting, (págs. 1-25). Vancouver, BC, Canada.

- Cacioppo, J.T. & Freberg, L.A. (2013). *Discovering Psychology: The Science of Mind*. Belmont: Wadsworth, Cengage Learning.
- Citilab. (2013). *Scratch conference: Scratch Connecting Worlds*. Recuperado el 28 de Noviembre de 2013, de Citilab-Cornellá:
<http://scratch2013bcn.org/program>
- Code (2013). *Every student in every school should have the opportunity to learn computer science*. Recuperado el 12 de Noviembre de 2013, de
<http://code.org/>
- CREA. (2012). *Maestría en Educación: Aprender para enseñar mejor, futuro prometedor para todos*. Recuperado el 27 de Noviembre de 2013, de Icesi:
<http://www.icesi.edu.co/maestrias/educacion/objetivos.php>
- Department for Education. (2013). *National curriculum in England: computing programmes of study*. Recuperado el 17 de Abril de 2014, de GOV.UK:
<http://bit.ly/1gmCybj>
- Ehrlich, K. (1982). *Spatial Descriptions and Referential Continuity*. Journal of Verbal Learning and Verbal Behavior, 21, 296-306.
- Futschek, G. (2006). *Algorithmic Thinking: The Key for Understanding Computer Science*. ISSEP 2006, 159 – 168.
- Hernández, R. F. (2010). *Metodología de la investigación* (Quinta ed.). México: McGraw-Hill.
- Hmelo-Silver, C. E. (2004). *Problem-Based Learning: What and How Do Students Learn?* Educational Psychology Review, 16 (3), 235-266.
- INSA. (2013). *Quiénes somos, Instituto Nuestra Señora de la Asunción*. Recuperado el 17 de Octubre de 2013, de INSA: <http://www.insa-csb.co/>

- ISTE. (2011). *Operational definition for computational thinking for K-12 education*. Recuperado el 12 de Septiembre de 2013, de ISTE:
<http://www.iste.org/learn/computational-thinking/ct-operational-definition>
- ISTE. (2012). *ISTE Classroom Observation Tool (ICOT)*. Recuperado el 5 de Mayo de 2014, de Eduteka: <http://www.eduteka.org/ICOT.php>
- Ivic, I. (1994). *Lev Semionovich Vygotsky*. (UNESCO, Ed.) Perspectivas: revista trimestral de educación comparada, XXIV (3-4), 773-779.
- Jiménez, A. (2008). *Scaffolding tutoring strategy on virtual environments for training*. Ingeniare. Revista chilena de ingeniería, 16 (2), 220-231.
- Jonassen, D. (2000). *Computers as Mindtools for schools*. New Jersey: Prentice-Hall.
- Joyanes, L. (2001). *Fundamentos de programación, algoritmos y estructura de datos* (2da ed.). México: Mc Graw Hill.
- Kerlinger, F. (2002). *Investigación del comportamiento* (Cuarta ed.). México: McGraw-Hill.
- Larkin, M. (2002). *Using Scaffolded Instruction To Optimize Learning*. ERIC Digest, 1-6.
- López, J. C. (2009). *Algoritmos y programación, guía para docentes* (Segunda ed.). Cali: Eduteka.
- López, J. C. (2011). *Programación con Scratch: Cuaderno de Trabajo para Estudiantes* (Cuarta ed.). Cali: Eduteka.
- López, J. C. (2012). *Identificación y regulación de emociones con Scratch*. En J. P. Hernández, Tendencias emergentes en Educación con TIC (págs. 67-81). Barcelona: Espiral.

- López, J. C. (2013a). *Scratch en la educación escolar*. Recuperado el 15 de Febrero de 2014, de Eduteka: <http://www.eduteka.org/modulos/9/278/>
- López, J. C. (2013b). *Herramienta para analizar problemas*. Recuperado el 18 de Abril de 2014, de Eduteka: <http://www.eduteka.org/analisisproblemas.php>
- López, J. C. (2014). ¿Por qué es importante promover que los estudiantes desarrollen el pensamiento computacional? En Mirada Relpé, reflexiones iberoamericanas sobre las TIC en educación (págs. 78-83). Buenos Aires: Relpé.
- McLuhan, M. (1964). *Understanding Media: The Extensions of Man*. Nueva York: McGraw-Hill.
- Morra, S. (2001). *On the information-processing demands of spatial reasoning*. Thinking and reasoning, 7 (4), 347-365.
- Moursund, D. (1999). *Project-Based learning using information technology*. Eugene: ISTE Publications.
- National Research Council. (1999). *Being Fluent with Information Technology*. Washington, D.C.: National Academies Press.
- Papert, S. (1986). *Constructionism: A New Opportunity for Elementary Science Education*. Boston: Massachusetts Institute of Technology. Epistemology & Learning Research Group.
- Papert, S. (1988). *Computer as Material: Messing About with Time*. Recuperado el 15 de Octubre de 2013, de papert.org: <http://www.papert.org/articles/ComputerAsMaterial.html>
- Piaget, J. (1977). *Seis estudios de psicología*. Barcelona: Seix Barral (Éditions Gonthier, Ginebra, 1964).

- Polya, G. (1957). *How to solve it* (segunda ed.). Princeton: Princeton University Press.
- Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. ACM Creativity & Cognition conference, Washington DC, Junio 2007.
- Resnick, M. et al. (2009). *Programming for All*. *Communications of the ACM*, 60-67.
- Resnick, M. (2010). *The Scratch Programming Language and Environment*. ACM Transactions on Computing Education 10(4), Article No. 16.
- Resnick, M. (2013). *Learn To Code, Code To Learn*. Recuperado el 15 de Septiembre de 2013, de EdSurge: <https://www.edsurge.com/n/2013-05-08-learn-to-code-code-to-learn>
- Schunk, D. (1997). *Teorías de aprendizaje* (2da ed.). México: Prentice-Hall.
- Stager, G. (2003). *En pro de los computadores*. Recuperado el 15 de Abril de 2014, de Eduteka: <http://www.eduteka.org/modulos/9/272/224/1>
- Stager, G. & Libow, S. (2013). *Invent to learn: Making, tinkering, and engineering in the classroom*. Torrance, CA: Constructing Modern Knowledge Press.
- Taborda, H. & Medina, D. (2012). *Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria*. Cali: Universidad Icesi.
- Triantafyllou, E. & Timcenk, O. (2013). *Applying Constructionism and Problem Based Learning for Developing Dynamic Educational Material for Mathematics At Undergraduate University Level*. The 4th International Research Symposium on Problem-Based Learning (págs. 1-8). Kuala Lumpur: IRSPBL.

Vigotsky, L. S. (1995). El desarrollo de los procesos psicológicos superiores.
Barcelona: Grijalbo Mondadori.

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

Woolfolk, A. E. (1999). *Psicología educativa* (7a ed.). México: Prentice Hall.

ANEXOS

ANEXO 1

Instrumento de medición del pensamiento computacional

Universidad Icesi Facultad de Derecho y Ciencias Sociales Departamento de Estudios Psicológicos PROYECTO: Impacto de Scratch en el desarrollo del Pensamiento Computacional		
--	---	---

Tareas para instrumento de medición de Pensamiento Computacional

Nombre: _____	Edad: _____
Fecha: _____	Grado: _____

1. Conociendo el tamaño del ratón

Lee la siguiente historia y contesta las preguntas que aparecen luego.

“Carlos y Jorge son hermanos, ellos saben que en su casa hay un ratón, pero nunca lo han visto. Saben que sólo hay un ratón y Carlos piensa que es un ratón muy grande, pero Jorge piensa que es un ratón muy pequeño”.

Un día, los niños deciden alimentar al ratón. Tienen dos cajas: una con una puerta grande y otra con una puerta pequeña.

¿En cuál caja deben poner la comida para asegurarse de que el ratón pueda entrar, comer y volver a salir por la puerta?

¿Qué te hace estar seguro de que esa caja sirve?

¿Qué pasaría si usan la otra caja?

“Otro día, Carlos y Jorge quieren averiguar si el ratón es grande o es pequeño. Ahora no les preocupa alimentarlo, sino conocer su tamaño. Entonces deciden poner comida en una de las cajas. Si en la mañana la comida ha desaparecido, entonces podrán saber si el ratón es grande o es pequeño”.

¿En cuál caja deben poner la comida si quieren averiguar si el ratón es grande o es pequeño?

¿Qué te hace estar seguro de que esa caja sirve?

¿Qué pasaría si ponen el alimento en la otra caja?



Fundación
Gabriel Piedrahita Uribe



2. Dibuja y ordena los objetos

Mira los siguientes 4 objetos.



balón



lápiz



libro



oso

Dibújalos todos dentro del cuadro, de acuerdo con las siguientes instrucciones:

El balón está arriba del libro

El libro está al lado del oso

☐ El balón está abajo del lápiz

Dibújalos otra vez, con estas instrucciones:

El oso está al lado del lápiz

El libro está abajo del balón

El balón está al lado del oso





3. Organiza los animales

En este cuadro hay un grupo llamado nombres de animales:

Nombres de animales
caballo, oso, abeja, tigre, piraña, perro, zancudo, serpiente, mosca, lombriz, ballena, águila, gato, pingüino, rana, lobo, delfín, vaca, león, oveja, búho, cocodrilo.

Divide los animales en tres grupos y escribe cada grupo de animales dentro de los cuadros de abajo. Debes poner un nombre diferente a cada grupo, que identifique los animales que pusiste ahí.

1	2	3

4. Gánate los puntos

A continuación, hay un fragmento del poema "Pastorcita" de Rafael Pombo. Si buscas en el texto puedes ganar puntos así:

- Por cada palabra que tenga las **últimas tres letras iguales** a otra palabra, **ganas 5 puntos**.
- Por cada palabra que comienza con **P** y que termina con **a**, **ganas 1 punto**.

Subraya las palabras con las que ganaste puntos y anota al frente el número de puntos ganados, tal como está en el ejemplo.

Pastorcita	Puntos
<u>Pastorcita</u> <u>perdió</u> sus <u>ovejas</u>	1 + 5
¡y <u>quién</u> sabe por <u>dónde</u> <u>andarán</u> !	
-No te enfades, que oyeron tus <u>quejas</u>	5
y ellas mismas bien pronto <u>vendrán</u> .	
Y no <u>vendrán</u> <u>solas</u> , que <u>traerán</u> sus <u>colas</u> ,	
Y <u>ovejas</u> y <u>colas</u> <u>gran</u> <u>fiesta</u> <u>darán</u> .	
Pastorcita se queda dormida,	
Y <u>soñando</u> las oye <u>balar</u> .	
Se <u>despierta</u> y las llama <u>enseguida</u> ,	
Y <u>engañada</u> se <u>tiende</u> a <u>llorar</u> .	
No <u>llores</u> , <u>pastora</u> , que <u>niña</u> que <u>llora</u>	
Bien pronto la oímos <u>reír</u> y <u>cantar</u> .	
Total Puntos:	





5. Completa los códigos

5.1. Llena los espacios en blanco con los códigos correspondientes a cada letra, tal como se indica en el ejemplo

A		B		C		D	
X	O	O	O	X	X	O	X

A	B	C	D	A	B
X	O	X	X	O	X
A	B	C	D	A	B
X	O	O	O		
A	B	C	D	A	B
X	O				
A	B	C	D	A	B
X	O				
A	B	C	D	A	B
A	B	C	D	A	B

¿Cuál fue el orden que seguiste para llenar los espacios? Señala con flechas dentro del cuadro

¿Qué les recomendarías a tus amigos para completar más fácilmente este cuadro?





5.2. Llena los espacios en blanco con los códigos correspondientes a cada letra, tal como se indica en el ejemplo:

A		B		C		D	
X	O	O	O	X	X	O	X

A		B		C		D		A		B	
X	O	O	O	X	X	O	X	X	O		
D		A		B		C		D		A	
O	X	X	O	O	O	X	X				
C		D		A		B		C		D	
X	X	O	X	X	O						
B		C		D		A		B		C	
O	O	X	X								
A		B		C		D		A		B	
X	O										
D		A		B		C		D		A	

¿Cuál fue el orden que seguiste para llenar los espacios? Señala con flechas dentro del cuadro.

¿Qué les recomendarías a tus amigos para completar más fácilmente este cuadro?



ANEXO 2

Rejilla de observación de clase (ISTE, 2012).

Fundación Gabriel Piedrahita Uribe		Observación de clases				Formato Original: "Classroom Observation Tool v3.1.1e CT" © ISTE 2012, 2013	
Introducir datos en las casillas o celdas sombreadas. Haga clic en los botones para definir los datos							
		Almacenar Datos		Limpiar Datos		Limpiar Formulario	
Datos	Click para insertar Fecha	<mm/dd/yy>	Grado	<Notas a las condiciones de la Institución>			
	Proyecto o Actividad de Clase	<name or code>	Asignatura	No aplica			
	I.E.	<name or code>	# Estudiantes	<enter #>			
	Observador	2. Andrés Ávila	# Dispositivos Digitales	<enter #>			
	Profesor	<name or code>	Estudiantes por Dispositivo	#¡VALOR!			
Click para definir Hora inicial de Observación		Waiting Data		Click para definir Hora final de		Waiting Data	
No hay Observación en Progreso -- Click 'Iniciar período de observación' a Iniciado							
Rol del Docente en el aula	Magistral	☐	Agrupación en el aula de los Estudiantes		<Notas a Roles/Grupos>		
	Guía - Facilitador	☐	Individual	☐			
	Modelador	☐	Parejas	☐			
	Exposición de Casos	☐	Grupos de tres o más	☐			
	Moderador de debates	☐	Otro tipo de agrupación	☐			
	Otro Rol	☐		☐			
No Observation in Progress -- Click 'Start Observation Period' to Begin							
Actividades de Aprendizaje	Elaborar Presentaciones	☐	WebQuest	☐	<Notas a las actividades>		
	Búsquedas Efectivas	☐	Test-Evaluación en Línea	☐			
	Evaluación de Fuentes	☐	Estadísticas - Gráficos de Datos	☐			
	Uso de Simuladores	☐	Alfabetismo en Medios	☐			
	Análisis de Información	☐	Foros de Discusión	☐			
	Organización de Información	☐	Other activity	☐			
No Observation in Progress -- Click 'Start Observation Period' to Begin							
Valoración	Uso de TIC en el aula	No Entro	<Notas de valoración del compromiso del estudiante>				
	Estudiantes no comprometidos	<enter #>					
	Comprometido %	#¡VALOR!					
No Observation in Progress -- Click 'Start Observation Period' to Begin							
TIC usadas	Hardware usado por:	Docente	Estudiante	Software usado por:	Docente	Estudiante	<Notas sobre las TIC>
	Calculadora	☐	☐	Análisis de Datos	☐	☐	
	Computador	☐	☐	Comunicación por Internet	☐	☐	
	Cámara Digital	☐	☐	Evaluación/Activ. Digitales	☐	☐	
	Sensores Digitales	☐	☐	Gráficos	☐	☐	
	Tablero Digital	☐	☐	LMS (Moodle)	☐	☐	
	Proyector	☐	☐	Esquema /Mapa Conceptual	☐	☐	
	Clickers	☐	☐	Editor Multimedia	☐	☐	
	Tablets	☐	☐	Simulaciones	☐	☐	
	Videoconferencias	☐	☐	Editor de Texto	☐	☐	
	Otros	☐	☐	Internet Información	☐	☐	
	Otros	☐	☐	Otro	☐	☐	

NETS para Estudiantes			Nivel	Uso de Computación
1. Creatividad	Aplican el conocimiento existente para generar nuevas ideas, productos o procesos.	No apl	Formular problemas de manera que permitan usar computadores y otras herramientas para solucionarlos.	No apl
	Crean trabajos originales como medios de expresión personal o grupal.	No apl		
	Usan modelos y simulaciones para explorar sistemas y temas complejos.	No apl		
	Identifican tendencias y prevén posibilidades.	No apl		
2. Comunicación	Interactúan, colaboran y publican con sus compañeros, con expertos o con otras personas, empleando una variedad de medios y de formatos.	No apl	Organizar datos de manera lógica y analizarlos.	No apl
	Comunican efectivamente información e ideas a múltiples audiencias, usando una variedad de medios y de formatos.	No apl		
	Desarrollan una comprensión cultural y una conciencia global mediante la vinculación con estudiantes de otras culturas.	No apl		
	Participan en equipos que desarrollan proyectos para producir trabajos originales o resolver problemas.	No apl		
3. Investigación	Planifican estrategias que guíen la investigación.	No apl	Representar datos mediante abstracciones, como modelos y simulaciones.	No apl
	Ubican, organizan, analizan, evalúan, sintetizan y usan éticamente información a partir de una variedad de fuentes.	No apl		
	Evalúan y seleccionan fuentes de información y herramientas digitales para realizar tareas específicas, basados en criterios.	No apl		
	Procesan datos y comunican resultados.	No apl		
4. Pensamiento crítico	Identifican y definen problemas auténticos y preguntas significativas para investigar.	No apl	Automatizar soluciones mediante pensamiento algorítmico (una serie de pasos ordenados).	Aplica
	Planifican y administran las actividades necesarias para desarrollar una solución o completar un proyecto.	No apl		
	Reúnen y analizan datos para identificar soluciones y/o tomar decisiones informadas.	No apl		
	Usan múltiples procesos y diversas perspectivas para explorar soluciones alternativas.	No apl		
5. Ciudadanía digital	Promueven y practican el uso seguro, legal y responsable de la información y de las TIC.	No apl	Identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de Generalizar y transferir ese proceso de soluciones de problemas a una gran diversidad de situaciones.	No apl
	Exhiben una actitud positiva frente al uso de las TIC para apoyar la colaboración, el aprendizaje y la productividad.	No apl		
	Demuestran responsabilidad personal para aprender a lo largo de la vida.	No apl		
	Ejercen liderazgo para la ciudadanía digital.	No apl		
6. Conceptos	Entienden y usan sistemas tecnológicos de Información y Comunicación.	No apl		
	Seleccionan y usan aplicaciones efectivas y productivamente.	No apl		
	Investigan y resuelven problemas en los sistemas y las aplicaciones.	No apl		
	Transfieren el conocimiento existente al aprendizaje de nuevas tecnologías de Información y Comunicación (TIC).	No apl		
<Notas a los estándares				
<Standards notes>				
Tiempo de	Uso de TIC	Por profesor	Por Estudiante	<Notas al tiempo de aprendizaje
	(Tech timer off)	0	0	
	Tiempo sin usar las TIC	0:00:00	0:00:00	
	Promedio de Uso de las TIC	0%	0%	
	Duración Obs.:		0:03:05	
Do not delete any of the next 50 rows below.				
Doing so will delete cells that ICOT needs for data recording.				
Click the Data Preview button to see the hidden data recording cells.				

ANEXO 3

Plantilla para analizar problemas (López, 2013b)⁶

ANÁLISIS DE PROBLEMAS	
	Formular el problema: _____ _____ _____ _____ _____ _____ _____ _____
	Resultados esperados: _____ _____ _____ _____ _____ _____ _____
	Datos disponibles: _____ _____ _____ _____ _____

⁶ Descargar de: http://www.eduteka.org/pdfdir/AnalisisProblemas_PLANTILLA.pdf



Restricciones: _____



Procesos necesarios (en pseudocódigo): _____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____
17. _____

ANEXO 4

Categorías de análisis

1. DIMENSIÓN HABILIDADES DE PENSAMIENTO ALGORÍTMICO (PA)			
Habilidades evidenciadas a <i>nivel individual</i> que permiten implementar un conjunto de pasos ordenados para resolver un problema o lograr un objetivo.			
CATEGORÍAS	CÓD.	CONCEPTOS	DESCRIPCIONES
1.1 Conceptos del PA relacionados con Analizar Problemas DEFINICIÓN Según Schunk (1997), analizar un problema se entiende como especificar los resultados que se desean obtener (metas), identificar la información disponible (organización de datos), establecer las restricciones que impone el problema y definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (operaciones).	1.1.1	Descomposición funcional	Descompone un problema en varios más sencillos de resolver con un enfoque de arriba-abajo (Joyanes, 2001).
	1.1.2	Metas	Establece los resultados que se esperan; para ello, identifica la información relevante, ignora los detalles sin importancia, entiende los elementos del problema y activa el esquema correcto que permita comprenderlo en su totalidad (Woolfolk, 1999).
	1.1.3	Organización de datos	Identifica los datos que se presentan en el enunciado de un problema y el nivel de conocimiento que posee en el ámbito del problema que está tratando de resolver. Elimina los datos inútiles.
	1.1.4	Restricciones	Identifica aquello que está permitido o prohibido hacer y/o utilizar para llegar a una solución
	1.1.5	Procesos	Determina los pasos sucesivos que permiten llegar a los resultados esperados, a partir de los datos disponibles y teniendo en cuenta las restricciones planteadas.
	1.1.6	Comprensión lingüística	A partir del enunciado del problema planteado, formula en sus propias palabras cuál es el problema a resolver.
	1.1.7	Planificación cognitiva	Registra en una plantilla de análisis tanto su comprensión del problema como los resultados esperados, los datos disponibles, las restricciones impuestas y los procesos necesarios para solucionar el problema. Demanda planificar, formular hipótesis y anticipar qué sucederá.
1.2 Conceptos del PA relacionados con Diseñar una solución DEFINICIÓN Diseñar una solución se entiende como la representación mental o gráfica, mediante símbolos geométricos o dibujos, de la secuencia lógica de los procesos.	1.2.1	Planificación cognitiva	Repite los pasos que han sido previstos en la plantilla de análisis. Selecciona y decide las operaciones a efectuar. Demanda planificar, formular hipótesis y anticipar qué sucederá.
	1.2.2	Procesos	Incluye en el diseño los pasos identificados en la fase de Análisis.
	1.2.3	Estructuras de control	Incluye tres estructuras de control básicas: secuenciales, repetitivas y condicionales.
	1.2.4	Fondos	Crea los fondos del escenario y determina cuál será su comportamiento.
	1.2.5	Objetos	Tiene en cuenta que los objetos tienen apariencia, sonidos y comportamientos. Que los comportamientos deben programarse.

1.1 Conceptos del PA relacionados con Implementar una solución DEFINICIÓN Traducción a un lenguaje de programación como Scratch del diseño de la solución.	1.3.1	Instrucciones repetitivas	Estructura de control que permite repetir un conjunto de instrucciones un determinado número de veces.
	1.3.2	Instrucciones condicionales	Crea algoritmos que utilizan instrucciones condicionales (<i>estructuras de control</i>)
	1.3.3	Instrucciones secuenciales	Crea algoritmos que utilizan instrucciones secuenciales (<i>estructuras de control</i>)
	1.3.4	Eventos	Acciones generadas por el usuario del programa o por otra parte u objeto del programa (Resnick)
	1.3.5	Automatización	Generalizar y parametrizar una solución para poder hacer reuso de ella en otro contexto (National Research Council, 1999).
	1.3.6	Paralelismo	Ejecutar dos o más pilas de instrucciones al mismo tiempo, dos o más hilos independientes que se ejecutan en paralelo.
	1.3.7	Variables	Organización básica de datos mediante la clasificación y almacenamiento de números o cadenas alfanuméricas (National Research Council, 1999).
	1.3.8	Comandos	Crea o utiliza en los algoritmos comandos de Scratch correspondientes a las categorías de movimiento, apariencia, sonido, lápiz, sensores, operadores o variables.
	1.3.9	Reuso	Aprovechar bloques de código de un objeto en el mismo objeto o en otros. También aprovechar objetos existentes para modificarlos con el editor de pinturas y generar nuevos disfraces.
	1.3.10	Interfaz	Opciones de menú, uso del editor de pinturas, crear/importar objetos, crear disfraces.
1.2 Conceptos del PA relacionados con Depurar una solución DEFINICIÓN Hace referencia a probar, validar y refinar una solución (Stager, 2003).	1.4.1	Depuración	Selecciona algoritmos programados previamente y genera cambios y/o refinamientos sobre los mismos.
	1.4.2	Planificación cognitiva	Realiza cambios a los pasos que ha previsto en la plantilla. Demanda planificar, formular hipótesis y anticipar qué sucederá.
	1.4.3	Optimización	Experimentar nuevas formas de soluciones que hagan más fácil de realizar la tarea.
	1.4.4	Prueba y validación	Ejecutar los programas elaborados para validar que funcionan de acuerdo a lo planificado.

2. DIMENSIÓN INTERACCIONES			
Alude a las formas de interacción tanto del docente con sus estudiantes, como de estudiantes unos con otros (pares).			
CATEGORÍAS	CÓD.	CONCEPTOS	DESCRIPCIONES
2.1 Interacción docente-estudiantes. DEFINICIÓN Alude a las formas de interacción del docente que promueven habilidades de solución de problemas.	2.1.1	Instruir	Explicaciones y ejemplificaciones magistrales que el docente provee a todo el grupo para que estos desarrollen la actividad de aprendizaje propuesta. Retoma la actividad de la clase anterior como punto de partida para presentar un nuevo tema. Utiliza el tablero para los pasos que el estudiante debe realizar en Scratch. Acompaña las explicaciones con ejemplos cotidianos que faciliten la comprensión. Presenta aclaraciones sobre las instrucciones que ha dado.
	2.1.2	Apoyar	Explicaciones, ejemplificaciones y preguntas que el docente provee de manera individual a los estudiantes que lo requieran para desarrollar la actividad de aprendizaje propuesta. Realiza preguntas a los estudiantes que les permita recordar algoritmos y procedimientos trabajados en clases anteriores. Formula preguntas a los estudiantes para que ellos identifiquen el uso de comandos en Scratch antes de explicar su uso. Promueve la toma de decisiones por parte de los estudiantes. Pregunta al estudiante por qué ha utilizado determinada secuencia de comandos en Scratch.
	2.1.3	Evaluar	El docente revisa el trabajo realizado por los estudiantes y asigna una calificación. Promueve el orden en el aula y asigna puntos negativos a quienes hacen desorden.
2.2 Interacciones entre pares. DEFINICIÓN Hace referencia a los diálogos y a las acciones que se dan entre pares, las cuales favorecen el desarrollo de habilidades de solución de problemas cuando trabajan con Scratch.	2.2.1	Resolver	El estudiante resuelve el problema de su compañero sin explicarle cómo lo hizo.
	2.2.2	Explicar	El estudiante resuelve el problema de su compañero explicándole verbalmente cómo lo hizo.
	2.2.3	Preguntar	El estudiante le formula preguntas a su compañero de manera que éstas lo induzcan a resolver por sí mismo el problema.
	2.2.4	Observar	El estudiante observa cómo su compañero resuelve el problema propuesto.
	2.2.5	Dialogar	Dos o más estudiantes conversan sobre algún aspecto de la tarea o de la solución del problema propuesto.
	2.2.6	Ejemplificar	El estudiante presenta un ejemplo a su compañero para explicarle cómo resolver el problema.
	2.2.7	Colaborar	Dos o más estudiantes se unen y trabajan en un computador para intentar resolver el problema

ANEXO 5

Frecuencia de cada categoría en los datos cualitativos

1. Conceptos de Pensamiento Algorítmico (PA)	165
1.1 Conceptos de PA relacionados con Análisis	56
1.1.1 Descomposición funcional	0
1.1.2 Metas	9
1.1.3 Organización de datos	9
1.1.4 Restricción	10
1.1.5 Procesos	9
1.1.6 Comprensión lingüística	18
1.1.7 Planificación cognitiva	1
1.2 Conceptos de PA relacionados con Diseño	26
1.2.1 Planificación cognitiva	11
1.2.2 Procesos	5
1.2.3 Estructuras de control	1
1.2.4 Fondos	0
1.2.5 Objetos	9
1.3 Conceptos de PA relacionados con Implementación	61
1.3.1 Instrucciones repetitivas	17
1.3.2 Instrucciones condicionales	0
1.3.3 Instrucciones secuenciales	16
1.3.4 Eventos	1
1.3.5 Automatización	2
1.3.6 Paralelismo	4
1.3.7 Variables	1
1.3.8 Comandos	7
1.3.9 Reuso	3
1.3.10 Interfaz	10
1.4 Conceptos de PA relacionados con Depuración	22
1.4.1 Depuración	7
1.4.2 Planificación cognitiva	0
1.4.3 Optimización	1
1.4.4 Prueba y validación	14
2. Interacciones	101
2.1 Interacción docente-estudiantes.	78
2.1.1 Instruir	39
2.1.2 Apoyar	20
2.1.3 Evaluar	19
2.2 Interacción estudiante-estudiante.	23
2.2.1 Resolver	1
2.2.2 Explicar	5
2.2.3 Preguntar	3
2.2.4 Observar	3
2.2.5 Dialogar	8
2.2.6 Ejemplificar	2
2.2.7 Colaborar	1
TOTAL:	266

ANEXO 6

Plantilla de Diseño (López, 2011; p 81)



ESCENARIO

Fondo 1



Nombre: _____

Fondo 2



Nombre: _____

Fondo 3



Nombre: _____

¿Qué hace el escenario (programas)? : _____

Bloques de Scratch a utilizar en el escenario (consultar el Anexo 1) : _____

Secuencia de instrucciones para el escenario en formato Seudocódigo:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____

si te hace falta espacio, puedes continuar en el cuaderno



Sonido 1 del escenario: _____

Sonido 2 del escenario: _____



OBJETOS

OBJETO 1

Nombre del objeto 1: _____

Imagen del objeto 1: _____

Nombre del disfraz 1: _____

Nombre del disfraz 2: _____

Nombre del disfraz 3: _____

Nombre del disfraz 4: _____

Nombre del disfraz 5: _____



Sonido 1 del objeto 1: _____

Sonido 2 del objeto 1: _____

¿Qué hace el objeto 1 (programas)? : _____

Bloques de Scratch que se utilizan en el objeto 1 (consultar el Anexo 1): _____

Secuencia de instrucciones para el objeto 1 en formato Seudocódigo:

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____