

**METODOLOGÍA PARA LA ENSEÑANZA INICIAL DE MACROINSTRUCCIONES
Y PROCESOS LÓGICO- MATEMÁTICOS A NIÑOS DE GRADO QUINTO EN
INSTITUCIONES OFICIALES DE EDUCACIÓN BÁSICA PRIMARIA**



**Proyecto de Tesis de Grado para optar por el título de
Magister en Gestión, Aplicación y Desarrollo de Software**

CARLOS ANDRÉS PALMA SUÁREZ

**Director Ph.D. ROMÁN EDUARDO SARMIENTO PORRAS
Línea de Investigación E-Learning y materiales educativos computarizados
Grupo de Investigación Preservación e Intercambio Digital de Información y
Conocimiento – PRISMA**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB
FACULTAD DE INGENIERÍA DE SISTEMAS
MAESTRÍA EN GESTIÓN, APLICACIÓN Y DESARROLLO DE SOFTWARE
BUCARAMANGA, JULIO DE 2014**

AGRADECIMIENTOS

Antes que todo, quiero dar las gracias a mi Dios y a María Santísima, quienes me dieron la fortaleza espiritual para poder poner a rodar esta idea y emprender este proyecto. Doy gracias también a mi familia: mi esposa Adriana y mis hermosos hijos Sergio Andrés y Carlos Eduardo, mis amores y motores, sin ellos mi proyecto (y mi vida) no tendrían rumbo. A mis padres, Sixto y Lucila, quienes me han servido de soporte y consuelo en los momentos que más lo he necesitado.

Deseo expresar mis agradecimientos al Dr. Román Eduardo Sarmiento, mi director de proyecto, quien ha sabido orientarme, explicarme y exigirme en esta faceta investigativa, a la Universidad Autónoma de Bucaramanga, al Ing. Daniel Arenas por brindarnos este espacio para crecer como investigadores y como personas, al Dr. Jorge Andrick Parra, con quien aprendí muchas cosas respecto a cómo se realiza una buena investigación, y a mis compañeros de clase, Tania, Rosaura, Julián, Vermen, y muchos otros con quienes compartí y construí lazos muy fuertes que me permitieron crecer en lo personal y en lo académico.

De igual manera, quiero dar mi agradecimiento a la Licenciada Margarita Cruz Lozano por permitirme presentar e implementar este proyecto en el Colegio San Juan de Girón Sede B, a la Coordinadora de la Jornada Mañana Sede B, Ligia Gil Pinzón y a todo su equipo administrativo, a los docentes Carmen Alicia Albarracín y Carlos Buitrago, a todo el personal general del colegio, docentes, seguridad y servicios, y especialmente a los estudiantes de los grados 5-1 y 5-2 que participaron en este proyecto. Espero que este proyecto les haya servido para abrir sus mentes y para crecer no solo en cuerpo sino en mente.

“Most teachers waste their time by asking questions that are intended to discover what a pupil does not know, whereas the true art of questioning is to discover what the pupil does know or is capable of knowing.”

Albert Einstein

Citado por Moszkowski en Conversaciones con Einstein (1920)

CONTENIDO

1.	PLANTEAMIENTO DEL PROBLEMA	11
1.1	INTRODUCCIÓN	11
1.2	CONTEXTO ACTUAL DEL PROBLEMA	12
1.3	ANTECEDENTES	14
1.4	PROBLEMA DE INVESTIGACIÓN	16
1.5	HIPÓTESIS DE LA INVESTIGACIÓN	19
1.6	OBJETIVOS DE LA INVESTIGACIÓN	20
1.6.1	Objetivo General	20
1.6.2	Objetivos Específicos	20
1.7	JUSTIFICACIÓN DE LA INVESTIGACIÓN	21
1.8	LIMITACIONES	22
1.9	DEFINICIÓN DE TÉRMINOS	23
2.	MARCO TEÓRICO	25
2.1	ESTADO DEL ARTE	25
2.2	EXPERIENCIAS DE ENSEÑANZA DE PROGRAMACIÓN PARA NIÑOS Y JÓVENES	26
2.2.1	Experiencias internacionales	26
2.2.2	Experiencias locales	32
2.2.3	Determinación y clasificación de técnicas	32
2.3	TEMÁTICAS Y CONTEXTOS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS	42
2.4	HERRAMIENTAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN EN NIÑOS	46
2.5	TÉCNICAS DE APLICACIÓN DE LA ENSEÑANZA DE LA PROGRAMACIÓN EN EL AULA	50
2.6	MODELOS DE EVALUACIÓN	53
3.	MARCO METODOLÓGICO	55
3.1	DESCRIPCIÓN DEL MÉTODO	55
3.2	ESTUDIO COMPARATIVO DE TÉCNICAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS	56
3.2.1	Uso de videojuegos para la enseñanza	57
3.2.2	Creación de proyectos con temática fija	59

3.2.3	Creación de entornos para generación de situaciones	60
3.2.4	Creación de videojuegos	61
3.2.5	Uso de herramientas computacionales como apoyo a la resolución de problemas	62
3.2.6	Generación de torneos o competencias	62
3.2.7	Comparativo de técnicas	63
3.3	ESTUDIO COMPARATIVO DE HERRAMIENTAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS	66
3.3.1	Revisión de Piktomir	69
3.3.2	Revisión de Lightbot	69
3.3.3	Revisión de Manufactoria	70
3.3.4	Revisión de Blockly	70
3.3.5	Comparación de herramientas	71
3.4	GENERACIÓN DE PLAN CURRICULAR PARA EL ÁREA DE TECNOLOGÍA	73
3.4.1	Objetivos	73
3.4.2	Contenido	74
3.4.3	Actividades	75
3.4.4	Recursos a utilizar	76
3.4.5	Compilación de elementos del plan	76
3.4.6	Forma de evaluación	80
3.5	APLICACIÓN DEL CURSO Y DE LAS PRUEBAS	82
3.5.1	Diseño de los instrumentos de medición	83
3.5.2	Aplicación de prueba inicial y recopilación de datos	87
3.5.3	Circunstancias que rodearon la aplicación del curso	87
3.5.4	Aplicación de prueba final y recopilación de datos	88
4.	RESULTADOS Y ANÁLISIS DE LAS PRUEBAS DE DESEMPEÑO	91
4.1	RESULTADOS DE LAS CALIFICACIONES OBTENIDAS EN LAS PRUEBAS	91
4.2	RESULTADOS POR VARIABLE	92
4.3	ANÁLISIS ESTADÍSTICO	96
4.3.1	Fiabilidad de las variables	96
4.3.2	Análisis de resultado de la calificación final	96
4.3.3	Análisis secundario por variable	100
4.4	RESULTADO DE LA REVISIÓN DE COMPETENCIAS	105

4.4.1	Propuesta de competencias a aplicar en un curso de tecnología referente al desarrollo del pensamiento algorítmico	107
5.	DISCUSIÓN Y CONCLUSIONES	113
5.1	HALLAZGOS EN LOS RESULTADOS OBTENIDOS	113
5.2	DISCUSIONES GENERALES	116
5.2.1	Uso de videojuegos como técnica para enseñar programación a niños	116
5.2.2	Enseñanza de la programación como refuerzo en la habilidad de interpretación de operaciones matemáticas	117
5.2.3	No afectación en la habilidad de ejecución de operaciones matemáticas	119
5.3	CONCLUSIONES	119
5.4	LIMITANTES Y RECOMENDACIONES PARA TRABAJO FUTURO	121
6.	BIBLIOGRAFÍA	123
	ANEXOS	133

LISTA DE TABLAS

Tabla 1. Cuadro comparativo de 18 experiencias de enseñanza de la programación a niños y jóvenes en el mundo	27
Tabla 2. Clasificación de técnicas utilizadas para la enseñanza de programación en niños y jóvenes	33
Tabla 3. Comparación de ventajas, desventajas y listado de herramientas por técnica	63
Tabla 4. Comparativo de evaluaciones de herramientas usadas para enseñanza de la programación	66
Tabla 5. Parámetros de evaluación de herramientas usadas para enseñanza de la programación	67
Tabla 6. Evaluación de herramientas para implementar la técnica de uso de videojuegos en la enseñanza de la programación	71
Tabla 7. Estructura de contenidos, actividades y recursos para implementar en el plan curricular	76
Tabla 8. Variables identificadas para medir en las pruebas pretest y final	84
Tabla 9. Tabulación de los valores de las calificaciones finales en las pruebas de los estudiantes de los dos grupos	91
Tabla 10. Tabulación de los valores de las variables medidas en las pruebas de los estudiantes de los dos grupos	93
Tabla 11. Resumen de resultados de los porcentajes de variables en las pruebas de los estudiantes de los dos grupos	95
Tabla 12. Resultado de correlación entre las variables y la calificación final mediante alfa de Cronbach	96
Tabla 13. Análisis estadístico mediante prueba T resultados pretest (Comparación G1 - G2)	97
Tabla 14. Análisis estadístico mediante prueba T resultados prueba final (Comparación G1 - G2)	98
Tabla 15. Análisis estadístico mediante prueba T (Comparación de calificaciones por grupos)	99
Tabla 16. Análisis estadístico mediante prueba T variables evaluadas en pretest (Comparación G1 - G2)	101

Tabla 17. Análisis estadístico mediante prueba T variables evaluadas en prueba final (Comparación G1 - G2)	102
Tabla 18. Análisis estadístico mediante prueba T (Comparación de variables por grupos)	103
Tabla 19. Estándares e indicadores existentes identificados para referenciar competencias	105
Tabla 20. Competencias propuestas por componente	107
Tabla 21. Logros propuestos para medición de competencias	108

LISTA DE FIGURAS

- Figura 1. Comparación de porcentajes según niveles de desempeño en el departamento de Santander y en el país en matemáticas, quinto grado, año 2013 13
- Figura 2. Comparación del desempeño por países en lectura, matemáticas y ciencias, año 2009 14
- Figura 3. Gráfico comparativo e histogramas de las calificaciones finales en las pruebas de los estudiantes de los dos grupos 92
- Figura 4. Gráfico comparativo de los porcentajes de variables en las pruebas de los estudiantes de los dos grupos 95

LISTA DE ANEXOS

Anexo 1.	Formato de carta de consentimiento presentada a los padres de familia	134
Anexo 2.	Fotografías tomadas durante la aplicación del curso en el Colegio San Juan de Girón	135
Anexo 3.	Propuesta de plan curricular enfocado en competencias	138

1. PLANTEAMIENTO DEL PROBLEMA

1.1 INTRODUCCIÓN

El proceso de aprendizaje en los niños se puede determinar por el avance que pueda desarrollar el niño en cuanto a habilidades específicas, como la motricidad gruesa, fina, social, emocional, lenguaje y la cognición (Kliegman, Stanton, Schor, St. Geme III, & Behrman, 2011). Cada una de estas habilidades es generada mediante secuencias de cambios que guían hasta lograr capacidades concretas, dentro de las que se encuentran la lógica y la matemática, que se encuentran dentro del desarrollo cognitivo. La etapa infantil se ha vuelto un objeto de estudio muy importante, el cual ha generado muchas investigaciones sobre las que se han elaborado diferentes teorías de aprendizaje. Jean Piaget – citado en Albornoz (2007) – ha afirmado que los niños entre 7 y 11 años entran en un estadio de desarrollo cognitivo denominado “operatorio concreto” en el cual desarrollan estas habilidades lógico-matemáticas dado a que su naturaleza del pensamiento cambia radicalmente; tomando los símbolos el lugar de objetos y acciones de una manera cada vez mayor (Kliegman, Stanton, Schor, St. Geme III, & Behrman, 2011). Para los niños que terminan su educación básica primaria, los procesos de razonamiento y desarrollo lógico-matemáticos se convierten en competencias muy importantes que deben adquirir con el fin de comprender y resolver problemas de diversas complejidades para su nivel, realizando procesos de formulación, comparación y ejercitación de procedimientos, y logrando con ello prepararlos también para “el manejo de calculadoras, el uso de hojas de cálculo, la elaboración de macroinstrucciones y aun para la programación de computadores.” (Ministerio de Educación Nacional, 2003).

Dado el creciente impacto en la apropiación y uso de las tecnologías de la información y la comunicación – TIC, se ha generado la imperiosa necesidad de capacitar personas para que adquieran habilidades particularmente en la comprensión y desarrollo de sistemas software, viéndose esto reflejado en la creación de proyectos y estrategias tanto nacionales – como el programa Talento Digital del Ministerio de las TIC (Ministerio de Tecnologías de la Información y las

Comunicaciones, 2011) –, como internacionales – como la iniciativa Hour to Code (Code.org, 2013) –. Lograr establecer una forma de enseñar los primeros procesos lógicos y matemáticos referentes a secuenciación y manipulación de información en los niños, podría generar en ellos la adquisición de habilidades en la elaboración de secuencias de instrucciones que son la base fundamental en el desarrollo de diversos sistemas software en el futuro, además de contribuir de alguna manera significativa con el mejoramiento de las competencias matemáticas que deben adquirir los niños de estas edades.

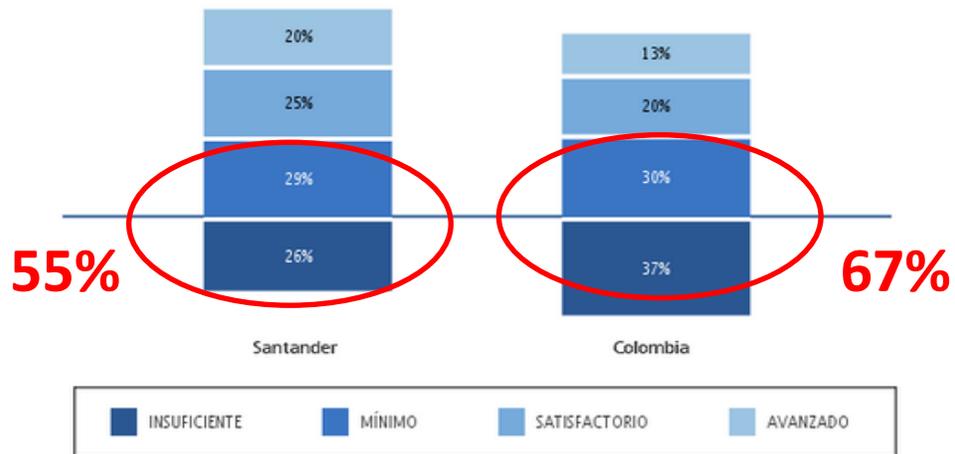
Esta tesis generó un modelo inicial de trabajo en el que logró enseñar a niños pertenecientes particularmente al grado 5° de educación básica primaria (con rangos estimados de edades entre 9 y 11 años) la elaboración de macroinstrucciones a partir del entendimiento de procesos lógicos y matemáticos utilizando para ello un conjunto de herramientas software de apoyo y complementándolo con la aplicación de técnicas adecuadas de enseñanza de estos temas en particular. Con ello se pretendió mejorar las competencias matemáticas en los estudiantes de primaria, pero también se podría a futuro ayudar a generar otras competencias que se planteen en nuevos modelos de formación tecnológica en el nivel básico de educación utilizando la comprensión y el desarrollo de sus primeros algoritmos.

1.2 CONTEXTO ACTUAL DEL PROBLEMA

Las competencias que tienen que ver con la ejecución de procesos lógico–matemáticos en los niños entre 7 a 11 años han sido trabajadas desde hace tiempo en diversas instituciones educativas de educación básica primaria, basados en los estándares de competencias establecidos por el Ministerio de Educación Nacional (2003). Sin embargo, durante los últimos años, como lo demuestran los resultados de las pruebas SABER de los años 2002 – 2005 – 2009 – 2012 y 2013 en el área de matemáticas en el nivel 5° de educación básica primaria (Instituto Colombiano para la Evaluación de la Educación - ICFES, 2013), se ha visto un deterioro notable en la calidad educativa en el área de matemáticas. Los últimos resultados son evidentes: revisando los niveles de desempeño en el

área de matemáticas en el departamento de Santander y presentando una comparación con los resultados generales del país, se observa claramente que no se logra alcanzar un nivel al menos satisfactorio en más del 50% de la población estudiantil, ni a nivel departamental ni a nivel nacional.

Figura 1. Comparación de porcentajes según niveles de desempeño en el departamento de Santander y en el país en matemáticas, quinto grado, año 2013



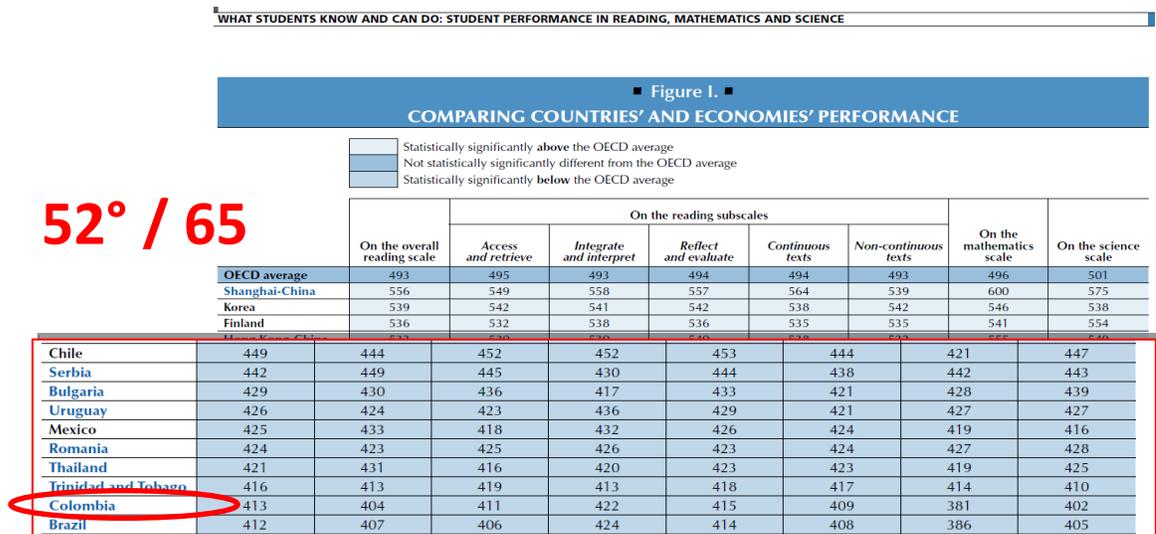
Fuente: (Instituto Colombiano para la Evaluación de la Educación - ICFES, 2013)

La deficiencia en estas áreas influye en la dificultad en adquirir nuevas competencias en los grados superiores de educación media. Este deterioro en la calidad se ve incluso reflejado en las últimas pruebas internacionales PISA, organizadas por la OECD (OECD Programme for International Student Assessment, 2010). Los resultados de estas pruebas muestran a Colombia en un preocupante nivel de desempeño académico en las áreas de lectura, matemáticas y ciencias, ocupando en el año 2009 el puesto 52 y en el año 2012 el puesto 62 entre 65 países. El ICFES realizó un estudio de los resultados de las pruebas del año 2009 (Instituto Colombiano para la Evaluación de la Educación - ICFES, 2010, pág. 32), encontrando que:

...el 70,6% de los alumnos no logra el desempeño mínimo establecido por PISA (nivel 2), en el cual las personas están en capacidad de participar activamente en la sociedad. El 20,3% de los estudiantes se ubicó en el nivel 2; el 7,5% en el 3; y sólo el 1,8% restante en los niveles 4, 5 y 6. Estos resultados son muy preocupantes, pues además de ser los más deficientes entre las tres áreas evaluadas, contrastan con los de Shanghái, Finlandia y

Corea, países en los que más de la mitad de los alumnos se clasificó por encima del nivel 3.

Figura 2. Comparación del desempeño por países en lectura, matemáticas y ciencias, año 2009



Fuente: (OECD Programme for International Student Assessment, 2010)

Esta deficiencia se evidencia por último en las falencias que presentan los estudiantes que logran terminan su educación secundaria e ingresan a la educación superior. Las dificultades académicas de algunos estudiantes, así como el poco gusto por el estudio, se convierten en algunas de las causas para la deserción y la repitencia académica (Ministerio de Educación Nacional, 2010), y esto conlleva a efectos más graves a futuro, como la pérdida económica que el gobierno y la sociedad acarrearían al presentarse este fenómeno de deserción (Pulido & Velasco, 2009). Adicionalmente, se presentaría una disminución en la productividad laboral debido al bajo nivel educativo y por ende conllevaría a un bajo crecimiento económico nacional (Cárdenas, 2001).

1.3 ANTECEDENTES

La deficiencia académica de los estudiantes no es un fenómeno que se presenta únicamente en Colombia, por el contrario, es más común de lo que se cree en otros países (Velez, Schiefelbein, & Valenzuela, 1994). Dentro de las causas que

generan un bajo rendimiento académico, se encuentra en el alto desinterés de niños para el aprendizaje, quienes ven en la falta de aplicación de nuevos ambientes en la educación una sensación de insatisfacción al trabajar con los viejos modelos educativos (Miranda, 2007). Este desinterés ha surgido debido a la persistencia de algunos docentes en el uso de metodologías tradicionales para la enseñanza sin uso de nuevos ambientes tecnológicos (Bauer & Kenton, 2005; Barrera Osorio & Leigh L., 2009). Muchos docentes se muestran renuentes a trabajar en nuevos ambientes que impliquen una apropiación de las tecnologías de la información y la comunicación (Miranda, 2007). Y no es para menos. Según un estudio realizado por Don Tapscott, plasmado en su libro *Grown Up Digital* (Tapscott, 2008), los niños y jóvenes en la actualidad están inmersos en este nuevo ambiente tecnológico, presentando rasgos no muy halagadores. Características que se presentan en algunos niños como una mayor ignorancia, una frecuente adicción a los videojuegos o a las redes sociales, irrespeto y falta de responsabilidad, inseguridad, violencia, entre otros, son varios de los rasgos que se evidencian en estos nuevos usuarios, quienes viven “conectados” con las nuevas tecnologías. A pesar de esta visión actual, muchos artículos y libros han demostrado desde hace tiempo que un adecuado uso de las nuevas tecnologías en el ámbito académico permite generar una mejora sustancial en la calidad educativa (Nickerson & Zodhiates, 1988; Eggers, 2005; Sancho Gil & et.al, 2006). Incluso Tapscott (2008), en su estudio, promueve una serie de técnicas para que los educadores puedan modificar sus metodologías tradicionales aplicando de una forma correcta el uso de las nuevas tecnologías.

Mejorar la calidad educativa se ha convertido en un propósito muy importante a cumplir en nuestro país, por todos los actores implicados en el proceso académico. Tanto es así que, como un objetivo misional propuesto por el Ministerio de Educación Nacional, se encuentra el mejoramiento en la calidad de la educación en todos los niveles académicos, mediante el fortalecimiento del desarrollo de competencias (Ministerio de Educación Nacional, 2013). Es por esto que no se deben escatimar recursos para incrementar la calidad educativa a un nivel mucho mayor que el actual. Sin embargo, una manera para contribuir a la solución de este gran problema no es encaminar únicamente a utilizar la tecnología para apoyar los procesos académicos, sino que también se ha buscado la enseñanza de la tecnología en sí, por lo que desde hace mucho tiempo se ha hecho hincapié en la enseñanza de la tecnología, particularmente de la enseñanza

de la programación en los niños. En el libro “Structure and Interpretation of Computer Programs”, se habla sobre la importancia de esta enseñanza, a tal punto de presentarla como un complemento de la matemática, dado que mientras “la matemática provee un marco de trabajo para tratar en forma precisa con nociones de tipo -qué-, la computación provee un marco de trabajo para tratar en forma precisa con nociones de tipo -cómo-” (Abelson, Sussman, & Sussman, 1996). Iniciativas en otros países como los proyectos Code.org (2013), o Scratch entre muchos otros, buscan que mediante la programación, los niños aprendan a pensar creativamente, razonar sistemáticamente y trabajar colaborativamente (MIT Media Lab, 2013). En palabras de muchos investigadores, la enseñanza de la programación puede hacer que los niños mejoren las habilidades de pensamiento y resolución de problemas. De acuerdo con Delgado y colaboradores (2013), “la programación de ordenadores aporta algo positivo y diferente a la formación de una persona: hábitos y conocimientos que tienen un cierto valor práctico en el día a día de, como mínimo, cualquier persona que viva en un entorno urbano del primer mundo”.

En Colombia, dentro de los lineamientos curriculares y los estándares de competencias para el nivel de primaria que ha planteado el Ministerio de Educación, existe un conjunto de orientaciones generales para la educación en tecnología, los cuales brindan un marco de trabajo para orientar la enseñanza de la importancia tecnológica desde temprana edad. Según el Ministerio de Educación Nacional (2008), la alfabetización tecnológica “busca que individuos y grupos estén en capacidad de comprender, evaluar, usar y transformar objetos, procesos y sistemas tecnológicos, como requisito para su desempeño en la vida social y productiva.”

1.4 PROBLEMA DE INVESTIGACIÓN

Uno de los aspectos que abarca la enseñanza tecnológica está en la comprensión de los procesos lógicos como fases secuenciales de operaciones que transforman recursos con el fin de cumplir con ciertos objetivos y lograr un resultado esperado, necesitando para ello tomar decisiones que asocian “propósitos, recursos y

procedimientos para la obtención de un producto o servicio” (Ministerio de Educación Nacional, 2008). Esta definición concuerda en buena parte con uno de los cinco procesos generales de la actividad lógica-matemática, según se presenta en los Estándares Básicos de Competencias en Matemáticas (Ministerio de Educación Nacional, 2003). Según estos estándares, la formulación, comparación y ejercitación de procedimientos, es un proceso en el cual se busca enseñar a los estudiantes a construir y ejecutar en forma precisa un conjunto de procedimientos secuenciales y lógicos llamados *algoritmos*.

Se podría afirmar entonces que la enseñanza de la formulación de procesos secuenciales en los niños es un tema que atañe, no solo al área de conocimiento matemático sino también al área tecnológica. Una propuesta de iniciar en los niños de 5° de primaria la enseñanza de elaboración de instrucciones secuenciales no suena ahora tan descabellada, sino más bien, se vuelve una herramienta complementaria que ayudaría a generar y mejorar competencias en las áreas de matemáticas y de tecnología.

Existen varios recursos digitales en español en diversos repositorios (como la mediateca del Banco de Objetos de Aprendizaje del portal Colombia Aprende) que hablan sobre la integración de la tecnología y la enseñanza de procesos lógico-matemáticos, sin embargo se enfocan de una forma diferente:

- Se encuentran diversos objetos de aprendizaje para enseñar los primeros conceptos lógicos para niños entre 4 y 7 años, como secuenciación y seriación, pero no se enfocan en la generación de instrucciones, dada su corta edad para entender estos temas.
- Se encuentran diversos objetos de aprendizaje para enseñar conceptos matemáticos utilizando tecnologías de la información, como calculadoras, elementos graficadores, hojas de cálculo, pero tampoco se enfocan en generar instrucciones.
- Se encuentran otros recursos de aprendizaje en los que enseña sobre las tecnologías de la información a los niños, pero se enfocan en el uso de internet o de otros programas ofimáticos, o en los conceptos de hardware y software.

Esto muestra claramente que se encuentra poco material educativo nacional relacionado con la enseñanza de este tema particular. Sin embargo, este tema no

debe ser novedoso. Se sabe que en muchos países a los estudiantes se les enseña desde temprana edad a generar instrucciones secuenciales que pronto se convierten en algoritmos más completos, y que utilizan para desarrollo de diversos software, especialmente para aplicaciones en robótica. En Japón, por ejemplo, se desarrolla el proyecto “NPO Hito Project’s Robot Summer School”, en el cual, estudiantes desde los 9 años aprenden conceptos básicos de robótica, ingeniería y una sesión de programación (Akihabara News, 2013). Según la página de Akihabara News:

“en realidad muchos de los chicos probablemente no terminarán estudiando carreras afines con la robótica, pero habrán ganado no solamente un conocimiento básico de sistemas robóticos a nivel físico y de software, sino también una invaluable capacidad lógica para resolución de problemas y una aptitud mecánica mejorada”.

En esta época de modernización y globalización, es necesario generar en nuestro país un espacio para que los estudiantes de educación básica primaria, especialmente los estudiantes de grado 5° mejoren sus competencias para resolver problemas lógico-matemáticos. En este punto surgió una gran incógnita para abordar este tema:

¿De qué manera es posible mejorar el desarrollo de las capacidades de razonamiento para resolución de problemas, en los niños de grado 5° de educación básica primaria, en algunas instituciones de educación de Bucaramanga?

De este gran interrogante, surgieron otras preguntas más específicas para lograr comprender y generar su respuesta: ¿De qué manera se puede presentar la temática de la lógica y la secuenciación para la resolución de problemas a partir de la generación de instrucciones? ¿Qué metodologías y técnicas de enseñanza serían las más adecuadas para orientar en esta temática en particular? ¿Qué herramientas adecuadas y de fácil acceso existen para la enseñanza de este tema? ¿Qué características deberían tener estas herramientas para el desarrollo de estas habilidades? ¿Qué metodologías de enseñanza brindan a los niños en algunos países considerados como tecnológicamente aventajados en el área de las tecnologías de la información? ¿Cómo se podría aplicar las metodologías y técnicas de enseñanza y combinarla con un conjunto de herramientas que ayuden

en el proceso de aprendizaje de la lógica de la programación? ¿Qué eficacia tendría este proceso de enseñanza en el mejoramiento de las competencias de resolución de problemas y cómo se mediría?

1.5 HIPÓTESIS DE LA INVESTIGACIÓN

De acuerdo con la teoría cognitiva de Piaget, citada en Kliegman y colaboradores (2011), los niños construyen el conocimiento por sí mismos a través de procesos vinculados de asimilación (tomando nuevas experiencias de acuerdo a esquemas existentes) y la acomodación (creando nuevos patrones de entendimiento para adaptarse a la nueva información), generando de esta forma una reorganización de sus procesos cognitivos en forma continua y activa. Esta es la base de las técnicas y estrategias cognoscitivas que se aplican en la enseñanza actualmente, incluso se ha usado como base para la construcción de otras teorías del aprendizaje, como el construccionismo, una teoría propuesta por Seymour Papert (1980), la cual ha sido un soporte fundamental de los trabajos que está desarrollando actualmente MIT Media Lab, quienes han mostrado cómo los niños toman un rol activo con el fin de construir programas de computador, pudiendo generar conocimientos y habilidades adicionales en el proceso, tales como la resolución de problemas (Kafai, 1996). Un aspecto interesante discutido aquí es que la técnica utilizada para la enseñanza de la computación para niños, puede ser soportada también en otros tipos de aprendizaje.

Por otro lado, de acuerdo a las teorías del comportamiento conocidas también como conductistas (Kliegman, Stanton, Schor, St. Geme III, & Behrman, 2011), la generación de aprendizaje radica en la determinación y modificación de factores medibles que aumentan o disminuyen la frecuencia de las ocurrencias de los comportamientos que se pretenden enseñar. La esencia de estos conjuntos de técnicas y estrategias radica en que los comportamientos que están positivamente reforzados se generarán más frecuencia mientras que los comportamientos que son negativamente reforzados o ignorados se generarán con menos frecuencia, haciendo de esta técnica sencilla y amplia de aplicar, y con una eficacia comprobada en muchas investigaciones y trabajos.

La hipótesis que se planteó en este trabajo es la siguiente:

Es posible mejorar las capacidades de razonamiento en cuanto a secuencialidad y desarrollo de procesos lógico-matemáticos para resolución de problemas, en los niños pertenecientes al grado 5° de educación básica primaria mediante el uso de las tecnologías de la información y de la comunicación con el fin de vincular la enseñanza de la lógica algorítmica desde temprana edad.

Para ello se puede utilizar una o varias herramientas software de apoyo determinadas a partir de un proceso de selección de las más adecuadas para este fin, aplicando una combinación de estrategias instruccionales cognoscitivas y/o conductistas tomadas a partir de referentes aplicados efectivamente en varios países, y por último, planteando un conjunto de actividades para ser aplicadas dentro de un plan curricular propuesto, y complementando el plan existente de la asignatura de tecnología e informática de educación básica primaria para el nivel académico propuesto.

1.6 OBJETIVOS DE LA INVESTIGACIÓN

1.6.1 Objetivo General

Establecer un marco de trabajo educativo para la enseñanza de macroinstrucciones a partir del entendimiento de procesos lógicos y matemáticos a niños pertenecientes al grado 5° de educación básica primaria en una institución educativa oficial en el Área Metropolitana de Bucaramanga.

1.6.2 Objetivos Específicos

- Desarrollar un estudio comparativo entre diferentes técnicas de enseñanza y herramientas educativas más adecuadas para orientar en la temática de elaboración de macroinstrucciones, obtenidas a partir de la revisión del estado del arte sobre experiencias similares.

- Estructurar una integración de actividades en el plan curricular de tecnología, aplicando las técnicas de enseñanza y recursos establecidos, en el proceso de aprendizaje de la lógica de macroinstrucciones para niños de grado 5° de primaria.
- Desarrollar un curso piloto con estudiantes de 5° de primaria pertenecientes a una institución oficial de educación básica primaria en el Área Metropolitana de Bucaramanga, implementando la serie de actividades planteada.
- Implementar una serie de pruebas de desempeño en competencias de resolución de problemas con indicadores de medición apropiados, con el fin de determinar el grado de mejoramiento de la habilidad de resolución de problemas como consecuencia del proceso de enseñanza de esta temática.

1.7 JUSTIFICACIÓN DE LA INVESTIGACIÓN

El proceso de aprendizaje en los niños es impresionante. Se sabe que incluso desde su nacimiento, el cerebro del niño crece y se consolida a una tasa asombrosa, mucho mayor que en cualquier otra etapa de la vida dado su alto grado de interconexiones sinápticas (Corporación para la Nutrición Infantil - CONIN, 1988), permitiendo en su primera década de vida desarrollar una amplia gama de habilidades. De acuerdo con la UNICEF, el niño requiere adquirir una amplia variedad de competencias cognitivas, sociales y prácticas para desenvolverse en este mundo cada vez más complejo. Es por ello que esta etapa es muy importante para la enseñanza y apropiación de habilidades que tengan que ver con el desarrollo de competencias de procesos lógicos ya que éstas influirán en ellos por el resto de su vida.

Por otra parte, la necesidad por la enseñanza de la lógica de la programación se ve en muchos otros países, incluso en Estados Unidos, donde la programación no ha sido enseñada como un tema primordial en las escuelas. De acuerdo con Doug Rushkoff, columnista de CNN, “las escuelas necesitan incorporar la programación de computadores dentro del currículo principal o se quedarían relegadas” (Great Schools, 2013). Esto denota la importancia del aprendizaje de esta temática a

nivel mundial, y la preocupación por suplir esta necesidad desde la escuela, ya que la utilización y generación de tecnología ha llegado a influir incluso en la economía de una nación. Tanto es así que, de acuerdo al informe de CEPAL (2012) sobre estudios estadísticos y prospectivos, hubo una considerable contribución al crecimiento económico de las tecnologías de la información y las comunicaciones y de la productividad en varios países de América Latina, especialmente en Brasil, Argentina, Chile y México.

Nuestro país no se quiere quedar atrás. El Ministerio de Tecnologías de la Información y la Comunicación MinTIC y el Ministerio de Educación Nacional han venido desarrollando proyectos con el fin de impulsar la formación en TIC, evidenciándose en planes que se han impulsado para mejorar el acceso y promover la apropiación y aplicación de las tecnologías, en particular en los niveles de educación básica primaria.

El momento de iniciar el acercamiento de los niños al área computacional es ahora, ya que se cuenta con información suficiente sobre experiencias, técnicas y herramientas aplicadas en otros entornos para establecer métodos de trabajo que se puedan aplicar al plan curricular de enseñanza básica primaria, y buscando no solo incursionar en estos nuevos conceptos en el área tecnológica, sino también contribuir al mejoramiento en la calidad educativa usando las tecnologías de la información.

1.8 LIMITACIONES

El presente estudio comprendió la revisión del estado del arte de sobre experiencias de enseñanza de programación para niños y jóvenes, con el fin de encontrar soportes adicionales a la hipótesis planteada, diferentes herramientas especializadas para la enseñanza de la programación para niños y jóvenes y diferentes técnicas de aplicación de la enseñanza de la programación en el aula. No se pretendió hacer un recorrido por todas las técnicas existentes ni hacer un debate sobre cuáles son las mejores sino que se seleccionaron las más apropiadas para estos tipos de temas, basados en investigaciones previas, delimitadas por características particulares y encontradas en revistas de

investigación o en proyectos conocidos que hicieron referencia a diversas técnicas y herramientas que apoyaron la enseñanza de la programación en niños y jóvenes.

Por otro lado, la aplicación de este estudio comprendió la generación de un plan curricular que abarcó solamente un bimestre académico para un grupo de estudiantes de nivel 5° de una institución de educación básica primaria del Área Metropolitana de Bucaramanga. Para ello se contó con la participación del Colegio San Juan de Girón Sebe Primaria “Eloy Valenzuela” del municipio de Girón, Santander. El colegio contó con condiciones óptimas de infraestructura tecnológica en el colegio, las herramientas implementadas se pudieron trabajar instaladas en los equipos y los recursos fueron suficientes para permitir el uso del computador por parte de los niños. No obstante se encontraron limitaciones como la restricción de tiempo para la aplicación del proyecto con los estudiantes, dado que sólo se contó con una hora semanal durante la clase de Tecnología para poder desarrollar la intervención; además, el personal docente no poseía una preparación teórica ideal para la enseñanza de los conceptos de programación propuestos. Por esta razón se diseñó el plan de área para personal docente que no contara con conocimientos avanzados de programación.

1.9 DEFINICIÓN DE TÉRMINOS

Algoritmo: Es un conjunto ordenado y finito de pasos, instrucciones u operaciones que permite hallar la solución de un problema. También se le conoce al método y notación en las distintas formas de cálculo para resolución de problemas matemáticos, en la cual dados un estado inicial y una entrada, siguiendo pasos secuenciales se llega a un estado final obteniendo una solución (Real Academia Española, 2001).

Constructivismo: Es una corriente pedagógica que postula la necesidad de entregar al alumno herramientas que le permitan crear sus propios procedimientos para resolver una situación problemática, lo cual implica que sus ideas se modifiquen y que pueda seguir aprendiendo (Ministerio de Educación Nacional, 2001).

Construccionismo: Es una teoría del aprendizaje que destaca la importancia de la acción, es decir del proceder activo en el proceso de aprendizaje, la cual parte del supuesto de que, para que se produzca aprendizaje, el conocimiento debe ser construido (o reconstruido) por el mismo sujeto que aprende a través de la acción, de modo que no es algo que simplemente se pueda transmitir, considerando además las actividades de confección o construcción de artefactos como facilitadoras del aprendizaje (Ackermannn, 2002).

Conductismo: Es una teoría del aprendizaje en la cual se aplica un condicionamiento en el sujeto que aprende, como un proceso por el cual las respuestas se unen a un estímulo particular que acarrea un significado positivo, partiendo de un estímulo neutro que no tenía ningún significado especial antes del condicionamiento, o también generando una acción en particular seguida por algo deseable o no deseable, modificando su comportamiento y generando así un proceso de aprendizaje (Skinner, 1977).

Macroinstrucciones: Son instrucciones equivalentes a otro grupo de instrucciones o comandos que, en conjunto, realizan una tarea más compleja. Esta equivalencia se define con el fin de ejecutar instrucciones con una sola orden, permitiendo simplificar la tarea de escribir programas complejos o extensos, ya sea creando instrucciones más fáciles de entender o reutilizando código. Además, permite escribir los programas en una sintaxis habitual de alto nivel y hace los textos de los programas más transparentes para la lectura y la percepción (Janalta Interactive Inc., 2010).

Competencias: En educación, se habla de competencias como el conjunto de conocimientos, procedimientos, valores y actitudes combinados, coordinados e integrados, que permite al estudiante resolver problemas específicos de forma autónoma, flexible y eficaz en una situación determinada (Ministerio de Educación Nacional, 2003).

2. MARCO TEÓRICO

2.1 ESTADO DEL ARTE

La revisión del estado del arte permitió generar un marco teórico que se abordó fundamentalmente en los siguientes conceptos:

- Experiencias de enseñanza de programación para niños y jóvenes
- Temáticas y contextos adecuado de enseñanza de la programación para niños
- Caracterización de herramientas adecuadas para la programación en niños
- Técnicas de aplicación de la enseñanza de la programación en el aula
- Modelos de evaluación de resultados en experiencias generadas de enseñanza.

Lameras y colaboradores (2010) presentaron un conjunto de propuestas pedagógicas y entornos de trabajo para mejorar el desempeño en las habilidades de programación en estudiantes de bachillerato de últimos niveles y universitarios, en el contexto de un proyecto europeo que involucró el estudio del currículo de ciencias de la computación en Austria, Estonia, Grecia e Italia. Otros 2 artículos (Alt, Astrachan, Forbes, Lucic, & Rodger, 2006; Lin, Zhang, Beck, & Olsen, 2009) presentaron diferentes alternativas para el uso y motivación hacia las ciencias de la computación incluso para la resolución de problemas reales sin necesidad de enseñar programación. Alt y colaboradores (2006) utilizaron para ello el análisis de datos estadísticos para conocer el comportamiento de las redes sociales, mientras que Lin y colaboradores (2009) por su parte utilizaron el contexto de la bioinformática y el análisis genético de cadenas de código ADN para la comprensión de relaciones evolutivas.

2.2 EXPERIENCIAS DE ENSEÑANZA DE PROGRAMACIÓN PARA NIÑOS Y JÓVENES

2.2.1 Experiencias internacionales

En los últimos 10 años se han generado varias experiencias de enseñanza de programación en niños y jóvenes en diferentes instituciones alrededor del mundo. En la tabla 1 se presenta un cuadro comparativo con un conjunto de 18 de estas experiencias de enseñanza, presentando sus diferentes características. En muchos países europeos (como por ejemplo Austria, Rep. Checa, Israel, Serbia, Dinamarca, Eslovaquia, Rusia), así como en Norteamérica (Estados Unidos, Canadá) y en Asia (Taiwan) se evidenció la preocupación por enseñar programación de computadores desde temprana edad, pero no se presentó un consenso en el que se establezca un currículo estándar establecido por sus gobiernos sobre las temáticas a tratar o los niveles académicos desde donde se debe comenzar dicha enseñanza.

Estas experiencias abarcaron todos los niveles educativos, desde preescolar hasta los primeros niveles universitarios, implicando su aplicación en contextos temáticos diferentes de acuerdo a su nivel. Ninguna temática fue aplicada de manera forzada sino que se adaptó al nivel tratado en el contexto de cada institución en cada país. Al comparar con el contexto curricular colombiano, se encontró que, por ejemplo, en educación primaria no se enseña de manera obligatoria el concepto del funcionamiento de circuitos eléctricos o de programación de computadores. La revisión de las herramientas y las metodologías adecuadas para la enseñanza de macroinstrucciones tuvo en cuenta que el contexto aplicado debe estar acorde con el contenido presentado en los lineamientos curriculares nacionales en el área de tecnología y de matemáticas.

Tabla 1. Cuadro comparativo de 18 experiencias de enseñanza de la programación a niños y jóvenes en el mundo

Autores y año	País o países de aplicación	Herramienta seleccionada	Temática presentada y contexto en el que aplica	Población objetivo (estudiantes) y presaberes	Técnica aplicada	Modelo pedagógico aplicado
(Abramovich, 2013)	Austria, Canadá, Rep. Checa, Israel, Serbia, USA	Utilitarios de matemáticas: GeoGebra, VisualMath, Maple	Matemáticas; contexto dependiente del tema	Diversos niveles educativos; con conocimientos previos de matemáticas	Resolución de problemas según el método Polya	Ninguno propuesto
(Liu, Cheng, & Huang, 2011)	Taiwan	Juego de simulación Train B&P	Programación; contexto sobre control de transporte de un sistema de ferrocarril	Primer año de universidad; sin experiencia en programación	Diseño experimental, simulación y generación de hipótesis, interpretación de datos, resolución de problemas según el método Polya	Prueba y error, Aprendizaje por ejemplo, Razonamiento, Aprendizaje por construcción.
(Valente, 2004)	Dinamarca	Cartas computacionales	Teoría de las ciencias de la computación y la información; contexto sobre circuitos básicos y dinámicos, problemas de probabilidad y transmisión de información	Primaria, de 8 a 10 años; sin experiencia en programación, incluso con limitación de formación matemática	Armado de circuitos de flujo (ubicación de cartas) y aplicación de reglas formales, armado de circuitos especiales para reutilización	Manipulación directa y aprendizaje por realización (construcción), desarrollo de proyectos top-down y bottom-up
(Felleisen, Findler, Flatt, & Krishnamurthi,	USA	Entorno de desarrollo DrScheme	Matemática plana, álgebra y programación funcional; contextos	Bachillerato, de 10 a 14 años; con presaberes de matemática	Programación funcional mediante modelos matemáticos	Modelamiento matemático

2009)			diferentes para crear simulaciones, animaciones y juegos	básica		
(Wolz, Leitner, Malan, & Maloney, 2009)	USA	Entorno de desarrollo Scratch	Programación básica, programación avanzada (eventos, concurrencia, hilos); contexto diferente de acuerdo a la temática	Primer semestre universidad, con conocimiento nulo o muy básico en programación	Enseñanza de programación mediante armado, transición a lenguaje de programación real	Aprendizaje por construcción
(Tomcsányiová & Tomcsányi, 2011)	Eslovaquia	Cada tarea bajo un applet	Uso de TIC para comunicación, Diagramas de flujo, Resolución de problemas, Pensamiento algorítmico, Principios de TIC, Sociedad de la Información; varios contextos con diferentes niveles de dificultad	Primaria, estudiantes de 2° a 4° (7-9 años), con habilidad mental	Desarrollo de torneo competitivo sobre resolución de tareas interesantes y evaluación de resultados	Competencia y evaluación. Trabajo con objetos concretos sin uso de abstracción. Clasificación, adecuación, Formulación de tareas.
(Doerschuk, Liu, & Mann, 2012)	USA	Plataforma GreenFoot	Conceptos básicos de programación, POO; contexto para programación de juegos	Jóvenes de grado 10° y 11°; ningún conocimiento sobre programación	Campamento de programación intensivo	Enseñanza de material instruccional preparado por estudiantes de niveles superiores

(Rogozhkina & Kushnirenko, 2011)	Rusia	Entorno PictoMir	Conceptos fundamentales de programación; contexto sobre juego de control de robot para llenado de casillas	Preescolar (5 años y medio a 7 años); manejo de mouse, ninguna o poca habilidad de lectura.	Juego para cumplimiento de objetivos, sin uso y con uso de computador	Instrucción básica, pruebas escritas gráficas, pruebas aplicadas en computador
(Burke & Kafai, 2010)	USA	Entorno de desarrollo Scratch	Programación básica, orientación a objetos. Contexto: creación de historias.	Estudiantes de educación media, entre 10 y 14 años	Programación básica para narración de historias y creación de juegos	Generación de sentido narrativo secuencial, capacidad de escritura.
(Javidi & Sheybani, 2009)	USA	Entornos de programación Kahootz y Squeak.	Programación para videojuegos. Contexto: temáticas sobre importancia de conservación del medio ambiente y salud.	Estudiantes de educación media de áreas urbanas y rurales	Diseño de videojuegos, creación de ideas con historias, preparación para Liga Lego.	Trabajo en equipo, exploración de software instruccional, autoreflexión.
(Kelleher, Pausch, & Kiesler, 2007)	USA	Entorno de desarrollo Storytelling Alice	Programación básica. Contexto: creación de historias.	Estudiantes de mujeres de educación básica y media (5° a 9°)	Enseñanza de herramienta y de conceptos de programación, quiz de conceptos y actitudes, construcción de historias.	Enseñanza a través de tutorial, uso de constructos de programación básica.
(Lee & Ko, 2011)	USA	Juego de programación Gidget	Programación básica, sintaxis para programación. Contexto: juego para ayudar a un robot a corregir sus	Personas de niveles académicos desde bachillerato hasta	Enseñanza de comandos básicos, composición de comandos más complejos, comprensión	Enseñanza a través de tutorial, presentación de variaciones sintácticas con errores. Aprendizaje por error.

			fallas de programación para cumplir misiones.	doctorados alrededor del mundo, sin ningún conocimiento previo de programación.	de errores.	
(Meyers, Cole, Korth, & Pluta, 2009)	USA	Entorno de programación Processing	Programación básica y avanzada (funciones, gráficos y fractales), procesamiento de sonidos. Contexto: análisis y generación de melodías e imágenes.	Estudiantes entre 12 y 17 años, con proficiencia en matemáticas y conocimiento en música.	Enseñanza de programación y música computacional. Representación de imágenes, manipulación y sintetización de sonidos en forma electrónica.	Construcción de imágenes y música.
(Rizvi, Humphries, Major, Jones, & Lauzun, 2011)	USA	Entorno de programación Scratch	Programación básica. Implementación de proyectos. Contexto: creación de proyecto multimedia con documentación.	Estudiantes recién ingresados a universidad	Curso de programación utilizando Scratch, explicación de constructos básicos y avanzados de programación.	Aprendizaje tradicional con uso de TIC
(Rodger, y otros, 2010)	USA	Entorno de programación Alice	Matemáticas, música, programación. Contexto: uso de videojuegos para diferentes contextos dependiendo de la	Todos los niveles educativos, desde primaria hasta universidad	Implementación de proyectos 3D de tipo videojuegos, para aplicar en matemáticas, programación y música.	Aprendizaje tradicional con uso de TIC

			temática.			
(Sipitakiat & Nusen, 2012)	Tailandia	Sistema de construcción por bloques físicos Robo-Blocks	Programación para robotica. Contexto: creación de minirobots	Niños entre 8 y 9 años.	Creación de robots mediante construcción de bloques. Enseñanza de modelo de depuración.	Aprendizaje por construcción
(Tarkan, y otros, 2010)	USA	Lenguaje de programación para niños Toque	Diseño de órdenes secuenciales. Contexto: Escenarios de cocina para preparación de recetas	Niños entre 7 y 11 años.	Proceso de diseño e implementación de recetas de cocina	Aprendizaje por construcción
(Zuckerman, Arida, & Resnick, 2005)	USA	Bloques de construcción electrónicos FlowBlocks y SystemBlocks	Conceptos matemáticos concretos y abstractos. Contexto: construcción de flujos para diversos escenarios	Niños entre 4 y 11 años.	Estructuras genéricas comparadas con objetos reales, nivel de abstracción, asociación semántica, analogías.	Manipulativos digitales inspirados en la metodología Montessori.

Fuente: Autor del proyecto

2.2.2 Experiencias locales

Con respecto a casos de estudio aplicados en Colombia que hayan utilizado tecnologías de la información en la escuela para la enseñanza de la programación a niños y jóvenes en Colombia, existe una iniciativa llamada Coderise (2012), que busca implementar un programa piloto con un grupo de estudiantes de bachillerato quienes toman un curso de nueve semanas en el que aprenden conceptos de desarrollo de software enfocado a aplicaciones web, sin embargo no se generó un estudio investigativo para determinar la eficacia en el desarrollo de habilidades de resolución de problemas en los estudiantes.

Una experiencia similar ha sido desarrollada por Motorola Foundation a través de la ONG Give to Colombia y la Fundación Gabriel Piedrahita Uribe (2013), en la cual se buscó capacitar a varios grupos de docentes de las áreas de Informática, Matemáticas y Ciencias Naturales de Educación Básica en diferentes regiones de Colombia, con el fin de enseñarles el uso educativo del entorno gráfico de programación de computadores Scratch, para que lo puedan implementar en sus aulas. Según los informes presentados por la Fundación, los niños y jóvenes que son beneficiados por la impartición de esta temática, no sólo quedan fascinados con el poder que les da Scratch, sino que les permite “mejorar sus aprendizajes y fortaleciendo competencias intelectuales como pensamiento computacional, pensamiento lógico, solución de problemas, creatividad, comunicación y trabajo colaborativo”. A pesar que la experiencia es muy interesante, en los informes no se presentó un soporte investigativo sobre la real eficacia del fortalecimiento de las competencias intelectuales al haber trabajado bajo esta metodología. Esto mostró una carencia de investigación sobre aplicaciones metodológicas para el mejoramiento de este tipo de competencias a nivel local, y es aquí donde se presenta una brecha de investigación, por la cual se enfocó el propósito de este proyecto.

2.2.3 Determinación y clasificación de técnicas

Se ha propuesto una clasificación de técnicas utilizadas para la enseñanza de la programación a partir de las experiencias encontradas, de la siguiente forma:

Tabla 2. Clasificación de técnicas utilizadas para la enseñanza de programación en niños y jóvenes

Técnica	Experiencias en que se aplicó
Uso de videojuegos para la enseñanza	Rogozhkina & Kushnirenko (2011); Lee & Ko (2011)
Creación de proyectos con temática fija	Liu, Cheng, & Huang (2011); Valente (2004); Sipitakiat & Nusen (2012); Zuckerman, Arida, & Resnick (2005); Tarkan, y otros (2010)
Creación de entornos para generación de situaciones	Felleisen, Findler, Flatt, & Krishnamurthi (2009); Wolz, Leitner, Malan, & Maloney (2009); Burke & Kafai (2010); Kelleher, Pausch, & Kiesler (2007); Meyers, Cole, Korth, & Pluta (2009); Rizvi, Humphries, Major, Jones, & Lauzun (2011)
Creación de videojuegos	Doerschuk, Liu, & Mann, (2012), Javidi & Sheybani (2009), Rodger, y otros (2010)
Herramientas computacionales como apoyo a la resolución de problemas	Abramovich (2013)
Generación de torneos o competencias	Tomcsányiová & Tomcsányi (2011)

Fuente: Autor del proyecto

Uso de videojuegos para la enseñanza: La primera técnica encontrada en estas 18 experiencias para la enseñanza de la programación se encontró en la generación de interés por este aprendizaje mediante el uso de videojuegos. Se clasificó en esta técnica el uso de elementos computacionales tipo juego que presentaran un objetivo particular, una historia de fondo, un conjunto de reglas a seguir y un conjunto de niveles a cumplir. Dentro de las experiencias revisadas en el estado del arte que utilizaron el uso de juegos para la enseñanza se encontraron la de Rogozhkina & Kushnirenko (2011) que usa un entorno de juego llamado PiktoMir, y la de Lee & Ko (2011) con un juego llamado Gidget.

El objetivo del experimento de Rogozhkina & Kushnirenko (2011) fue determinar la factibilidad de usar la herramienta en la enseñanza de los primeros elementos de programación a niños de nivel preescolar dado que los ambientes de enseñanza de la programación requieren tradicionalmente de ambientes de escritura, y por consiguiente los estudiantes deben tener una habilidad para leer y escribir apropiadamente, estableciendo un rango de edad limitado para que los niños pudieran tener la posibilidad de aprender. Se trabajó con 42 niños entre 5 y 7 años de una escuela preescolar en Moscú, con poca habilidad para la lectura y la escritura, utilizando el programa PiktoMir, un juego que presenta un ambiente sin texto que brindara soporte a la enseñanza de la programación a los niños, el cual trata sobre un pequeño robot llamado Fidget que debe arreglar recubrimientos dañados en el piso, y al cual se le deben asignar las órdenes dado que este robot no puede tomar decisiones por sí mismo. Se realizó el experimento en 8 semanas, en 2 jornadas semanales de 20 minutos cada una. En la primera jornada se enseñaba la teoría del juego sin el uso del computador, y la segunda jornada se realizaban ejercicios de programación. Durante la ejecución del experimento se tomaba un registro en el que se apuntaba el material de enseñanza, procedimientos de clase, retroalimentaciones y observaciones generales. Al completar el curso los niños fueron sometidos a un examen final, en el cual se presentaban 3 bloques de tareas, cada uno de 6 tareas, en los cuales el estudiante debía dibujar el desplazamiento del robot después de presentar un conjunto de instrucciones o encontrar el error de una serie de instrucciones mostrando el movimiento del robot.

En cuanto a la experiencia de Lee & Ko (2011), se tuvo como objetivo mostrar una mejora motivacional en la programación para un aprendizaje exitoso, por medio de una herramienta que brindara una retroalimentación más adecuada de los errores que pueda presentar, sin el uso de mensajes de fallo en un programa. Para el proyecto se reclutaron inicialmente 250 personas de todo el mundo a través de un sistema llamado Amazon Mechanical Turk. Se presentaron personas entre los 18 y los 59 años, para que jugaran una aplicación llamada Gidget, desarrollada en HTML5 y JavaScript mediante jQuery, la cual se guiaba a través de niveles que enseñaban el diseño y el análisis de los algoritmos usados por un robot, el cual debía ayudar a limpiar un derrame de químicos en un terreno con el fin de proteger la vida silvestre, encontrando elementos en el terreno, desplazándose, chequeando sus características y moviéndolos a otro lugar. Se requirió para el

experimento que los participantes completaran un conjunto de 15 niveles en el juego, comparando los resultados de personas que recibieron una retroalimentación más adecuada de los errores contra los resultados de las personas que no recibieron retroalimentación suficiente.

Creación de proyectos con temática fija: Dentro de las experiencias revisadas en el estado del arte, se encontró una técnica interesante que consiste en la creación de diversos tipos de proyectos usando para ello simuladores o aplicaciones que admitieran manipular el entorno de trabajo para generar diversas situaciones, permitiendo así plantear problemas particulares que pudieran ser resueltos a través del razonamiento lógico aplicando procedimientos algorítmicos. Estas técnicas pudieron clasificarse dependiendo del sistema utilizado (ya sea hardware especializado, software o una combinación de ambas). Además, se tuvo en cuenta que en los experimentos presentados se muestra una limitante de trabajo hacia una temática particular. Dentro de estas experiencias se encontraron el uso de cartas computacionales de Valente (2004), los sistemas de bloques de (Sipitakiat & Nusén, 2012) y de (Zuckerman, Arida, & Resnick, 2005); el trabajo de Liu, Cheng, & Huang (2011) usando el simulador Train B&P y la experiencia de creación de recetas de (Tarkan, y otros, 2010) usando el programa Toque.

El experimento de las cartas computacionales de Valente (2004) se basó en una propuesta de un curso exploratorio para niños con el fin de enseñar programación de una manera teórica, dado que los algoritmos y la programación han sido considerados a menudo como campos muy formales o complejos y se ha buscado desde hace tiempo representarlos de una forma más sencilla, a través de objetos físicos e interacciones con otros. El objetivo de este artículo fue presentar el curso propuesto para la enseñanza de la programación. Cada carta se comporta como un elemento sobre el cual se transporta un elemento computacional que es definido, construido y transportado dependiendo de cómo se coloquen las cartas y de un conjunto de reglas formales definidas. Cada carta se conformaba de una pequeña pieza cuadrada de papel con un dibujo en su parte superior y con indicadores de entrada y salida llamados puertos. Luego de armar un circuito con las piezas cuadradas, se procede a utilizar un indicador que se debía desplazar sobre estas cartas, respetando una serie de reglas.

Los trabajos de Sipitakiat & Nusén (2012) y de Zuckerman, Arida, & Resnick (2005) presentaron sistemas de programación tangibles, los cuales se han

convertido en una nueva tendencia en la enseñanza de la programación, permitiendo que los estudiantes puedan crear un programa conectando bloques físicos compuestos por componentes electrónicos que generaban un conjunto particular de instrucciones, los cuales se debían combinar para controlar el movimiento de un robot. En el experimento de Sipitakiat & Nusen (2012) se trabajó con el sistema RoboBlocks, con 52 niños entre 8 y 9 años, realizando actividades que implicaran mover el robot entre un laberinto o realizando un gráfico en el suelo. Se trabajó también un proceso de depuración con los estudiantes cuando no se obtenía el resultado esperado. Se trabajó con tres formas de depuración: la revisión paso a paso, la inserción de banderas en el lugar del problema y el transportador (un medidor de giro para el robot). Un aspecto particular encontrado en este experimento fue que aunque fue bastante motivador para los niños al principio, se perdía esta misma motivación al no presentar actividades que brindaran una solución a problemas diferentes a los movimientos restringidos de los robots. El experimento de Zuckerman, Arida, & Resnick (2005), se basó también en el ensamble de bloques electrónicos, pero para encender luces o emitir sonidos. En este caso particular, se les catalogó como manipulativos tipo Montessori porque estos bloques físicos permitían generar un mejor modelamiento de conceptos más abstractos. Se realizó el experimento con 25 niños entre 4 y 11 años, desarrollando un conjunto de proyectos que permitían establecer conceptos más complejos como tasas, acumulación, retroalimentación y probabilidad. Sin embargo se logró apreciar que aunque se intentó enseñar temas con un alto nivel de abstracción, solo estudiantes de 10 años lograban establecer analogías interesantes dado que a esta edad se encuentran hasta ahora desarrollando la capacidad cognitiva de abstracción.

La experiencia con el simulador de Liu, Cheng, & Huang (2011) se realizó para investigar el efecto del uso de simuladores en la resolución de problemas computacionales. Participaron en esta experiencia 117 estudiantes de bachillerato que ingresaron a una universidad al noreste de Taiwan, quienes se inscribieron a la asignatura "Introducción a las Ciencias de la Computación" y no tenían ninguna experiencia en la programación. La creación y el control de transporte de un sistema de ferrocarril fue el contexto de trabajo para el desarrollo de dicho experimento, dado que podía contar con un micromundo que le permitiera a los aprendices ser "arquitectos", generando ideas de construcción y probándolas, utilizando lo que se conoce como aprendizaje por construcción, el modelo

pedagógico presentado por Papert (1980). El programa Train B&P es una simulación de un sistema de construcción de ferrocarriles, en el cual los estudiantes pueden diseñar sistemas de carriles y establecer mediante programación los comportamientos del desplazamiento de los trenes en los rieles. Previo a la aplicación del experimento con la herramienta, se les presentó un contenido basado en lecciones teóricas sobre programación básica (algoritmos), y se les hizo una prueba antes de trabajar con la herramienta para medir su nivel de “flujo”. Luego, los estudiantes participaron durante dos semanas en el uso de esta herramienta, tiempo en el cual se tomaron indicadores adicionales de actividad. Durante este tiempo, a los estudiantes se les enseñaban los conceptos de orientación a objetos, condicionales, iteraciones y comunicaciones entre objetos. Al finalizar el experimento se realizó una prueba final de experiencia de aprendizaje en la que pudieron obtener impresiones sobre lo que aprendieron con el juego, esperando que se tuviera una visión de cómo el juego de simulación había generado habilidades de resolución de problemas computacionales obtenidos de los registros de actividad y de las pruebas.

El trabajo de Tarkan y colaboradores (2010) se basó fundamentalmente en mostrar la importancia del uso de la narración, y la generación de argumentos para la enseñanza de la programación con el fin de superar limitaciones impuestas por la aplicación. Para ello los investigadores, en conjunto con un grupo de niños, desarrollaron un lenguaje de programación llamado Toque implementándolo en un escenario de cocina, para crear programas tipo “recetas” con animaciones controladas con un dispositivo de detección de movimiento. Los niños que participaron en el proceso de creación del programa fueron 9 niños de 7 a 11 años y el programa se desarrolló durante dos años, quienes no solo desarrollaban el programa sino que también jugaban con su propia creación para determinar aspectos por mejorar y trabajar proyectos al respecto.

Creación de entornos para generación de situaciones: Se encontró una técnica en la cual se utiliza una herramienta de desarrollo para crear un entorno particular desde cero, y por medio de procesamientos secuenciales lógicos y/o matemáticos se establecen situaciones interesantes en los cuales los niños pueden explorar y manipular mediante la programación. Entre las experiencias que utilizaron esta técnica se encontraron las de Felleisen, Findler, Flatt, &

Krishnamurthi (2009) mediante el uso del entorno de programación Dr. Scheme, los trabajos de Wolz, Leitner, Malan, & Maloney (2009); Burke & Kafai (2010) y Rizvi, Humphries, Major, Jones, & Lauzun (2011) quienes usaron el entorno de programación Scratch, la experiencia de Kelleher, Pausch, & Kiesler (2007) con el entorno Alice, y el trabajo de Meyers, Cole, Korth, & Pluta (2009) con el entorno de programación Processing.

Felleisen, Findler, Flatt, & Krishnamurthi (2009) crearon en su propuesta entornos que requirieron funciones matemáticas para determinar y controlar el comportamiento de cohetes usando Dr. Scheme, un entorno que utiliza el paradigma de programación funcional. En dicho experimento se buscó proveer un currículo que presentara una forma de enseñar conceptos pre-algebraicos en estudiantes de bachillerato, trabajando en pruebas de teorías matemáticas y diseño orientado a objetos. En este experimento se trabajó con estudiantes de educación media (de 10 a 14 años), y han trabajado con estos estudiantes desde el año 2006 creando un conjunto de proyectos tipo juego e implementando un curso especial de 9 semanas de duración con sesiones de dos horas semanales. Los autores de dicha experiencia ya habían trabajado con este entorno de programación desde el año 2003 con estudiantes universitarios, realizando otro tipo de proyectos.

Wolz, Leitner, Malan, & Maloney (2009) utilizaron el entorno Scratch para crear todo tipo de entornos 2D que involucraran procesos asincrónicos y concurrentes, tales como videojuegos, historietas, tarjetas animadas, parodias de televisión, tutoriales educativos, simulaciones científicas, entre otros, siendo este experimento aplicado en estudiantes que iniciaban el curso de ciencias de la computación en la universidad. En su artículo, los autores presentaron una justificación de integración del trabajo con Scratch en el currículo de ciencias de la computación para los primeros grados de universidad, basados en la capacidad de generar una transición más amigable en la enseñanza de la programación para lenguajes más robustos como Java o C. Rizvi, Humphries, Major, Jones, & Lauzun (2011) utilizaron Scratch para la creación de animaciones, proyectos multimedia y juegos con el fin de aplicarlo en el curso inicial de ciencias de la computación y determinar la eficacia en este curso de la mejora de la retención, el rendimiento y la actitud en cuanto a sus habilidades de programación. En esta experiencia se trabajó con estudiantes que recién se matricularon a la universidad y tomaron el

curso de ciencias de la computación. En particular, centraron su atención en estudiantes “en riesgo”, es decir, estudiantes con una preparación muy débil en matemáticas. Con dichos estudiantes se buscó trabajar en la elaboración de varios proyectos multimedia, los cuales involucraron el uso de una cantidad significativamente alta de gráficos, elementos animados y sonidos. Scratch fue usado también por Burke & Kafai (2010) para trabajar con jóvenes de último grado de bachillerato (K-12), con el fin de crear historias y determinando si la creación de programas de computador por parte de los niños podría ayudarles a desarrollar su capacidad de narrar cuentos y mejorar sus habilidades de escritura. Aunque se trabajó en la enseñanza de los conceptos básicos de la programación en dicho experimento, se observó que durante un tiempo de 6 semanas se enfocó más en los conceptos usados en el desarrollo de las historias que en los componentes técnicos del desarrollo de los programas.

Kelleher, Pausch, & Kiesler (2007) trabajaron también en la creación de historias usando la herramienta Alice. En este experimento se trabajó con 88 niñas de las tropas Scout de un promedio de 12.5 años, con el fin de demostrar que se puede desarrollar un sistema de enseñanza de la programación que pueda presentar a los niños y muy particularmente a las niñas una experiencia positiva con la programación de computadores, inspirando a más estudiantes a continuar con estos tipos de asignaturas de ciencias de la computación e incluso mejorar la representación de las mujeres en esta área. Los autores de esta experiencia habían trabajado previamente con la herramienta Alice desde el año 2005.

Meyers, Cole, Korth, & Pluta (2009) usaron la herramienta Processing para enseñar ciencias de la computación mediante la creación de música computacional. En esta experiencia se trabajó con niños de los grados 6° a 12° (entre 12 y 17 años) con cierta experiencia musical y habilidad matemática. Se realizó un curso intensivo de 16 días, los cuales se trabajaban entre semana 2 horas y media y los sábados durante una jornada de 7 horas.

Creación de videojuegos: Otra técnica muy interesante es el uso de herramientas de desarrollo para creación de videojuegos. Se asemeja mucho a la técnica anterior en cuanto que se debe crear un entorno desde cero y se debe también generar procesamientos secuenciales lógicos y/o matemáticos para

establecer situaciones de juego. Estas situaciones de juego se logran obtener mediante una explicación adicional sobre las bases para la creación de juegos. Entre las experiencias que utilizaron esta técnica se encontraron las presentadas por Doerschuk, Liu, & Mann (2012) mediante el entorno de programación Greenfoot, el experimento de Javidi & Sheybani (2009) quienes trabajaron con dos entornos llamados Kahootz y Squeak, y la experiencia de Rodger, y otros (2010) quienes enseñaron a desarrollar videojuegos utilizando Alice.

Doerschuk, Liu, & Mann (2012) presentaron en su experiencia una forma para poder incrementar la participación estudiantil en los programas de Ciencias de la Computación en la Universidad Lamar en Texas (EE.UU.). Para ello desarrollaron un plan estratégico llamado “INSPIRED”, que abordó la enseñanza de la creación de videojuegos a estudiantes de primer semestre de universidad. En conjunto con un grupo de estudiantes de último semestre crearon los materiales instruccionales, enseñaron y ayudaron a aplicar este plan. Para ello utilizaron la herramienta Greenfoot, un entorno gratuito para el desarrollo de videojuegos basado en Java, y prepararon una serie de recursos y ambientes para preestablecer los escenarios, los objetivos y las reglas de los juegos. A su vez, prepararon el material educativo y los códigos fuente de varios de los juegos que los estudiantes debían modificar, significando que en varios proyectos los estudiantes no partían desde cero sino que aprendían conceptos a partir de recursos y proyectos ya existentes.

Javidi & Sheybani (2009) desarrollaron un plan similar para estudiantes de bachillerato en Virginia (EE.UU.), llamado “DIGINSPIRED” que tuvo como objetivo no solo enseñar sobre programación de computadores sino también sobre la creación de gráficos computacionales y animación, y enseñar sobre tecnologías requeridas para la creación de este tipo de videojuegos. El trabajo consistió en la creación de juegos cuya temática abordara una de 4 áreas de ciencias: reciclaje, nutrición, ejercicio físico o consumo de drogas. Se elaboró el proyecto con 89 estudiantes de 3 diversos colegios de bachillerato durante el año, asistiendo a 15 sesiones los días sábados y un curso adicional de verano. Se observó que en el desarrollo de videojuego, varios estudiantes demoraron dos semanas en su implementación.

Rodger y colaboradores (2010) integraron el desarrollo de videojuegos en varias áreas en los niveles de educación básico, medio y universitario. En esta experiencia se identificó que, utilizando esta técnica, los estudiantes de 3° a 5°

grado deberían estar en la capacidad de crear y seguir un algoritmo para poder codificar y probar un sencillo programa secuencial mientras que un estudiante de 6° a 8° debería poder ser capaz de codificar y probar programas para resolver un problema usando variables, decisiones y ciclos. También se observó que han trabajado en este proyecto durante 8 años a nivel de bachillerato y 4 años a nivel universitario, habiendo generado una serie de recursos entre tutoriales y archivos para usar en el desarrollo de los juegos.

Uso de herramientas computacionales como apoyo a la resolución de problemas: Se encontró esta técnica revisando la experiencia de investigación de Abramovich (2013). El autor presentó una compilación de 10 reportes de experiencias en varios países que trabajaron con esta técnica. Abramovich argumentó que la efectividad en la enseñanza de las teorías matemáticas depende del uso apropiado de tecnologías para soportar su enseñanza, demostrando la dualidad de la enseñanza y el uso de los computadores. En las experiencias presentadas por el autor, se mencionaron algunas herramientas software para el apoyo en la enseñanza de las matemáticas, como GeoGebra, Rhinoceros, VisualMath, Calculadora Gráfica, Mathematica, Maple, MiniTab, SimCalc, MathWorlds, SketchPad, Pathom, NonEuclid, entre otras. Estas herramientas permiten generar modelos matemáticos a partir del ingreso de ecuaciones, variables y restricciones con el fin de analizar datos y presentar soluciones basadas en números. Presentó Abramovich que usando GeoGebra en particular se buscó reformular problemas tradicionales algebraicos en entornos más significativos; con Rhinoceros se mostró el valor del modelado tridimensional mediante el uso de la geometría descriptiva y el uso de ecuaciones matemáticas para crear sólidos de revolución; con VisualMath se mostró también la posibilidad de crear libros electrónicos, presentando diversos gráficos interactivos que soportaron exploraciones de conceptos matemáticos particulares. Particularmente se mostró cómo usando estas herramientas, manipulando datos de variables y realizando ciertas codificaciones se podrían presentar casos de estudio de problemas reales y comprender determinadas situaciones o encontrar resultados adecuados.

Generación de torneos o competencias: Se encontró una técnica que utiliza el desarrollo de torneos competitivos como un método para la enseñanza y además para probar el conocimiento adquirido por los niños en el área de informática elemental, una asignatura que se brinda en el currículo de educación primaria en Eslovaquia. Dentro de esta asignatura se enseñan conceptos de información, comunicación, TIC, resolución de problemas, pensamiento algorítmico, flujos, entre otros, dando a los estudiantes las bases de la programación y la habilidad para resolver problemas mediante el uso de las tecnologías digitales. Tomcsányiová & Tomcsányi (2011) prepararon y probaron un torneo especial para niños entre 8 y 9 años, con el fin de generar actividades intraclase y extraclase en los colegios para mejorarles sus habilidades para su participación. Se basaron en un torneo existente llamado Torneo Bebras que se realiza en Lituania con estudiantes de educación secundaria para desarrollar su propio torneo llamado Pequeño Bebras. Dentro de las actividades desarrolladas para su adaptación se encontraron la traducción de tareas, la adaptación de los datos, la recategorización de problemas, la simplificación de tareas, el rediseño de imágenes y la implementación de software para generar tareas interactivas. Se encontraron otros aspectos más pedagógicos para bajar el complejo nivel de las preguntas, reduciendo los textos, mostrando elementos concretos más que intentar presentar ideas abstractas, adaptando las imágenes a colores y tamaños apropiados, y delimitando la cantidad de tareas para no cansar a los niños.

2.3 TEMÁTICAS Y CONTEXTOS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS

Se ha utilizado la enseñanza de la programación en muchos contextos, entre cuales se incluyen creación de historias (Burke & Kafai, 2010; Kelleher, Pausch, & Kiesler, 2007), conservación del medio ambiente y salud (Javidi & Sheybani, 2009), procesamiento de sonidos musicales (Meyers, Cole, Korth, & Pluta, 2009), escenarios de cocina para preparación de recetas (Tarkan, y otros, 2010), control de transporte (Liu, Cheng, & Huang, 2011), así como también los clásicos contextos matemáticos como modelamiento (Abramovich, 2013; Felleisen, Findler, Flatt, & Krishnamurthi, 2009) y tecnológicos, como programación de robots reales

(Sipitakiat & Nusen, 2012), programación de computadores (Wolz, Leitner, Malan, & Maloney, 2009), uso de videojuegos para cumplimiento de objetivos (Lee & Ko, 2011; Rodger, y otros, 2010; Rogozhkina & Kushnirenko, 2011; Tomcsányiová & Tomcsányi, 2011), creación de videojuegos (Doerschuk, Liu, & Mann, 2012), flujos (Zuckerman, Arida, & Resnick, 2005; Valente, 2004) y multimedia (Rizvi, Humphries, Major, Jones, & Lauzun, 2011). A pesar de todos estos diversos contextos y temáticas centrales, siempre se enseña dentro de éstos la organización de un conjunto de órdenes secuenciales para la ejecución de tareas particulares. Este tópico particular llamado algoritmia es enseñado a través de un conjunto de temas más específicos. De acuerdo a estas experiencias, estos conjuntos de temas son lo que denominaron “constructos de programación” los cuales hacen referencia a los siguientes temas:

- Tipos de variables (numéricas, textuales, booleanas)
- Sintaxis (lenguaje formal) de la herramienta
- Reusabilidad de ejemplos y experimentación
- Secuencialidad, diseño de órdenes (ideación y formulación de modelos computacionales), analogías con escenarios reales.
- Entradas y salidas de información
- Entendimiento de respuestas del sistema
- Depuración (corrección de errores) pre-ejecución, en ejecución, post-ejecución
- Estructuras de control: condicionales, iteraciones, anidaciones
- Arreglos y matrices
- Funciones y expresiones matemáticas
- Subrutinas
- Gráficas y coordenadas cartesianas
- Imágenes, animaciones y sonido (multimedia)
- Interactividad con teclado y mouse, programación de eventos
- Orientación a objetos, clases, herencia
- Transición a un lenguaje de programación (Java, C++)
- Simulación de comunicación (sincronización)
- Actividades de no programación (diseño de escenarios e interfaz)

Determinar la edad adecuada en los niños para explicar estos conceptos fue fundamentalmente importante. Según Michael Kölling - citado en Utting y

colaboradores (2010) – no se presenta un rango especial de edades para la enseñanza de la programación en general, incluso asegura que entre más temprano se enseñe, mucho mejor. Sin embargo afirmó también que se generan “fronteras” con respecto a sus desarrollos cognitivos para poder comprender conceptos y tecnologías específicas. Se reflejaron problemas en la comprensión de sintaxis complejas por parte de niños pequeños (10 años), y una vez que desorganizaron el código, no fueron capaces de repararlo fácilmente. A pesar que los niños de estas edades pueden entender los conceptos de programación, no logran lidiar con sintaxis complejas. En el experimento realizado por Sipitakiat y Nusen (2012), se intentó explicar las operaciones básicas de un nuevo entorno llamado Robo-Blocks a niños de 5 a 12 años, pero se dieron cuenta que los niños menores de 8 años a menudo tenían dificultad entendiendo estas operaciones, en particular por los comandos de girar, desplazarse, pues eran confusas las órdenes que se usaron. Estas razones permitieron formalizar el rango ideal de edades en los estudiantes para poder iniciar esta investigación, el cual se estableció en 10 años, siendo esta la edad promedio de los estudiantes que asisten a grado 5° de educación básica primaria.

En la experiencia desarrollada por la Fundación Gabriel Piedrahita Uribe, se presentó el contenido temático que se aplicó para el grado quinto de primaria en el Instituto de Nuestra Señora de la Asunción – INSA (2012). Dicha experiencia se fundamentó particularmente en el uso del entorno Scratch, en la cual se presentaron los siguientes “constructos”:

- Utilización de operaciones matemáticas y booleanas
- Creación y utilización de variables
- Concepto de algoritmo
- Determinación de pasos para resolver problemas
- Uso de identificadores
- Asignación
- Pseudocódigo
- Diagramas de flujo y reglas de construcción
- Operadores y expresiones
- Contadores y acumuladores
- Estructuras condicionales
- Ciclos

Por otra parte, se planteó un contexto apropiado para la aplicación de estos nuevos conceptos, enfocado principalmente a la resolución de problemas. En un artículo (Grover, 2009) se afirmó que la resolución de problemas, el pensamiento crítico y la gestión de la información son reforzados a través de la profundización de estos temas que integran las ciencias de la computación, y que lo interesante de estos temas es que vienen soportados sobre bases matemáticas, particularmente de niveles que se ven en los colegios. Justamente uno de los contextos encontrados con más frecuencia fue en la aplicación de la programación en el área de matemáticas. Grover aseguró que el pensamiento algorítmico ayuda a los estudiantes a establecer una serie de pasos de un problema y plasmarlos en un programa, permitiendo que el estudiante pueda adquirir habilidades adicionales para definir y establecer claramente el problema, romper el problema en subproblemas más pequeños y manejables, y describir la solución en un conjunto bien definido de pasos.

De acuerdo con los estándares de competencias para el área de matemáticas establecidos por el Ministerio de Educación Nacional (2003) para el grado 5°, los estudiantes de este nivel pueden construir diferentes tipos de modelos matemáticos y generar diversos métodos para la resolución de problemas referentes a los siguientes temas:

- Propiedades de objetos (atributos)
- Estimaciones de medidas
- Operaciones con magnitudes
- Datos para estadística
- Probabilidades de ocurrencia
- Patrones matemáticos y secuencias
- Expresiones numéricas
- Propiedades de figuras geométricas
- Fracciones y notación decimal
- Aplicación de propiedades de números naturales
- Proporcionalidades directas e inversas
- Estrategias de cálculo para resolución de problemas
- Potenciación y radicación
- Objetos bidimensionales y tridimensionales
- Ángulos en diversos contextos

- Sistemas de coordenadas y localización espacial
- Congruencia y semejanza
- Descomposición de sólidos

2.4 HERRAMIENTAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN EN NIÑOS

Existe un sinnúmero de herramientas para la enseñanza de la programación para niños, cada uno con características muy diversas. Tarkan y colaboradores (2010), hablaron sobre una clasificación inicial de herramientas de programación para la enseñanza, dividiendo estas en dos grandes categorías. En una de ellas se habló sobre lenguajes de programación visuales, es decir, ambientes gráficos que se pueden usar para crear historias interactivas y juegos entre muchas otras. Dentro de estas se encuentran los lenguajes escritos mediante bloques de construcción (como por ejemplo Alice o Scratch), o mediante una sintaxis sencilla basada en reglas (como ToonTalk, Hands o Kodu). En la otra se encontraron los sistemas de programación tangibles, sistemas físicos para aprender a programar, tales como bloques físicos (Zuckerman, Arida, & Resnick, 2005; Sipitakiat & Nusén, 2012), o incluso cartas físicas (Valente, 2004).

Kelleher y Pausch (2005) propusieron una especie de clasificación taxonómica de estas herramientas. Lo interesante de este estudio fue la clasificación en cuanto a diversos atributos con los que cuenta cada herramienta de acuerdo a los criterios de enseñanza que se deben manejar. Estos fueron los siguientes:

- Estilo de programación (procedimental –estructural-, funcional, orientado a objetos, basado en objetos, orientado a eventos, basado en eventos y basado en estado de máquina).
- Constructos de programación disponibles (dependiendo de la herramienta se brinda la posibilidad de trabajar con diferentes comandos particulares de programación, como creación de variables, generación de ciclos, control de flujo mediante condicionales, entre otros)

- Forma de presentación del código en el entorno de programación (mediante texto, imágenes, diagramas de flujo, animaciones, formularios, máquinas de estado u objetos físicos)
- Forma en la que el usuario debe construir el código (tecleando órdenes, ensamblando gráficos, demostrando acciones mediante una interfaz, seleccionando opciones válidas, llenando datos en formularios, ensamblando objetos físicos)
- Soporte adicional para comprender el comportamiento de sus códigos (mediante una historia de fondo para explicar el funcionamiento, mediante soporte de depuración, escogiendo comandos físicos para actuar en el programa, permitiendo hacer cambios en tiempo de ejecución, generando programas de ejemplo)
- Prevención de errores en la interfaz de programación (mediante la forma de los objetos para conectarlos correctamente, seleccionando un conjunto de opciones válidas basado en su posición actual, usando edición dirigida de sintaxis, o suministrando mensajes de error para que el usuario los identifique más rápidamente)
- Intento explícito de hacer el lenguaje más fácil de entender (limitación de comandos, selección por palabras clave, remover puntuaciones innecesarias, usar frases para hacerlo lo más natural posible, removiendo redundancias en el lenguaje)
- Soporte colaborativo (soporte en parejas, soporte multiusuario, compartir resultados)
- Consideración de uso adicional (por diversión y motivación, utilidad del sistema para resolver un problema particular, como sistema educativo para ayudar a la enseñanza)

Lameras y colaboradores (2010) generaron otra propuesta de caracterización de herramientas software para la enseñanza de la programación mediante el uso de lenguajes de programación educativos. Estas características fueron las siguientes:

- Soporte de orientación a objetos (facilidad de transición a lenguajes actuales y procesos adicionales de abstracción (características y funciones)
- Disponibilidad en forma de un IDE (presentar un entorno de programación similar a los entornos de programas de alto nivel)

- Soporte de programación visual (contribución de trabajar con imágenes para disminuir o eliminar errores de sintaxis)
- Gratuito y de código abierto (con el fin de disminuir costos educativos y para adaptar las herramientas a diferentes contextos o idiomas)
- Desarrollos en comunidades activas
- Existencia de versiones estables
- Rangos de edades indicados
- Idiomas que soporta
- Soporte de localización
- Sistema operativo
- Documentación disponible

Varias investigaciones hablan sobre la necesidad de trabajar con lenguajes de programación especialmente diseñados para la enseñanza de la programación. Como un ejemplo, Schwartz, Stagner y Morrison (2006) propusieron un lenguaje de programación especializado para niños llamado KPL, y propusieron criterios especiales llamados puntos de diseño que tuvieron en cuenta para la creación de KPL. Entre ellos se encuentran la diversión que generan, la accesibilidad y usabilidad, la atracción, la simplicidad, la recompensa al presentar resultados inmediatos, su funcionalidad máxima usando mínima codificación, su capacidad de progresividad para la enseñanza, la adecuada preparación que genera para permitirle conocer otro lenguaje de programación, la consistencia con diseños modernos de software, la capacidad de ser publicable, la soportabilidad y la posibilidad de internacionalización.

Un aspecto muy importante a tener en cuenta para la selección de las herramientas adecuadas para la enseñanza de la programación es la valoración de la sencillez de la sintaxis, incluso por encima de la importancia de un ambiente gráfico. Muchos críticos aseguran que no se presenta un desarrollo claro de las habilidades de los niños para resolver problemas luego de ser sometidos a un curso de programación, puesto que no entendían cómo resolver problemas usando estructuras de programas puesto que se estancaban en errores sintácticos, y no se veían claramente los beneficios educativos (Sipitakiat & Nusen, 2012). Según afirmaron Fidge y Teague (2009), las personas que incursionan en la programación tienen dificultades en desarrollar un problema algorítmico tratando de dar soluciones a errores y restricciones sintácticas usando

los lenguajes formales de programación, pudiendo llegar a causar, además de una pérdida de su objetivo inicial, una sensación de frustración que lo que hace es perjudicar más que ayudar. En esto concuerda Stephen Cooper – citado en Utting y colaboradores (2010) – quien afirmó que gran parte de las primeras frustraciones asociadas con la depuración (solución de errores) se minimizan utilizando un entorno fácil de trabajar, pues así que los estudiantes son capaces de avanzar por su cuenta, en su intento de resolver problemas específicos. Fidge y Teague (2009) hablaron sobre la programación "libre de sintaxis" con el fin de separar la noción de codificación de la habilidad para resolver problemas mediante programación.

La exploración a través de estas investigaciones permitió generar como resultado una compilación de varias características importantes que deben tener la herramienta o herramientas que fueron analizadas para su selección:

- Debe soportar los primeros temas de programación para su enseñanza (creación de variables, generación de ciclos, control de flujo mediante condicionales, entre otros)
- Debe presentar el código en una sintaxis muy fácil de aprender, y de ser posible que presente un método gráfico para armar o representar el comportamiento del programa por medio de imágenes.
- No necesariamente debe prevenir errores durante su programación, sino más bien debe poder permitir depurar código de una forma sencilla –ya sea mostrando mensajes de error explicando sobre el error y la posible forma de solucionarlo.
- Debe presentar un buen soporte a usuario, con el fin que docentes y estudiantes puedan obtener documentación adicional que le sirva de aporte para la creación de proyectos más complejos.
- Dada la restricción de edades y la competencia actual de los niños en cuanto a su experiencia con otros idiomas, se propuso que la interfaz (de ser posible, así como la sintaxis de programación) sean en español.

2.5 TÉCNICAS DE APLICACIÓN DE LA ENSEÑANZA DE LA PROGRAMACIÓN EN EL AULA

El desarrollo del aprendizaje se entiende mejor a través de la teoría cognitiva planteada por Jean Piaget (Kliegman, Stanton, Schor, St. Geme III, & Behrman, 2011). De acuerdo a esta teoría, Piaget describe cómo los niños construyen activamente el conocimiento por sí mismos a través de procesos vinculados de asimilación (tomando nuevas experiencias de acuerdo a esquemas existentes) y la acomodación (creando nuevos patrones de entendimiento para adaptarse a la nueva información), y de esta forma, los niños están reorganizando los procesos cognitivos en forma continua y activa. Los conceptos básicos de la teoría cognitiva de Piaget se han mantenido y se han utilizado como base para la creación de otras teorías del aprendizaje, como el construccionismo, propuesto por Seymour Papert. Según Papert (1980), el aprendizaje de un nuevo concepto se vuelve fácil si se logra asimilar la nueva idea aprendida comparándola y acoplándola a la colección de los modelos existentes en su conocimiento previo. Si no se logra el aprendizaje se torna difícil. En otras palabras, lo que un individuo puede aprender y cómo lo aprende, depende principalmente en los modelos que tiene disponibles en su mente.

La edad apropiada para poder enseñar estos nuevos conceptos y que puedan no solo generar sino también mejorar sus habilidades se encuentra dentro de los 8 a los 11 años de edad. Según Warren Buckleitner – citado en Walton-Hadlock (2008) – las tecnologías apropiadas que sean utilizadas en los estudiantes de educación básica deberían estimularlos a generar socialización, a expandir sus habilidades y conocimientos y suministrándoles múltiples niveles de reto, que sean fáciles de utilizar y que puedan fácilmente ser enlazados con otros tipos de medios, como libros o tutoriales. El nuevo modelo que se enseñó en este proyecto está basado en el creación de un pensamiento algorítmico, con el fin que los niños pudieran generar nuevas estructuras de aprendizaje aplicables en el área de matemáticas para la resolución de problemas. Mientras que Grover (2009) lo llama pensamiento algorítmico, en National Academy of Sciences (2010), se le denominó pensamiento computacional, el cual “debería ser concebido incluso como una habilidad intelectual fundamental comparable a la lectura, la escritura, la

narración y la aritmética”, es decir, debe ser una habilidad cognitiva que una persona debiera tenerla en esta época actual.

Para lograr transmitir esta habilidad en forma eficaz a los estudiantes, es necesario generar un curso estructurado y diseñado adecuadamente. Con el fin de poder generar un buen diseño de un curso que se fundamente en la enseñanza de la programación, se enfocó particularmente a desarrollar las habilidades de los estudiantes por aplicar su conocimiento a la solución de problemas particulares mientras “se logre al mismo tiempo generar programas coherentemente desarrollados” (Lameras, y otros, 2010). Resnick y colaboradores (2009) hablaron particularmente sobre 3 habilidades que se adquieren en la enseñanza de la programación: habilidad para pensar creativamente, habilidad para razonar sistemáticamente y habilidad para trabajar colaborativamente. Según Resnick, no se busca preparar a las personas para que se conviertan en profesionales programadores de software sino para nutrir a una nueva generación de pensadores sistemáticos y creativos mediante el uso de la programación para expresar sus ideas.

Uno de los retos que se presentan en la enseñanza de la programación es justamente generar el interés por este aprendizaje, balanceando los aspectos educativos y motivacionales mediante la alineación de materiales interesantes de trabajo con marcos de trabajo educativos bien fundamentados (Repenning & Ioannidou, 2008). Una de las formas más usadas para generar este interés por el aprendizaje se encontró en el uso de juegos para la enseñanza. De acuerdo con Kirriemuir y McFarlane (2004), se puede generar un gran interés en el que incluso los estudiantes se vuelvan ajenos a las distracciones, y esto se logra durante actividades que sean muy divertidas, como en el uso de los videojuegos ya que “al combinarlo además con la curiosidad y la fantasía, adquieren un nivel de compromiso tal que desaparecen las distracciones”.

Existe una amplia documentación al respecto del beneficio del uso de los videojuegos en la educación, pero también sobre la no conveniencia de estos tipos de elementos en el ámbito educativo. Mitchell y Savill-Smith en su estado del arte (2004), encontraron que el uso de los videojuegos puede estimular la motivación y el compromiso de los usuarios por la consecución de objetivos, mediante el desarrollo y el aprendizaje de ciertas habilidades sociales y cognitivas. Sin embargo, encontraron también que el uso frecuente de juegos de computador,

especialmente los que contengan algún contenido violento, que pueden generar sentimientos de ansiedad y tendencias psicosociales negativas como violencia o aislamiento, así como problemas de salud si su uso llegase a ser catalogado como adictivo. Una forma para poder utilizar estos tipos de herramientas fue la adecuación de éstos al contexto actual de trabajo en el grupo. Malone – citado en Kirriemuir y McFarlane (2004, pág. 4) – indicó que para que se logre el uso efectivo de un juego contextualizado en una experiencia de aprendizaje, las actividades que se les propongan a los estudiantes deberían ser estructuradas de tal forma que puedan incrementar o disminuir el nivel de dificultad del videojuego utilizado, con el fin de trabajar específicamente en el nivel de habilidad que posee, y de esta manera no caen en sentimientos de ansiedad, frustración o aburrimiento.

Otra forma utilizada para generar interés en el aprendizaje fue el presentado por Hug, Guenther y Wenk (2013) en el que se logró el interés en los estudiantes presentándoles una oportunidad de construir proyectos de la vida real para solucionar problemas específicos, pero sin salirse del diseño curricular que se les ofrece. Dichos proyectos fueron diseñados con sumo cuidado y sin improvisaciones, con el fin de poder orientar de una manera adecuada al estudiante sin que éste cayera en sentimientos de frustración al no lograr el objetivo, o incluso que debieran recurrir a técnicas como el ensayo y error para el desarrollo de su proyecto. Según Stephen Cooper – citado en Utting y colaboradores (2010) – el tipo de estrategia basado en ensayo y error no es conveniente trabajarlo frecuentemente puesto que no genera una construcción sólida de conocimiento, y es mejor que los estudiantes desarrollen otras estrategias exitosas hacia la resolución de problemas.

Igualmente, existen otras formas adicionales que han sido aplicadas exitosamente en la enseñanza por medio de tecnologías de la información, no solo aplicados en la temática de la programación, sino también en otros contextos. Por ejemplo, Wen-Yu Lee y Tsai (2013) en su revisión de literatura, encontraron adicionalmente varios tipos de técnicas de enseñanza que implicaban en uso de tecnologías de la computación, y especialmente para fomentar habilidades de orden superior como las habilidades de resolución de problemas o incluso el pensamiento crítico, tales como el uso de simulaciones, los materiales multimedia, los sistemas integrados, cursos mixtos, los tutoriales y las evaluaciones asistidas por computador, entre otros. Sin embargo, estas técnicas de enseñanza requieren esfuerzos

interdisciplinarios mayores y tiempos mayores de preparación curricular que el que podría generar el desarrollo de proyecto o el uso de un juego existente, por lo cual no se tuvieron en cuenta estas técnicas.

Uno de los aspectos que se tomaron en consideración para la aplicación de este proyecto fue la revisión de restricciones que se presentaron, ya que se tuvo que buscar no solo que hubiera facilidades en su desarrollo sino también que tanto el docente como los estudiantes no perdieran su motivación en explorar esta nueva temática con la aplicación de una tecnología que fue novedosa tanto para el docente que monitoreaba la clase, como para el estudiante que recibió este conocimiento. En un informe presentado por Osborne y Hennessy (2003), se encontró que dentro de las restricciones más importantes se presentan la falta de tiempo –tanto por parte del docente como del estudiante- para lograr adquirir confianza y experiencia con el uso de la tecnología, el acceso limitado a recursos, un currículo adicional sobrecargado con contenido que no esté enfocado correctamente al uso de la tecnología, y la falta de una guía específica para utilizar la tecnología para soportar adecuadamente el aprendizaje.

2.6 MODELOS DE EVALUACIÓN

En los experimentos de Agina (2012), se tomaron en cuenta una serie de variables en la realización de sus pruebas, como la cantidad de ejercicios ejecutados, el tiempo que le tomó desarrollar cada ejercicio, el número de respuestas correctas e incorrectas y el número de tareas que no pudo completar, entre otros. En ese mismo experimento se tomó una variable adicional para la medición del pensamiento creativo en la resolución de problemas, como la necesidad de los estudiantes en la solicitud de una ayuda extra por parte del docente para explicar un problema particular de la prueba.

Otra de las variables adicionales utilizadas tiene que ver con la sensación de satisfacción que causaba la aplicación de las pruebas. Eso se logró mediante una serie de preguntas posteriores, con las cuales se indagaba si las pruebas eran, para su concepto, entendibles y fáciles de realizar, entre otras. Esta determinación adicional de la sensación que causa esta actividad en los estudiantes se vio

también en el experimento de Giannakos y Jaccheri (2013), en el cual se realizaban preguntas para determinar el grado de satisfacción, utilidad y facilidad de las pruebas. Dichas variables ayudaron a determinar si habían adquirido intrínsecamente la habilidad para comprender los problemas que se le presentaron en la prueba. Liu, Cheng y Huang (2011) hablaron sobre esta sensación de satisfacción como un estado mental, conocido como "flujo", en el que puede estar un estudiante con el fin de generar un mejor aprendizaje y una mejor forma de resolver problemas. Este mismo estado se nombró en la investigación de Kirriemuir y McFarlane (2004), quienes argumentaron que es en este estado de "flujo", en el que se genera un mayor interés por parte de los estudiantes haciendo que incluso se vuelvan ajenos a las distracciones.

Liu, Cheng y Huang (2011) presentaron una forma de medir este estado de flujo basado en el modelo de flujo de 3 canales (aburrimiento, ansiedad, estado de flujo) al determinar un balance entre niveles de habilidades percibidas y retos percibidos.

3. MARCO METODOLÓGICO

3.1 DESCRIPCIÓN DEL MÉTODO

Con el fin de comprobar la veracidad de la hipótesis planteada, se estableció como propósito del desarrollo de esta tesis determinar la relación directa entre el aprendizaje de la técnica algorítmica de resolución de problemas impartida durante un curso de programación dado a un grupo de estudiantes de 5° grado de una institución de educación básica primaria en el área metropolitana de Bucaramanga, y el mejoramiento de la habilidad matemática en la resolución de problemas. Para la enseñanza de las técnicas algorítmicas se determinó apoyarse en el uso de la tecnología por medio de una o varias herramientas software, aplicando estrategias instruccionales adecuadas, y planteando un conjunto de actividades que fueron aplicados dentro de un plan curricular propuesto para un curso.

Para determinar esta relación, se realizó un conjunto de pruebas de desempeño y se generaron indicadores de medición para encontrar diferencias significativas en cuanto a los tiempos de respuesta de los estudiantes para resolver problemas matemáticos, la presentación de procedimientos correctos desarrollados y la precisión de los resultados presentados luego de aplicado dicho curso, para evidenciar mejoras en la capacidad de resolución de problemas de un grupo de estudiantes de 5° de primaria sometidos al contenido temático modificado para el área de tecnología en una institución de enseñanza básica primaria de Bucaramanga.

La metodología de desarrollo del proyecto constó de dos partes principales. En la primera parte se generó la preparación del conjunto de actividades y recursos a ser incorporados en el plan curricular. Durante esta fase se tomó como base el estado del arte para determinar mediante una revisión comparativa las técnicas aplicadas en la enseñanza de la programación en niños, la herramienta o herramientas software a utilizar para su implantación en el colegio y las temáticas a tratar en la enseñanza de la programación. Estos elementos conformaron el primer entregable propuesto en la entrega de este proyecto.

Dichos elementos fueron incorporados a un plan curricular para el nivel académico de 5° grado de educación básica primaria, con período de aplicación de dos meses, que complementó el área de Tecnología e Informática. Este plan curricular se trabajó en colaboración con la docente Carmen Alicia Albarracín, docente en el área de tecnología de la institución en la que se aplicó el proyecto, quien cuenta con una amplia experiencia pedagógica, con el fin de cumplir con los elementos psicopedagógicos de enseñanza adecuados al cumplimiento de las competencias requeridas. Este plan curricular propuesto conformó el segundo entregable propuesto en la entrega de este proyecto.

En la segunda parte se generó la aplicación del curso y un grupo de pruebas de desempeño para resolución de problemas matemáticos. Se realizó una prueba previa a la impartición de la temática en dos grupos de estudiantes: el grupo experimental al cual se presentó el contenido temático y un grupo de control en el que no se impartió el contenido. Se hizo además una prueba final en el que se impartió el contenido académico, la cual contempló una serie de ejercicios para resolución de problemas de un nivel mayor pero requiriendo las mismas competencias matemáticas para ambos grupos. Los resultados de este experimento se contrarrestaron en cada una de las pruebas se procesaron mediante herramientas estadísticas. La aplicación del curso y el análisis de resultados conformaron respectivamente el tercer y cuarto entregables para el cumplimiento del desarrollo del proyecto.

3.2 ESTUDIO COMPARATIVO DE TÉCNICAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS

Basado en la clasificación propuesta en el marco teórico de las 6 técnicas para la enseñanza de la programación en los niños, se realizó un análisis de cada una de las técnicas y a partir de allí seleccionar la herramienta o herramientas más adecuadas a ser utilizadas en este proyecto.

3.2.1 Uso de videojuegos para la enseñanza

Esta técnica se basa en el uso de aplicaciones tipo juego que muestran un objetivo particular a través de una historia de fondo, y con un conjunto de niveles de dificultad que debe cumplir, todo basado a través del planteamiento de instrucciones ejecutadas en forma secuencial. En las experiencias se pudo observar que estas técnicas fueron muy sencillas y rápidas de aplicar dada la facilidad de uso y la sencillez de la interfaz de los juegos. Por esta razón se pudo enseñar incluso a estudiantes menores de 10 años incluso en edad preescolar. Otra ventaja que tiene el uso de los videojuegos es su capacidad de generar y mantener la motivación en los que juegan. Adicionalmente se tiene que no se requiere un alto fundamento teórico inicial dada la facilidad del tema. Haciendo una revisión de los temas que se tocan en lo referente a algoritmos, estos videojuegos trataron los siguientes temas básicos:

- Secuencialidad, diseño de órdenes (concepto de algoritmo)
- Comprensión de sintaxis (lenguaje formal y reglas de construcción)
- Depuración (corrección de errores) pre-ejecución, en ejecución y post-ejecución.
- Subrutinas

Se observó que en estas experiencias no se dedican a explicar en forma concreta tipos de estructuras de control como condicionales o ciclos, aunque se usó el concepto de subrutinas como estructuras de reutilización de órdenes o repetición de procesos. Tampoco se le dio mayor relevancia a los conceptos de uso de datos y su almacenamiento en variables, ni se presentaron instrucciones de entrada o salida de información. Estas razones convierten esta técnica es un método efectivo para la enseñanza de estos temas incluso desde temprana edad. Sin embargo, la limitación de la temática podría llegar a convertirse en una desventaja dado que se limita la teoría a enseñar, pudiendo generar un avance lento y una desmotivación en los niños si se maneja esta misma técnica durante un largo tiempo. Esta técnica se recomienda entonces cuando se desee enseñar los fundamentos de algoritmos a los niños en un corto período de tiempo.

Con el fin de obtener otras herramientas que aplicasen esta técnica, se encontró un conjunto de aplicaciones software de tipo juego que se están utilizando para la enseñanza de la programación. Varios de ellos se están utilizando actualmente en

el proyecto “Hour to Code”, una iniciativa mundial que busca que millones de estudiantes alrededor del mundo puedan trabajar durante una hora de ciencias de la computación y la programación. Esta iniciativa es desarrollada por Computer Science Education Week (2014), un programa estadounidense dedicado a inspirar a los estudiantes de todos los niveles académicos para que se interesen en las ciencias de la computación y Code.org (2013), una organización sin ánimo de lucro que busca incrementar la participación de las ciencias de la computación en las escuelas de Estados Unidos, haciendo que en cada institución educativa se cuente con la posibilidad de aprender programación de computadores, y promocionando que estos tipos de actividades sean incluidos en el currículo básico en los colegios, junto con las clases de ciencias, tecnología ingeniería y matemáticas. Dentro de los juegos que se usan en esta iniciativa se encontró uno llamado LightBot, un videojuego publicado por la empresa Armor Games (2005), que trata sobre un robot que sigue un conjunto de instrucciones secuenciales con el fin de desplazarse por un campo tridimensional para encender un grupo de luces en el suelo. Cuenta una interfaz muy sencilla de aprender y una metodología de juego muy parecida a la presentada por los juegos PiktoMir y Gidget, por lo que fue fundamental ubicar este juego en internet para su descarga y uso.

Otro juego que se utilizó en esta iniciativa fue el entorno Blockly, que cuenta con una interfaz muy parecida al usado por Scratch, y cuenta con un conjunto de opciones de juego para seleccionar. Una de estas opciones se le conoce como “Laberinto”, en el cual se trata también de desplazar a una figura hacia un punto determinado a través de una secuencia de instrucciones colocadas ordenadamente.

Se encontró otro juego llamado Manufactoria que trata sobre ayudar a una empresa a controlar la producción de robots creando una estructura de cintas de transporte con un sistema de pruebas para determinar cuáles robots deben deshacerse o no. Para determinar si el robot debe deshacerse, se debe analizar su “estructura de programación”, la cual es una secuencia de colores. Los robots deben tener una secuencia particular, si cumplen se debe llevar con el sistema de cintas hacia la salida, de lo contrario se deben sacar de la cinta de transporte. La interfaz es más sencilla pero su complejidad es algo mayor, además de sus instrucciones, las cuales vienen en inglés.

3.2.2 Creación de proyectos con temática fija

Esta técnica consiste en la creación de varios tipos de proyectos en los que se utilicen simuladores o aplicaciones que permitan estructurar el entorno de trabajo para generar situaciones que planteen problemas particulares para su resolución a través del razonamiento lógico aplicando un proceso algorítmico.

Esta técnica muestra particularmente una limitante de trabajo hacia una temática muy particular. Sin embargo, una gran ventaja de aplicar esta técnica es la motivación adicional que genera el uso de estos elementos puesto que permiten crear situaciones nuevas no restringidas sino por su temática. Otra característica encontrada en esta técnica es que en estas experiencias ya se abordan temas adicionales a los referidos en la primera técnica:

- Ideación de modelos computacionales
- Uso de variables numéricas
- Entendimiento de respuestas del sistema
- Estructuras de control: condicionales, iteraciones, anidaciones
- Transición a un lenguaje de programación

En cuanto a los elementos hardware, se observó en las experiencias encontradas la motivación en los estudiantes en el uso de manipulativos físicos. Sin embargo no se puede perder el horizonte usando solamente los manipulativos físicos sin enseñar conceptos teóricos por lo que se requeriría para este tipo de elementos complementar con formación teórica. Al seleccionar los elementos hardware se podría contar con una limitante que es la no disponibilidad a estos recursos de una forma fácil, así como también el requerir personal especializado o capacitado en estos elementos hardware, y además, la necesidad de capacitar al personal docente en este tipo de técnicas.

Dado que no fue posible encontrar elementos hardware que suplieran esta capacidad de aplicación de esta técnica, se recurrió a encontrar elementos software que permitieran manipular el entorno de trabajo para la generación de diversas situaciones que pudieran ser resueltos a través del razonamiento lógico, y que contara con la presentación las temáticas nombradas previamente. Se encontraron juegos de desarrollo libre con una temática fija de construcción de dibujos, en los que a pesar que no son videojuegos por no contar con un objetivo

establecido ni un conjunto de niveles a superar, se podían generar entornos y desarrollar situaciones particulares. Estos elementos no son nuevos, se ha trabajado desde hace mucho tiempo con estos tipos de software; uno de ellos ha sido el clásico lenguaje de programación Logo, con el cual se maneja una tortuga mediante la escritura de una serie de instrucciones para que la tortuga realice un dibujo, usando para ello un lápiz que marca el piso mientras ésta se desplaza por el terreno. No se logró ubicar el entorno original Logo, pero se encontró una réplica llamada Turtle Graphics, una aplicación desarrollada en Java que realiza los mismos procesos que el original. Blockly cuenta también con un nivel llamado Tortuga Gráfica que tiene la misma metodología, pero se programa a través de bloques gráficos. Otro programa que se encontró se llama RoboMind, un entorno en el cual se programa un robot para desplazarse por un terreno y poder, no solamente pintar en el suelo sino también agarrar y desplazar elementos de sitio, y determinar su movimiento a través de unos “sensores” que permiten identificar si los lugares adyacentes cuentan con algunas características particulares para ejecutar alguna actividad.

Dado que el desplazamiento de los elementos en estos programas es controlado por las instrucciones que se le ingresen sin necesidad de cumplir con un objetivo determinado, podría volverse una desventaja con respecto a la técnica anterior ya que no cuenta con un reto de superación de niveles ni un objetivo definido y no genera una motivación por sí sola, por lo cual se sugeriría complementar la aplicación de estos programas con la presentación de ejercicios y retos que deban cumplir los estudiantes. Esto implicaría una inversión de tiempo mayor para la preparación de estos recursos.

3.2.3 Creación de entornos para generación de situaciones

Esta técnica utiliza una herramienta de desarrollo para crear todo un ambiente particular, con el fin de generar situaciones interesantes en las cuales los niños pueden explorar y manipular a través de la programación. En esta técnica las posibilidades de desarrollo son mucho menos limitadas que las anteriores. De hecho, en estas experiencias se abordaron un conjunto adicional de temas referentes a la programación de computadores:

- Tipos de variables (numéricas, textuales, booleanas)
- Reusabilidad de ejemplos y experimentación
- Ideación y formulación de modelos computacionales
- Analogías con escenarios reales
- Arreglos y matrices
- Funciones y expresiones matemáticas
- Gráficas y coordenadas cartesianas
- Interactividad con teclado y mouse, programación de eventos
- Orientación a objetos, clases, herencia
- Simulación de comunicación (sincronización)
- Actividades de no programación (diseño de escenarios e interfaz)

Sin embargo, la aplicación de esta técnica implica que se requeriría mayor tiempo de preparación previa de elementos audiovisuales dado el grado de libertad con el que cuenta, además de la vasta cantidad de funciones adicionales que se pueden utilizar, y por ende se requiere de un tiempo adicional de preparación teórica para los estudiantes y por supuesto para el docente.

3.2.4 Creación de videojuegos

Esta técnica también permite crear un entorno completo para establecer situaciones de juego que deben controlarse a través de procesamientos secuenciales lógicos y/o matemáticos. Esta técnica implica particularmente enseñar a los estudiantes conceptos adicionales sobre creación de juegos. Además, se observó en estas experiencias que para su aplicación se debe contar con muchos recursos adicionales no solo de elementos software y archivos para su desarrollo como imágenes, animaciones, sonidos sino de otros factores de diseño que implica la realización de un videojuego como la creación de las historias de juego, la creación de los personajes con sus acciones, la creación de los objetivos, las reglas y los niveles. Esto implica contar un tiempo mucho mayor de preparación inicial y de aplicación de estos proyectos, no solo por parte de los estudiantes sino también por parte del docente con el fin de no estar “patinando” en proyectos de manera improvisada o generar actividades con poco avance, y concentrarse en el uso y la enseñanza de la programación.

3.2.5 Uso de herramientas computacionales como apoyo a la resolución de problemas

Esta técnica consiste en usar apropiadamente algunas herramientas tecnológicas como soporte en la enseñanza de otras temáticas que presenten situaciones de resolución de problemas. Varias de herramientas software son utilizadas como apoyo en la enseñanza de las matemáticas, permitiendo generar modelos matemáticos a partir del ingreso y manipulación de diversas ecuaciones y variables y realizando ciertas codificaciones se podrían presentar casos de estudio de problemas reales y comprender determinadas situaciones o encontrar resultados adecuados.

A pesar de la efectividad de la técnica, se observó que su fundamentación no está basada en la enseñanza de la programación, sino más bien se convierte en un complemento para enseñar a manejar una herramienta que le permite manipular datos para comprender y resolver problemas. Estas herramientas tienen una complejidad algo mayor en su uso aunque no es necesario alejarse tanto en la selección de herramientas especializadas. Un referente cercano podría ser el uso de Microsoft Excel y sus procesos de manipulación de datos a través de fórmulas y macros que se pueden implementar en un proyecto específico. Sin embargo, esta técnica se desvía del objetivo directo buscado el cual es la enseñanza de la programación. Además, en esta técnica también es necesario contar con suficiente tiempo para preparar casos de estudio basados en datos y situaciones que presenten un reto de solución, además organizando dichos proyectos para que puedan enseñarse adecuadamente los constructos de programación con el orden de complejidad adecuado.

3.2.6 Generación de torneos o competencias

Es una técnica que permite a través de torneos competitivos generar un método de enseñanza de la programación, dado que se explican conceptos de información, comunicación, TIC, resolución de problemas, pensamiento algorítmico, flujos, entre otros, para que los estudiantes obtengan las bases de la programación y permitan desarrollar la habilidad para resolver los problemas que

se presentan en los torneos. En esta técnica es muy necesario realizar un proceso de adaptación de datos, recategorización de problemas, simplificación de tareas, así como rediseñar imágenes e implementar nuevo software, basados en los temas adquiridos en un currículo ya establecido. La adaptación al nivel de educación básica y la implementación de software generan una considerable inversión de tiempo dada la cantidad y la complejidad de las tareas a presentar.

3.2.7 Comparativo de técnicas

Luego de esta caracterización de las 6 técnicas encontradas en cuanto a temáticas tratadas y contextos, técnicas y modelos pedagógicos aplicados, tipo de ambiente en el que se trabajó y resultados obtenidos, se procedió a realizar un comparativo con el fin de obtener una compilación de actividades y características para establecer un marco de trabajo base para su implementación en el aula. Este cuadro presenta una síntesis general de ventajas, requisitos y herramientas adecuadas para la aplicación de estas técnicas.

Tabla 3. Comparación de ventajas, desventajas y listado de herramientas por técnica

Técnica y experiencias	Ventajas	Requisitos	Herramientas
Uso de videojuegos para la enseñanza Rogozhkina & Kushnirenko (2011); Lee & Ko (2011)	Preparación rápida, no requiere trabajo especializado Software sencillo de usar y asequible Se puede complementar con formación teórica básica Genera motivación dado la característica de trabajo con juegos		PikoMir Gidget LightBot Blockly (Laberinto) Manufactoria
Creación de proyectos con temática fija Liu, Cheng, & Huang (2011); Valente (2004); Sipitakiat & Nusen (2012); Zuckerman,	Genera mayor motivación para los niños dado el trabajo con manipulativos Se puede complementar con formación teórica	Puede requerir hardware especializado Puede requerir preparación especializada por parte del docente (referente a	Train B&P Turtle Graphics Blockly

<p>Arida, & Resnick (2005); Tarkan, y otros (2010)</p>	<p>básica</p>	<p>hardware)</p>	<p>RoboMind Karel J Robot</p>
<p>Creación de entornos para generación de situaciones</p> <p>Felleisen, Findler, Flatt, & Krishnamurthi (2009); Wolz, Leitner, Malan, & Maloney (2009); Burke & Kafai (2010); Kelleher, Pausch, & Kiesler (2007); Meyers, Cole, Korth, & Pluta (2009); Rizvi, Humphries, Major, Jones, & Lauzun (2011)</p>	<p>Posibilidad de trabajo ilimitado</p>	<p>Requiere amplia preparación por parte del docente para proponer diferentes entornos</p> <p>Requiere complemento teórico extenso</p>	<p>EToys Scratch Alice Dr.Scheme Processing</p>
<p>Creación de videojuegos</p> <p>Doerschuk, Liu, & Mann, (2012), Javidi & Sheybani (2009), Rodger, y otros (2010)</p>	<p>Posibilidad de trabajo ilimitado</p>	<p>Requiere preparación especializada por parte del docente para proponer y desarrollar entornos de juego</p> <p>Requiere recursos adicionales (imágenes, animaciones)</p> <p>Requiere tiempo de preparación para las historias de juego, los objetivos y las reglas</p> <p>Requiere complemento teórico extenso</p>	<p>Greenfoot Kahootz Squeak Alice</p>
<p>Herramientas computacionales como apoyo a la resolución de problemas</p> <p>Abramovich (2013)</p>	<p>Posibilidad de trabajar con datos y situaciones reales, aunque podría limitarse la presentación de temas de programación de computadores</p>	<p>Requiere software especializado, aunque es posible usar herramientas ofimáticas</p> <p>Requiere preparación del docente para proponer</p>	<p>GeoGebra Rhinceros VisualMath Mathematica</p>

		distintos entornos	Microsoft Excel
Generación de torneos o competencias Tomcsányiová & Tomcsányi (2011)	Genera una mayor motivación dada la característica de uso de juegos para competencias	Requiere contar con conocimiento previo generado por la aplicación de un currículo Requiere software especializado Requiere tiempo de preparación para organizar competencias	No hay herramienta especial, se deben crear de acuerdo a las tareas presentadas en la competencia.

Fuente: Autor del proyecto

No se pretendió indicar en este comparativo que una técnica haya sido mejor que otra. Por el contrario, todas han demostrado su efectividad en sus ambientes de aplicación y es posible generar una propuesta curricular más completa tomando como referentes estas técnicas encontradas. Sin embargo, dadas las limitaciones de tiempo, de recursos actuales y la facilidad para apropiarse de la tecnología a utilizar, se seleccionó el uso de videojuegos para su aplicación en el Colegio San Juan de Girón durante un período de dos meses, tiempo con el que se contó para la ejecución de este proyecto, además que se presentan ventajas pedagógicas en el proceso de enseñanza a través de esta técnica.

De acuerdo a los conceptos presentados por Mitchell y Savill-Smith (2004), es frecuente el uso de los juegos como herramientas valiosas en el mejoramiento del aprendizaje, siendo vistos como medio para incentivar a los estudiantes que no tienen la suficiente confianza en sí mismos, también como para reducir tiempo de entrenamiento y carga instruccional al establecer una oportunidad de aplicar la metodología de "drill and practice", una forma de instrucción particular que permite que los estudiantes ensayen con un conjunto de elementos que siguen un patrón común. Adicionalmente, los autores sugieren que esta técnica mejora la retención y la adquisición de conocimiento.

3.3 ESTUDIO COMPARATIVO DE HERRAMIENTAS PARA LA ENSEÑANZA DE LA PROGRAMACIÓN PARA NIÑOS

Dada la selección de la técnica del uso de videojuegos, se procedió a realizar una búsqueda y comparación de varias herramientas software que pudieran servir para este propósito. Con el fin de determinar parámetros adecuados para la comparación de herramientas se revisó nuevamente los artículos de Tarkan y colaboradores (2010), Kelleher y Pausch (2005), Lamerás y colaboradores (2010), y Schwartz, Stagner y Morrison (2006).

En el artículo de Tarkan y colaboradores (2010), se habla de una clasificación inicial de herramientas de programación para la enseñanza, los lenguajes de programación visuales, y los sistemas de programación tangibles. Dado que los sistemas de programación tangibles no se encuentran dentro de la técnica que se seleccionó, esta clasificación no se tuvo en cuenta para la determinación de los parámetros.

Se realizó una compilación y clasificación de los parámetros encontrados en los otros 3 artículos, generando la siguiente tabla:

Tabla 4. Comparativo de evaluaciones de herramientas usadas para enseñanza de la programación

Área de análisis	Kelleher y Pausch (2005)	Lamerás y colaboradores (2010)	Schwartz, Stagner y Morrison (2006)
Usabilidad	Forma de presentación del código Forma de construcción de código	Soporte de programación visual Rangos de edades indicados Idiomas que soporta	Accesibilidad y usabilidad Internacionalización
Codificación	Soporte adicional para comprender comportamiento de códigos Prevención de errores en la interfaz de programación Intento explícito de hacer más entendible el lenguaje	-	Simplicidad en la codificación Recompensa al presentar resultados inmediatos Funcionalidad máxima usando mínima codificación
Estilo de	Estilo de programación	Soporte de orientación a	Consistencia con diseños

programación		objetos Disponibilidad como IDE	modernos de software
Enfoque de uso	Uso adicional (diversión, motivación, resolución de problemas, sistema de enseñanza)	-	Diversión que generan Atracción
Enseñanza	Constructos de programación	-	Capacidad de progresividad para la enseñanza Adecuada preparación que genera para permitirle conocer otro lenguaje de programación,
Requerimientos técnicos	-	Gratuito y/o de código abierto Existencia de versiones estables Soporte de localización Sistema operativo	-
Soporte cooperativo	Soporte colaborativo	Desarrollos en comunidades activas Documentación disponible	Capacidad de ser publicable Soportabilidad

Fuente: Autor del proyecto

Puesto que varios de estos parámetros tratan aspectos muy parecidos, se reorganizaron y se obtuvieron los siguientes:

Tabla 5. Parámetros de evaluación de herramientas usadas para enseñanza de la programación

Usabilidad	Presentación y construcción de código en forma textual Presentación y construcción de código en forma visual Presentación de interfaz general Idioma que soporta Rango de edad indicado
Codificación	Comprensión de código (explicación de cada comando) Sencillez o simplicidad de codificación (facilidad de aplicación) Inmediatez de resultados Prevención de errores
Estilo de programación	Orientación a objetos (consistente con diseños modernos de SW) Disponibilidad como IDE Estilo especial de programación
Enfoque de uso	Motivación (por diversión) Resolución de problemas
Enseñanza	Constructos de programación Capacidad de progresividad en la enseñanza Adecuada preparación para otro lenguaje

Requerimientos técnicos	Gratuito o de código abierto Versiones estables Soporte de localización Sistema operativo
Soporte cooperativo	Documentación Publicabilidad Soporte colaborativo

Fuente: Autor del proyecto

Se realizó la búsqueda de los 5 videojuegos que se presentaron y se ajustaron a las características descritas en la técnica:

- PiktoMir
- Gidget
- LightBot
- Blockly (Laberinto)
- Manufactoria

No se logró contar con Gidget dado que a la fecha no se liberó la versión con licencia de código abierto. Con los otros 4 programas se realizó una exploración de funcionalidad. Las herramientas presentan una interfaz visual para construir el código mediante el arrastre de íconos hacia la sección de codificación. Los íconos son fácilmente entendibles y se brinda motivación al presentar objetivos y niveles para superar, siendo adecuados para el uso con los niños. Estas herramientas presentan el resultado de la programación, mostrando el comportamiento del robot en forma inmediata presionando un solo botón, es decir, sin necesidad de pasar por una fase de compilación previa. Únicamente Blockly previene algunos de los errores generales de construcción en la programación dado que sus íconos cuentan con una forma de bloques con formas que permiten encajar solo con ciertas combinaciones, además que es el único que presenta cada ícono con un sencillo pseudocódigo textual muy entendible. No se encuentran diferencias en los estilos de programación en las 4 aplicaciones, aunque ninguno de ellos cuenta con un entorno de desarrollo para crear aplicaciones. Los temas que se presentan para la enseñanza de la programación se tocan de manera similar en todas las aplicaciones, y se observa el progreso en la enseñanza de los constructos mientras se avanza de niveles. En cuanto a los requerimientos técnicos, se encontraron las versiones para Windows en las 4 aplicaciones, y los programas se

presentan gratuitos en sus páginas principales, pero sus licencias no permiten realizar comercialización de estas aplicaciones. Ninguna de estas aplicaciones cuenta con un manual de usuario alterno a las indicaciones que presenta el mismo programa, y no permiten trabajo colaborativo. A continuación se describe las revisiones hechas a cada herramienta, presentando particularidades encontradas.

3.3.1 Revisión de Piktomir

El programa no requiere instalación, viene en una versión empaquetada que se encuentra en la página oficial, el cual se descomprime y se presenta listo para su uso. El programa se presentó inicialmente en ruso, y dado que no se encontró un manual de usuario para comprender las opciones y funciones de juego, se exploró a tanteo durante varios minutos su funcionamiento, encontrando la configuración de idioma para inglés o ruso. Durante los niveles no se presentó una especie de tutorial de funcionamiento del programa. Luego de la exploración se encontró un detalle en el programa: cuando se presiona el botón azul de revisión paso a paso y luego se presiona el botón verde de ejecución completa, la programación del robot se reinicia a la primera orden desde el puesto actual en el que se encuentra ubicado, mostrando un resultado de ejecución no esperado. Esto podría desorientar al jugador presentando un error en la programación cuando realmente no lo hay; por lo tanto se contaría con una restricción de uso el cual sería que únicamente pueden utilizar exclusivamente el botón verde de ejecución completa o el botón azul de paso a paso. En el nivel 9, el programa dejó de funcionar, lo cual se considera que la versión para Windows está todavía inestable, a pesar que ya se encuentran versiones para otros sistemas operativos (Mac, Linux, Android).

3.3.2 Revisión de Lightbot

La primera versión de Lightbot se encontró disponible para jugar online en el sitio de juegos gratuitos Armor Games. Su creador, Danny Yaroslavski (conocido como CoolioNiato), se vinculó a la iniciativa “Hour to Code” y puso a disposición su propia marca y una versión modificada que cuenta con un tutorial más completo y soporte de idiomas. Se encuentra disponible online, para iOS y Android, pero no se encontró una versión offline. Se encontró en internet una versión offline de la

primera versión y se trabajó con esta versión para eliminar la condición de conexión a internet. La única restricción que se tuvo para su funcionamiento es el requisito de instalación de Adobe Flash Player para su visualización. En el sitio Armor Games también se encontró la versión 2.0, que viene con más funciones como creación de niveles, pero no se logró encontrar la versión offline. La versión instalada solo viene en inglés, pero en los niveles se presenta un tutorial muy sencillo para el uso de los comandos.

3.3.3 Revisión de Manufactoria

Manufactoria fue creado en 2010 por Nicholas Feinberg, montado como una aplicación para uso gratuito no comercial para web en la página Pleasingfungus Games. La temática es un poco más compleja que en los otros porque a diferencia de generar el movimiento de un robot para que cumpla con un objetivo, se debe crear un sistema de transporte mediante cintas transportadoras, bifurcadores, entre otros elementos para analizar patrones de símbolos que vienen en cada robot, determinando cuáles pueden pasar o no de acuerdo al criterio pedido. La aplicación muestra una gran cantidad de texto en inglés para explicar el objetivo del juego y las funciones de cada instrucción, por lo que no se explica de una forma sencilla el funcionamiento de los comandos. Los colores principalmente usados en Manufactoria son los grises, y no se muestra una interfaz agradable ni colorida que permita generar una buena motivación en el niño. Al igual que la versión de Lightbot, se necesita Adobe Flash Player para su visualización.

3.3.4 Revisión de Blockly

Blockly fue concebido como un editor de programación en forma gráfica basado en la web, siendo esta una base fundamental para la creación de una amplia variedad de proyectos, incluyendo herramientas educativas tales como MIT App Inventor, BlocklyDuino, Romotive, GigaBryte, entre otros. Su interfaz es muy parecida a la que se presenta en Scratch en cuanto a la forma de trabajo uniendo bloques, pero más limpia (es decir, sin tantos comandos que saturan la pantalla y distraigan al usuario) dependiendo del nivel de juego en el que se desee trabajar. Se encontró una versión para trabajar offline pudiendo ser visualizado desde cualquier

navegador, sin embargo presenta incompatibilidad con Internet Explorer, por lo que recomiendan trabajar con otro. Dentro de las ventajas que se encontró con esta versión ha sido el idioma español, y la presentación del código mediante texto con una sintaxis muy entendible, además de los íconos en forma de bloques.

3.3.5 Comparación de herramientas

Luego de estas revisiones se realizó una tabla evaluativa que permitiera seleccionar la herramienta o herramientas más adecuadas para ser usadas con la técnica de uso de videojuegos, tomando como referencia los parámetros determinados previamente:

Tabla 6. Evaluación de herramientas para implementar la técnica de uso de videojuegos en la enseñanza de la programación

Característica a evaluar		Piktomir	LightBot	Manufactoria	Blockly
Usabilidad	Presentación y construcción de código en forma textual	0	0	0	1
	Presentación y construcción de código en forma visual	1	1	1	1
	Presentación de interfaz general	1	1	0	1
	Idioma que soporta	0	0	0	1
	Rango de edad indicado	1	1	1	1
Codificación	Comprensión de código (explicación de comandos durante el juego)	0	1	0	1
	Sencillez o simplicidad de codificación (facilidad de aplicación)	1	1	0	1
	Inmediatez de resultados	1	1	1	1
	Prevención de errores	0	0	0	1
Estilo de programación	Orientación a objetos (consistente con diseños modernos de SW)	0	0	0	0
	Disponibilidad como IDE	0	0	0	0
	Estilo especial de programación	0	0	0	0

Enfoque de uso	Motivación (por diversión)	1	1	1	1
	Resolución de problemas	0	0	0	0
Enseñanza	Constructos de programación	1	1	1	1
	Capacidad de progresividad en la enseñanza	1	1	1	1
	Adecuada preparación para otro lenguaje	0	0	0	0
Requerimientos técnicos	Gratuito o de código abierto	1	1	1	1
	Versiones estables	0	1	1	1
	Soporte de localización	0	0	0	0
	Sistema operativo	1	1	1	1
Soporte cooperativo	Documentación (manual de usuario)	0	0	0	0
	Publicabilidad	0	0	0	0
	Soporte colaborativo	0	0	0	0
		10	12	9	15

Fuente: Autor del proyecto

Se tomó como valores de calificación los números 0 y 1 para una mayor facilidad, indicando 1 como existencia o cumplimiento del parámetro en la aplicación y 0 como no existencia o falla en el parámetro evaluado. No se tomaron en cuenta importancias particulares ni por parámetros ni por categorías dado que los parámetros más importantes tuvieron valores similares en todas las aplicaciones y las diferencias no serían tan notorias si estos parámetros tuvieran pesos mayores que los otros, por eso se determinó que se tomaría en cuenta la cantidad de diferencias total. Al sumar los totales se encontró que las aplicaciones Lightbot y Blockly son las que actualmente podrían generar un mejor trabajo para la técnica del uso de videojuegos, comparadas con las versiones descargadas de los otros dos programas. No se deja por sentado que una aplicación sea mejor que otra, sino que con objeto de este proyecto, Lightbot y Blockly fueron las más adecuadas para poder aplicar la intervención planteada. Con la determinación de la técnica óptima, de las herramientas adecuadas y las temáticas a tratar para la aplicación de este proyecto, se dio por cumplido el primer objetivo específico, procediendo a desarrollar la propuesta de plan curricular para el área de tecnología del grado 5° del Colegio San Juan de Girón sede B.

3.4 GENERACIÓN DE PLAN CURRICULAR PARA EL ÁREA DE TECNOLOGÍA

Un plan de área es un documento que presenta un esquema estructurado de un área temática que forma parte de un currículo en un establecimiento educativo, el cual contiene una serie de elementos entre los que se incluyen la identificación de los contenidos y los temas del área, las actividades pedagógicas requeridas, la metodología de aplicación, los materiales didácticos y recursos adicionales de soporte, la distribución del tiempo y secuencias del proceso educativo, los logros, competencias y conocimientos que se deben alcanzar, los criterios y procedimientos de evaluación, los indicadores de desempeño entre otros (Ministerio de Educación Nacional, 2013). Basados en ese concepto inicial se revisaron ejemplos de planes de área de algunos colegios, incluido el colegio donde se aplicó la prueba, y se revisaron artículos referentes a formas de generación de planes curriculares como los de Gómez (2010) y Amézquita Zárate, Contreras Pineda, & Pardo Romero (1997), y se complementó con las metodologías de generación de planes curriculares presentados por la FAO (Organización de las Naciones Unidas Para la Agricultura y la Alimentación, 1999).

En estos artículos se encontró que se deben analizar 5 aspectos importantes en el desarrollo de un plan curricular: los objetivos, los contenidos, las actividades, los recursos y los sistemas de evaluación. Para tener una validez pedagógica, se contó con la colaboración de la docente del área de tecnología, profesora Carmen Alicia Albarracín, para realizar el proceso de validación de las temáticas, los objetivos y las metodologías de trabajo. El resultado de esta revisión se plasma a continuación.

3.4.1 Objetivos

En este aspecto se determinaron los objetivos referentes a los logros a alcanzar en la aplicación de este curso piloto, pero no se incluyeron los referentes generales de los objetivos como fines de la educación que se presentan en el artículo 5° de la Ley General del Educación (Ministerio de Educación Nacional,

1994), dado que estos son comunes para todas las áreas. Según un documento de la FAO referente a la generación de planes curriculares (1999), antes de elaborar el plan de estudio, es necesario hacer la distinción de cuáles serán los objetivos de aprendizaje, con el fin de determinar los contenidos y actividades que estructurarán el plan. Estos objetivos de aprendizaje deben indicar lo que el estudiante conocerá (conocimiento), lo que el estudiante será capaz de hacer (aptitud) y lo que deberá ser (actitud) al terminar el curso. Además, los objetivos deben ser específicos, medibles, alcanzables, realistas y contar con un tiempo determinado. De acuerdo a esto se estableció y se depuró el objetivo principal del curso, el cual se presentó así:

- Al finalizar el curso, los estudiantes deberán ser capaces de identificar secuencias adecuadas de pasos lógicos para resolver diversos tipos de situaciones problema. Para ello, se buscará generar en los estudiantes el concepto de pensamiento algorítmico con el fin de desarrollar nuevas estructuras de aprendizaje aplicables en diversos contextos para la resolución de problemas.

3.4.2 Contenido

De acuerdo con Amézquita y colaboradores (1997), para lograr los objetivos se deben definir unos contenidos y una metodología. En este aspecto se identificó como primera medida el contenido a tratar, basado en los ejes temáticos identificados en la aplicación de la técnica seleccionada y complementados con las experiencias presentadas previamente:

- Concepto de algoritmo, diseño y construcción de algoritmos (secuencialidad)
- Determinación de pasos para resolver problemas
- Diseño y construcción de algoritmos mediante pseudocódigo (sintaxis, reglas de construcción)
- Datos variables y datos constantes
- Depuración, corrección de errores
- Funciones y recursividad (Subrutinas)

Dado el corto tiempo con el que se contó para la preparación y la aplicación (2 semanas de preparación y 2 meses de aplicación), se determinaron los temas y se planteó una estructura secuencial de temas teniendo en cuenta el incremento de los niveles de complejidad de éstos. Sin embargo, al hacer revisiones posteriores los temas se han ido complementando por el análisis de las actividades y la determinación del uso de recursos. Para cada tema se determinaron también los objetivos específicos a alcanzar.

3.4.3 Actividades

Al realizar la revisión de las actividades encontradas en las experiencias que aplicaron la técnica de uso de los videojuegos para la enseñanza, se logró determinar la forma de aplicación y las secuencias de aplicación de las actividades. También se revisó la experiencia en Colombia que desarrolló la Fundación Gabriel Piedrahita Uribe en el INSA (Instituto de Nuestra Señora de la Asunción - INSA, 2012) y varias de las actividades propuestas en Code.org. Martín Fraile (2011) realizó en su artículo una selección de prácticas basadas en experiencias docentes de educación primaria en España, sirviendo como soporte adicional para la determinación de varias de estas prácticas con el fin de incluirlas en actividades particulares. Las prácticas presentadas se clasificaron en metodológicas, organizativas y curriculares. En las prácticas metodológicas y curriculares se encuentran las prácticas de trabajo individual y en equipo, la verbalización o memorización de contenidos, la actividad creativa y experiencias generadas por los estudiantes, los sistemas de premios, recompensas y castigos, la preparación de lecciones por parte del docente, la generación de actividades encaminadas a generar motivación, la generación de exámenes y la coparticipación con la comunidad educativa. A partir de estas se lograron relacionar algunas prácticas metodológicas que se desarrollaron en las experiencias revisadas, concebidas bajo el lineamiento inicial de la técnica del uso de videojuegos para la enseñanza: motivar al estudiante al aprendizaje de la programación, no solo en el ámbito computacional sino en situaciones reales, presentarle retos incrementándole los niveles de dificultad, repetir los procedimientos de aplicación de pasos secuenciales por medio de ejemplos y trabajos tanto en el computador como trabajos lúdicos. Se determinó en forma más específica las secuencias de actividades y la distribución de tiempo.

3.4.4 Recursos a utilizar

A pesar que los juegos Lightbot y Blockly brindaron la forma de presentar las temáticas solicitadas, con Lightbot no se puede presentar el concepto de variable mientras que con un módulo de Blockly sí se puede presentar. Esto conllevó a determinar que en las primeras actividades se trabajara con Lightbot y en las posteriores con Blockly. Se determinaron las actividades que se desarrollarían utilizando los programas y se establecieron tiempos de aplicación. Además, se prepararon actividades escritas para complementar los trabajos en computador y determinar la apropiación de procedimientos.

3.4.5 Compilación de elementos del plan

Al final de la revisión de contenidos, actividades y recursos, se generó una compilación general de actividades que fueron agrupadas por unidades y organizadas en secuencia, con los recursos y los tiempos de aplicación requeridos de la siguiente forma:

Tabla 7. Estructura de contenidos, actividades y recursos para implementar en el plan curricular

UNIDAD 1: Concepto de algoritmo, diseño y construcción de algoritmos	
Tema 1: Presentación, importancia de la programación en la actualidad	
Objetivos:	Que el estudiante comprenda la importancia de aprender a programar en algún entorno computacional
Descripción de actividades:	Video para los niños (inglés subtulado en español) y orientado por el docente para presentar la importancia de la programación en la tecnología. Se brinda una charla sobre el uso de la tecnología en la actualidad.
Recursos:	Físicos: Sala de informática y salón de clases, proyector, computador para docente. Software: Recurso encontrado en YouTube: https://www.youtube.com/watch?v=UCJ_YipIPZw
Tiempo requerido:	1 hora

Tema 2: Características básicas de un algoritmo, comprensión de instrucciones para desarrollo de algoritmos	
Objetivos:	Que el estudiante comprenda el concepto de algoritmo como una secuencia finita de pasos formados mediante instrucciones definidas para el cumplimiento de un objetivo
Descripción de actividades:	<p>Actividad lúdica (juego la gallina ciega versiones individual y grupal): Se selecciona al azar un niño y se le explica que simulará ser un robot, el cual debe seguir instrucciones particulares de sus compañeros quienes pueden hablar para poder recoger una pelota y llevarla a una canasta (órdenes generadas desde un inicio hasta un final con un objetivo claro). En la versión grupal se usará un intérprete por grupo quien será el único que puede hablar mediante instrucciones estrictas como moverse al frente, girar, agacharse, recoger, saltar (una orden a la vez, orden entendida por robot) y dar las instrucciones que su grupo le indique, mientras que cada grupo deberá quedarse en completo silencio, indicando solo con sus movimientos al intérprete para que le dé la orden a su robot a cargo.</p> <p>Se brinda charla sobre las características de un algoritmo (secuencial, finito, entendible, útil)</p>
Recursos:	Físicos: Salón de juegos, salón de clases.
Tiempo requerido:	1 hora
Tema 3: Instrucciones secuenciales para la consecución de objetivos particulares	
Objetivos:	Que el estudiante pueda construir un algoritmo lógico para la consecución de un objetivo deseado
Descripción de actividades:	Explicación y uso videojuego Lightbot niveles 1 al 5: Se brinda una teoría del funcionamiento del juego, así como de los constructos de desarrollo (órdenes y ubicación de secuencias). Uso del videojuego nivel 6 en adelante por parte de los niños.
Recursos:	<p>Físicos: Sala de informática y salón de clases, proyector, computador para docente, computadores para estudiantes.</p> <p>Software: Lightbot</p>
Tiempo requerido:	1 hora
Tema 4: Resultados esperados, detección y corrección de errores	
Objetivos:	Que el estudiante pueda determinar el resultado de un algoritmo lógico, detectar y corregir errores para la consecución de un objetivo deseado
Descripción de actividades:	<p>Charla sobre depuración (trazabilidad de código)</p> <p>Ejercicios con versiones escritas de Lightbot. Se dará un problema para determinar su solución mediante la misma mecánica del juego Lightbot pero en versión escrita.</p>

Recursos:	Físicos: Salón de clases.
Tiempo requerido:	1 hora
Tema 5: Funciones de tipo subrutina, reutilización de instrucciones	
Objetivos:	Que el estudiante pueda reutilizar instrucciones en un algoritmo lógico por medio de implementaciones de funciones de tipo subrutina
Descripción de actividades:	Explicación y uso videojuego Lightbot niveles 6 en adelante Ejercicios con versiones escritas de Lightbot. Se dará un problema para determinar su solución mediante la misma mecánica del juego Lightbot pero en versión escrita.
Recursos:	Físicos: Sala de informática y salón de clases, proyector, computador para docente, computadores para estudiantes Software: Lightbot
Tiempo requerido:	1 hora
Tema 6: Aplicación: secuencias de actividades en la vida diaria	
Objetivos:	Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo como una secuencias de pasos formados mediante instrucciones definidas para el cumplimiento de un objetivo
Descripción de actividades:	Charla sobre actividades de la vida diaria Ejercicios con versiones escritas de algoritmos
Recursos:	Físicos: Salón de clases
Tiempo requerido:	1 hora

UNIDAD 2: Estructuras de control de flujo y variables	
Tema 7: Reutilización de instrucciones para control iterativo	
Objetivos:	Que el estudiante pueda establecer un conjunto de instrucciones que puedan ser utilizadas a modo de órdenes iterativas con un elemento de control
Descripción de actividades:	Explicación y uso videojuego Blockly (Laberinto) niveles 1 al 5: Se brinda una teoría del funcionamiento del juego, así como de los constructos de desarrollo (órdenes, ubicación de secuencias, repeticiones). Uso del videojuego nivel 6 en adelante por parte de los niños.

Recursos:	Físicos: Sala de informática y salón de clases, proyector, computador para docente, computadores para estudiantes Software: Blockly (Laberinto)
Tiempo requerido:	1 hora
Tema 8: Instrucciones de control condicional para la consecución de objetivos, uso de un dato en instrucciones	
Objetivos:	Que el estudiante pueda establecer un conjunto de instrucciones que puedan definirse dentro de controles condicionales para la consecución de un objetivo específico.
Descripción de actividades:	Explicación y uso videojuego Blockly (laberinto) niveles 6 al 10: Se brinda una teoría del funcionamiento de las órdenes de control condicional.
Recursos:	Físicos: Sala de informática y salón de clases, proyector, computador para docente, computadores para estudiantes Software: Blockly (Laberinto)
Tiempo requerido:	1 hora
Tema 9: Definición de un dato, tipos de datos (numéricos, textuales, combinados, lógicos)	
Objetivos:	Que el estudiante comprenda el concepto de dato, identifique datos en la vida diaria, y pueda identificar los tipos de datos más comunes (textos, números, combinados, lógicos)
Descripción de actividades:	Charla sobre datos en la vida diaria, uso de los números, uso de las palabras, uso del sí y el no. Además se hablará sobre las características de objetos de la vida diaria y sus valores. Juego ¿Adivina qué es?: El docente presenta una matriz de 12 x 12 elementos con algunas características comunes y selecciona uno de los elementos para que los estudiantes adivinen cuál se seleccionó. Los estudiantes deberán hacer grupos de 2 personas y establecerán preguntas sobre características que ellos mismos deberán identificar.
Recursos:	Físicos: Salón de clases, tablero impreso de 144 objetos, 144 cartas, hojas de papel y lápices, tablero de clase.
Tiempo requerido:	1 hora
Tema 10: Uso de instrucciones con datos	
Objetivos:	Que el estudiante comprenda identificar datos para ser aplicados en conjunto con las instrucciones adecuadas para la consecución de un objetivo determinado
Descripción de actividades:	Explicación y uso videojuego Blockly (tortuga), así como de los constructos de desarrollo (secuencias, acciones, controles de flujo, variables). Se indicará la

actividades:	mecánica de trabajo y se propondrá un juego con el fin de generar un dibujo con ciertas especificaciones.
Recursos:	Físicos: Sala de informática y salón de clases, proyector, computador para docente, computadores para estudiantes Software: Blockly (Tortuga)
Tiempo requerido:	1 hora
Tema 11: Comprensión de eventos en situaciones reales y extracción de datos	
Objetivos:	Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo mediante instrucciones y datos definidos para el cumplimiento de un objetivo
Descripción de actividades:	Charla sobre actividades de la vida diaria Ejercicios con versiones escritas de algoritmos, identificando datos y acciones con los datos
Recursos:	Físicos: Salón de clases
Tiempo requerido:	1 hora

Fuente: Autor del proyecto

Como se observa, se lograron integrar algunas actividades lúdicas encontradas en las metodologías y experiencias revisadas, con el fin de aportar elementos tanto teóricos como motivacionales y no generar distracciones en la explicación de algunos conceptos con el uso excesivo de los computadores.

3.4.6 Forma de evaluación

Dentro de los elementos de evaluación de estos temas se identificaron los objetivos que deben adquirir los estudiantes en cada tema. Dado que actualmente en la asignatura de tecnología no se ven los temas referentes a algoritmos y la asignatura no está diseñada para aplicarse bajo la metodología de enseñanza por competencias, se planteó inicialmente trabajar bajo la modalidad de enseñanza tradicional por objetivos.

Con el fin de generar elementos de medición se estableció un conjunto de actividades complementarias en ambas partes del curso. Una de ellas fue generar quices escritos usando como dibujo la misma representación del juego LightBot, en el que se solicitaba a cada estudiante que dibujara el conjunto de instrucciones en el papel para que el robot pase por varios puntos y los pueda iluminar. Algo similar se planteó para la segunda parte con el uso de Blockly pero para la generación de dibujos por escrito usando las mismas instrucciones que debían darse a la tortuga. Para complementar y observar la aplicación de los temas en casos más reales, se generó un quiz en el que debían organizar una serie de instrucciones en forma secuencial para la realización de una actividad de la vida diaria. Otras técnicas de evaluación se basaron en la observación directa y en la revisión de resultados mientras se desarrollaban las actividades que requerían el uso del computador. Estas técnicas de evaluación se plantearon con el fin de que la docente pudiera obtener una forma de medición de logro de objetivos por parte de los estudiantes. Sin embargo, no se contempló para este proyecto tomar los resultados de estas evaluaciones para el diseño del instrumento de medición para determinar la capacidad de resolución de problemas matemáticos.

Luego de esta generación de elementos, se procedió a acoplar esta información y posteriormente a modificar su estructura para acoplarse al plan de área del Colegio San Juan de Girón. Este plan curricular se incorporó a la propuesta general del proyecto presentado ante la Directora de la institución educativa, quien dio aval para la aplicación de éste en el grado 5°. También se presentó un bosquejo de carta de consentimiento para los padres de familia de los estudiantes del grado 5°, explicando a manera general el objetivo del proyecto, solicitando permiso para reportar las actividades que se realizaron en la institución tanto en forma audiovisual como valorativa, y adicionando elementos legales como la renuncia por parte de los padres a cualquier tipo de contraprestación. Con la generación del plan curricular a proponer para el área de tecnología, logrando el visto bueno por parte de la dirección de la institución educativa, y obtenido el aval por parte de los padres de familia de los estudiantes, se dio por cumplido el segundo objetivo específico.

El plan curricular desarrollado y aplicado se basó inicialmente en un enfoque tradicional basado en objetivos, tal y como se había venido generando el tratamiento de la asignatura de tecnología. No obstante, luego de haber indagado

y realizado la propuesta curricular a partir de los objetivos trazados, y considerando que, de acuerdo a las políticas de mejoramiento de la calidad de la educación del Ministerio de Educación Nacional es muy importante trabajar desde la perspectiva de competencias, se encontró pertinente convertir esta propuesta a un enfoque basado en competencias. Por ello se complementó esta investigación realizando una revisión de las competencias que debían adquirir los estudiantes de acuerdo a los estándares planteados en el área de tecnología e informática. Los resultados de esta investigación adicional se presentan en el capítulo siguiente. Cabe aclarar en este punto que este aporte de generación de un nuevo plan enfocado por competencias no se pensó crear al momento de proponer el plan curricular presentado inicialmente.

3.5 APLICACIÓN DEL CURSO Y DE LAS PRUEBAS

Luego de obtener el aval por parte de la dirección de la institución, se realizó una socialización del proyecto con los coordinadores de la sede y al docente Carlos Barragán, del área de matemáticas, explicándole los objetivos del proyecto para que sirviera de apoyo para la aplicación de las pruebas de medición.

Con la ayuda de la docente Carmen Alicia Albarracín del área de tecnología se estudió la forma de aplicación del plan en Colegio San Juan de Girón, determinando los grupos de estudiantes disponibles. Se contó con los grupos 5-1 y 5-2 de la institución a quienes se podría aplicar el experimento, por lo que se procedió a realizar selección de manera aleatoria para determinar cuál de los grupos debía ser el experimental y cuál el de control. Se determinó que el grupo 5-1 conformado por 42 estudiantes sería el experimental, y el grupo 5-2 sería el grupo control, también de 42 estudiantes. Se estableció el diseño de la aplicación del experimento asignando los identificadores de los grupos e indicando el tratamiento en el grupo experimental, de la siguiente forma:

	Pretest	Intervención	Prueba final
G1 (Experimental)	0	X	0
G2 (Control)	0		0

Previa a la aplicación del curso, se buscó corroborar la homogeneidad de ambos grupos, tanto el de control como el experimental, por lo cual se diseñó un instrumento de medición para generar una prueba de base con el fin de determinar el nivel inicial de competencias de los estudiantes. Este mismo diseño se utilizó para la determinación de la adquisición de competencias a ser aplicado al final del curso, aunque modificando la complejidad de las preguntas de la prueba inicial y la final.

3.5.1 Diseño de los instrumentos de medición

Basados en los experimentos de Agina (2012), se planteó un diseño de prueba que contó con una serie de 5 preguntas de tipo matemático para resolver problemas referentes a una temática recientemente vista, y 3 preguntas adicionales sobre las opiniones que tiene sobre la prueba. Dada la facilidad de las preguntas, no hubo necesidad de generar un gráfico explicativo para la comprensión del contexto del problema. Se determinó en conjunto con el docente del área de matemáticas los temas a tratar en la evaluación, encontrando que al principio de año habían desarrollado un repaso de las operaciones básicas: suma, resta, multiplicación y división. La prueba midió las siguientes variables:

- Cantidad de ejercicios ejecutados
- Interpretación de la operación
- Procedimiento de resolución del ejercicio
- Número de respuestas correctas
- Tiempo que le tomó desarrollar los ejercicios
- Nivel de satisfacción que causó la aplicación de las pruebas

Esta última variable pretendió cubrir una variable adicional la cual se vio en el experimento de Giannakos y Jaccheri (2013), en el cual se realizaban preguntas para determinar el grado de satisfacción, utilidad y facilidad de las pruebas, por lo que se diseñó una serie de 3 preguntas adicionales para determinar estas variables que ayudaron a determinar si ya se había adquirido intrínsecamente la habilidad para comprender los problemas que se le presentaron en la prueba, además de servir como retroalimentación para mejorar las preguntas si se encontraba que no genera un nivel de satisfacción en forma general.

Las preguntas matemáticas generadas para la primera prueba se diseñaron para que fueran fácilmente identificables los datos iniciales y el tipo de operación matemática a realizar. Se dejó espacio para que los mismos estudiantes interpretaran y describieran los resultados obtenidos en la operación. Las preguntas fueron las siguientes:

- *Un niño llamado Juanito le pide dinero a su mamá para comprar dulces. Juanito quiere comprar unos chicles que cuestan \$ 400 cada uno. Su mamá le da \$ 5.000 para que compre los chicles que quiera. ¿Cuántos chicles alcanza a comprar Juanito y cuánto dinero le sobra?*
- *Un taxista recorre una distancia de 2675 kilómetros en la mañana y 1967 kilómetros en la tarde. ¿Cuántos kilómetros recorrió el taxista en el día?*
- *En una batalla, Goku tiene un poder de pelea de 357250, y su enemigo Vegeta tiene un poder de pelea de 348524. ¿Cuánta diferencia hay entre el poder de pelea de Goku y el poder de pelea de Vegeta? ¿Será que Vegeta le puede ganar a Goku?*
- *Tenemos que ayudarlo al profesor Carlos a arreglar el salón, porque las sillas están desordenadas. Hay que ordenarle en el salón 8 filas de 12 sillas cada una. ¿Cuántas sillas habrá que ordenar en el salón?*
- *Estamos en el año 2014. Si mi papá cumplió años hoy, y nació en el año 1951, ¿cuántos años cumplió mi papá?*

Se estableció la forma de medición basado en las variables expuestas por Agina, identificando exactamente lo que se evaluará en las pruebas para cada variable y relacionando la variable a medir con las habilidades tanto matemáticas como algorítmicas que debía adquirir el estudiante.

Tabla 8. Variables identificadas para medir en las pruebas pretest y final

Variab	a	Identificador	Interpretación para evaluar	Relación habilidad algorítmica/matemática
Cantidad de ejercicios ejecutados	de	Preguntas contestadas PC	Evaluar si el estudiante ha contestado la pregunta o no, es decir, si entendía que debe dar algún tipo de respuesta en cada pregunta	Algorítmica: Se debe llegar a un objetivo Matemática: Se debe solucionar una incógnita
Interpretación de la operación		Operaciones interpretadas	Evaluar si el estudiante ha interpretado la operación adecuada o no, es decir, si	Algorítmica: Se debe establecer los datos, las

	OI	determinaba cuál era la operación o la secuencia de operaciones matemáticas (suma, resta, multiplicación, división) que debe realizar con los datos que aparecen en la pregunta, y si los datos de entrada son ubicados correctamente para la operación	instrucciones apropiadas y secuenciadas correctamente. Matemática: Se debe determinar los datos, las operaciones y el plan de cálculo a seguir para hallar la incógnita
Procedimiento de resolución del ejercicio	Ejecuciones correctas EC	Evaluar si ha ejecutado adecuadamente la operación o no, es decir, si contaba con la capacidad de desarrollar correctamente las operaciones matemáticas con los datos que aparecen en la pregunta	Algorítmica: Se debe ejecutar la secuencia de instrucciones en forma precisa Matemática: Se debe ejecutar el plan de cálculo de manera correcta
Número de respuestas correctas	Respuestas acertadas RA	Evaluar si ha descrito la respuesta acertadamente o no, es decir, que podía comprender los resultados obtenidos para responder acertadamente la pregunta	Algorítmica: Se debe verificar que se llega al objetivo esperado Matemática: Se debe verificar que la solución obtenida es lógica respecto a lo esperado
Tiempo que le tomó desarrollar los ejercicios	Tiempo de desarrollo TD	Determinar en cuánto tiempo presentó la prueba, en el cual se estableció un límite máximo de 20 minutos en el pretest	-

Fuente. Autor del proyecto

Solo 2 de las 5 preguntas se plantearon como preguntas dobles, es decir, que al desarrollar la operación matemática se podría obtener 2 respuestas en la interpretación del resultado. Por esa razón, las variables de contestación de preguntas y acierto en las respuestas se contaron sobre 7, mientras que la interpretación de las operaciones adecuadas y su ejecución se contaron sobre 5. Finalmente, se diseñó la variable de calificación final que sirvió como indicador de puntaje total de la prueba, estableciéndose de la siguiente forma:

- La calificación final tomó un rango entre 0 y 10 puntos.
- Se estimó para cada variable un peso particular dependiendo de los parámetros comparativos establecidos entre las competencias matemáticas a medir y las competencias que se pretendieron adquirir a través de la enseñanza de la computación.
- La fórmula matemática diseñada para determinar la calificación final quedó de la siguiente forma:

$$Cf = 10 * \frac{\sum (w_i * \frac{v_i}{vmax_i})}{\sum w_i}$$

donde v_i representa cada una de las variables usadas (cantidad de preguntas contestadas, cantidad de operaciones interpretadas, cantidad de ejecuciones de operaciones correctas, cantidad de respuestas acertadas), v_{max_i} representa cada una de las cantidades máximas posibles por variable y w_i representa cada uno de los pesos propuestos para cada variable. El valor 10 representa el valor máximo posible de la calificación.

- Los pesos se establecieron con un valor entre 1 y 3, siendo 3 la variable que adquirió mayor importancia. Así, la variable de contestación de preguntas obtuvo un peso de 2 dado que se pretendió determinar con ésta variable si el estudiante comprendía que debía cumplir un objetivo (parte de la definición del algoritmo), la variable de interpretación de operaciones obtuvo un peso de 3, puesto que es el paso más importante en el diseño de un algoritmo (establecer los datos y las secuencias de operación), la variable de ejecución de operaciones tomó un valor de 2 puesto que esta variable es más operativa que algorítmica, y la variable de acierto en la respuesta obtuvo un valor de 1, dado que consistía solo en una interpretación del resultado obtenido.

Con esta información, la fórmula final para calificar el pretest quedó de la siguiente forma:

$$Cf = 10 * \frac{2 * \frac{PC}{7} + 3 * \frac{OI}{5} + 2 * \frac{EC}{5} + 1 * \frac{RA}{7}}{8}$$

Por otra parte, las 3 preguntas para determinar las variables de nivel de satisfacción se diseñaron para que se generara una selección de 1 de 3 posibilidades de respuesta para cada pregunta, de modo que fuera rápidamente contestada y no sobre-esforzar al estudiante a que describiera la actividad.

Las preguntas fueron las siguientes:

Ahora encierra en un círculo tu opinión:

- *¿Te parecieron fáciles las preguntas?*
 - *Si, muy fáciles*
 - *Más o menos*
 - *Estaban muy difíciles*

- *¿Sentiste nervios al contestar las preguntas?*
 - *Si, muchos nervios*
 - *Solo un poquito*
 - *No, nada nada*

- *¿Te gustó esta actividad?*
 - *Si, muy divertido*
 - *Más o menos*
 - *No me gustó. Fue muy aburrido.*

3.5.2 Aplicación de prueba inicial y recopilación de datos

El docente del área de matemáticas estuvo a cargo de la aplicación de las pruebas durante sus horas de clase, tanto para el grupo experimental como para el grupo control. El docente indicó a los participantes sobre la metodología de la actividad a desarrollar. Al iniciar la prueba, el docente tomó la hora exacta de inicio de esta, los niños se sometieron a la primera prueba durante un tiempo de 20 minutos, tiempo durante el cual los estudiantes desarrollaron los 5 ejercicios y contestaron las preguntas de satisfacción.

Al docente se le sugirió que cuando fuera requerido por algún estudiante para realizarle alguna pregunta sobre algún punto en particular de la prueba, registrara este requerimiento en la hoja, sin embargo ningún estudiante solicitó ayuda. Al terminar, el estudiante entregó sus resultados al docente. En conjunto con el docente, se realizó una revisión del procedimiento y los resultados. Los datos de las cantidades de procedimientos completos, operaciones interpretadas, procedimientos correctos, respuestas correctas, tiempos usados, y la información de las preguntas de satisfacción fue organizada y tabulada para su análisis.

3.5.3 Circunstancias que rodearon la aplicación del curso

Luego de la aplicación de la prueba se procedió a organizar los recursos con los que contaba la institución para la impartición de las clases al grupo experimental. El colegio contaba con 3 valijas procedentes del programa Computadores para Educar, suministradas por el Ministerio de las TIC. Cada una contaba con un conjunto de 10 computadores portátiles, mouse, cargadores, extensiones de tipo

regleta y extensiones de alto vatiaje. Se realizó una relación de recursos revisados, la cual se presentó a la coordinación de la institución. Luego se procedió a instalar los programas Lightbot y Blockly, asegurándose que funcionaran adecuadamente en los equipos y que no se tuviera alguna restricción.

Durante 10 jornadas de una hora cada una, se impartió una temática en el área de tecnología referente a la programación de computadores y se presentaron los temas básicos referentes a los algoritmos, determinados en el plan de área presentado. Los estudiantes tomaron esta temática dentro de la misma jornada de la asignatura Tecnología e Informática, pero no como un curso en paralelo ni como jornada adicional.

Se trabajó utilizando los recursos computacionales descritos anteriormente, complementando además con instrucción teórica para la comprensión de generación de algoritmos y otros aspectos, los cuales formaron parte del plan de área presentado. Dada la limitación en la cantidad de equipos, se trabajó con dos estudiantes por equipo. Adicionalmente, con el fin de evidenciar que cada estudiante generó la independencia de su habilidad adquirida con respecto a la de sus compañeros, se realizaron las actividades anexas como quices individuales en clase por parte de la docente de tecnología, con el fin de evaluar su rendimiento y su trabajo disciplinado durante la clase. En cuanto al área de matemáticas, los temas presentados en dicha asignatura no se relacionaron con los temas aplicados en el curso de tecnología durante la ejecución del proyecto, dado que para ese instante los estudiantes debían ya entrar a ver conceptos de geometría y no continuaron desarrollando ejercicios de repaso de operaciones básicas.

Con la implementación del curso piloto planteado para el área de tecnología en el grupo experimental, se dio por cumplido el tercer objetivo específico propuesto en este proyecto.

3.5.4 Aplicación de prueba final y recopilación de datos

En la décima semana de aplicación del curso se procedió a realizar la aplicación de la prueba final. Nuevamente el docente del área de matemáticas estuvo a cargo de la aplicación de las pruebas durante sus horas de clase, tanto para el grupo experimental como para el grupo control. En este caso, las preguntas

tuvieron un mayor nivel de complejidad, pero manteniendo la utilización de las 4 operaciones básicas para su desarrollo:

- *Un profesor llamado Eduardo ha ganado \$62800 el día lunes, \$71100 el día martes, y \$52500 el día miércoles. ¿Cuánto dinero ha ganado Eduardo en estos tres días?. Si el profesor Eduardo se quiere comprar un celular que cuesta \$250000, ¿cuánto dinero le haría falta para comprarlo?*
- *Un camión llevaba 7 cajas que pesaban cada una de a 15 kilos, y 8 cajas que pesaban 12 kilos cada una. ¿Cuánto pesaban todas las cajas que llevaba el camión?*
- *En una competencia de vuelo, Superman voló 152421 kilómetros en la primera vuelta y 146257 kilómetros en la segunda vuelta. IronMan voló 165483 kilómetros en la primera vuelta y 136654 en la segunda vuelta. ¿Quién ganó la competencia y cuántos kilómetros le llevó de ventaja al perdedor?*
- *Las gallinas de una finca pusieron 675 huevos en una semana. Si cada una de las gallinas puso 5 huevos, ¿cuántas gallinas hay en la finca?. Si se venden todos los huevos a \$200 cada uno, ¿cuánto podrían costar todos los huevos que pusieron las gallinas?*
- *En el ejército hay 4262 soldados. Han llegado 1318 soldados más y se han ido 825. ¿Cuántos hay ahora?*

Se utilizó el mismo sistema de calificación del pretest, sin embargo, el planteamiento para el desarrollo de las operaciones fue diferente: todas las preguntas debían utilizar una combinación de identificación de dos tipos de operaciones matemáticas, además, 2 de las 5 preguntas se plantearon como pregunta sencilla, es decir, que al desarrollar las dos operaciones matemáticas se debía obtener solo 1 respuesta en la interpretación del resultado, mientras que las otras 3 preguntas se concibieron como preguntas dobles, en las que por cada tipo de operación ejecutada se obtenía 1 respuesta. Por esa razón, las variables de contestación de preguntas y acierto en las respuestas se contaron sobre 8, mientras que la interpretación de las operaciones adecuadas y su ejecución se contaron sobre 10. Se planteó así con el fin de determinar si encontraban la forma secuencialmente correcta de desarrollar estas operaciones y de incrementar el

nivel de dificultad con respecto a la prueba inicial, a pesar que se planteó estudiar la misma habilidad de uso de operaciones básicas. La variable de calificación final se trabajó de la misma forma que en el pretest con el fin de mantener un parámetro similar para comparar:

$$Cf = 10 * \frac{\sum \left(w_i * \frac{v_i}{vmax_i} \right)}{\sum w_i}$$

Los pesos se mantuvieron exactamente igual que en la primera prueba, los únicos valores que se modificaron fueron las cantidades máximas de cada variable. Así, la fórmula para calificar la prueba final quedó de la siguiente forma:

$$Cf = 10 * \frac{2 * \frac{PC}{8} + 3 * \frac{OI}{10} + 2 * \frac{EC}{10} + 1 * \frac{RA}{8}}{8}$$

Al igual que la primera prueba, el docente tomó la hora exacta de inicio de esta, sin embargo para esta prueba intermedia los niños se sometieron a la primera prueba durante un tiempo de 30 minutos, tiempo durante el cual los estudiantes desarrollaron los 5 ejercicios y contestaron las preguntas de satisfacción. Al terminar, el estudiante entregó sus resultados al docente. En conjunto con el docente, se realizó una revisión del procedimiento y los resultados. Los datos de procedimientos completos, operaciones interpretadas, procedimientos correctos, respuestas correctas, tiempos usados, y la información de las preguntas de satisfacción fueron organizados y tabulados para su análisis estadístico.

4. RESULTADOS Y ANÁLISIS DE LAS PRUEBAS DE DESEMPEÑO

4.1 RESULTADOS DE LAS CALIFICACIONES OBTENIDAS EN LAS PRUEBAS

Las pruebas aplicadas en el pretest y en la prueba final fueron calificadas y tabuladas. Inicialmente se planteó realizar la comparación de las calificaciones finales de ambos grupos en el pretest contra los datos arrojados en la prueba final.

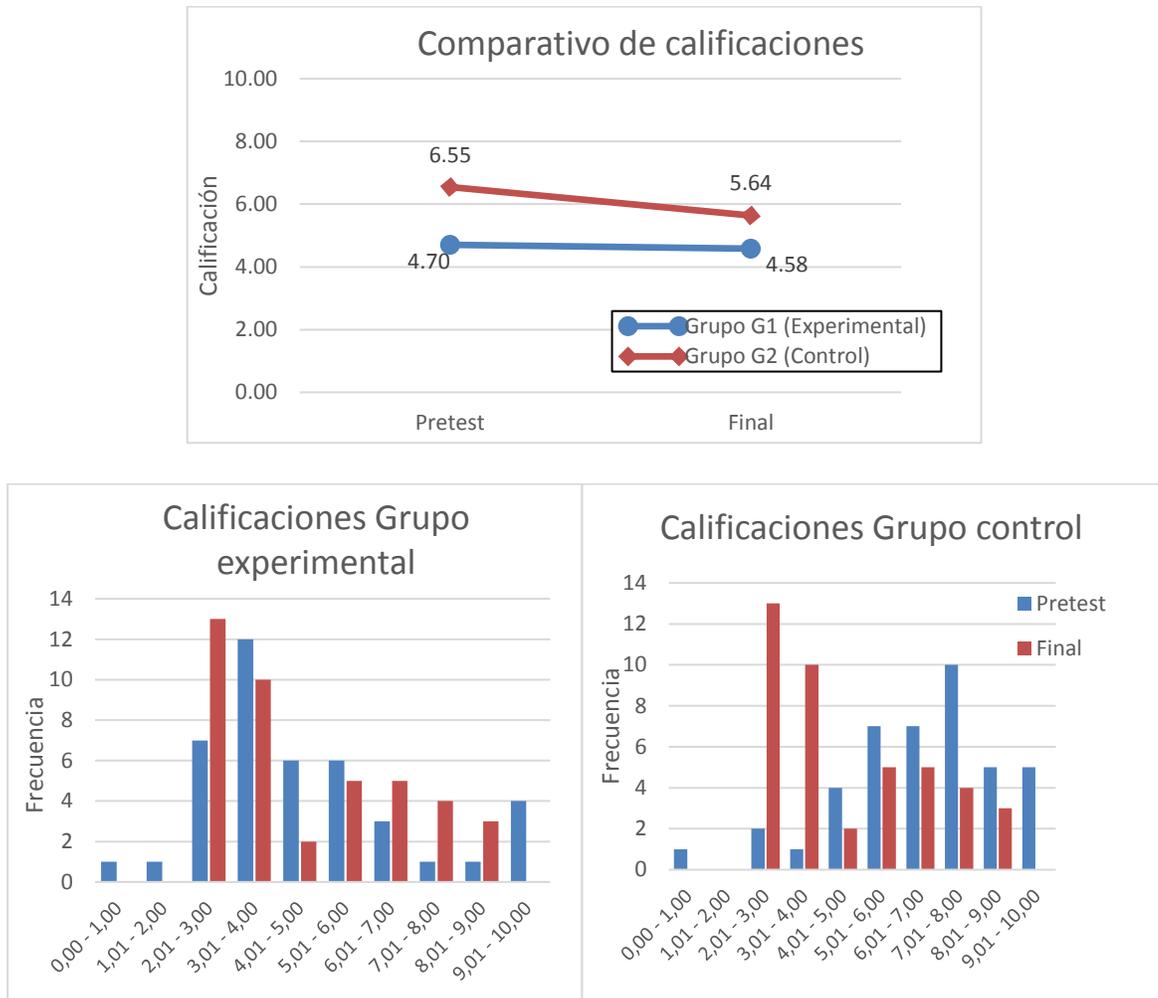
Tabla 9. Tabulación de los valores de las calificaciones finales en las pruebas de los estudiantes de los dos grupos

Calificación	Grupo G1 (Experimental)		Grupo G2 (Control)	
	Pretest	Final	Pretest	Final
0,00 - 1,00	1	0	1	1
1,01 - 2,00	1	0	0	2
2,01 - 3,00	7	13	2	6
3,01 - 4,00	12	10	1	2
4,01 - 5,00	6	2	4	4
5,01 - 6,00	6	5	7	5
6,01 - 7,00	3	5	7	9
7,01 - 8,00	1	4	10	4
8,01 - 9,00	1	3	5	7
9,01 - 10,00	4	0	5	2
n:	42	42	42	42
x:	4,70	4,58	6,55	5,64

Fuente: Autor del proyecto

Se pudo observar que el grupo de control obtuvo mejores calificaciones que el grupo experimental, tanto en el pretest como en la prueba final, sin embargo en ambos grupos disminuyó el valor promedio de las calificaciones obtenidas en la prueba final comparado con los obtenidos en el pretest. Se evidenció además que los estudiantes del grupo experimental mantuvieron un valor similar en las medias de los resultados ($x_{pre}=4.70$, $x_{fin}=4.58$), pero el grupo de control no logró mantener un nivel parejo en las dos pruebas ($x_{pre}=6.55$, $x_{fin}=5.64$). Se generó un gráfico de líneas y un histograma para observar el comportamiento general de las calificaciones obtenidas por los grupos.

Figura 3. Gráfico comparativo e histogramas de las calificaciones finales en las pruebas de los estudiantes de los dos grupos



Fuente: Autor del proyecto

4.2 RESULTADOS POR VARIABLE

Con el fin de estudiar a fondo lo ocurrido se realizó además la tabulación de los datos de las 5 primeras variables que se midieron en la prueba, generando resultados por porcentaje de variable cumplida. Los datos se presentan a continuación:

Tabla 10. Tabulación de los valores de las variables medidas en las pruebas de los estudiantes de los dos grupos

Resultados Pretest Grupo G1:

Preguntas contestadas		Operac. interpretadas		Ejecuciones correctas		Respuestas acertadas		Duración en la prueba	
Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	t (min)	Frec.
Ninguna	1 (2,38%)	Ninguna	3 (7,14%)	Ninguna	7 (16,66%)	Ninguna	4 (9,52%)	13	0 (0%)
1	0 (0%)	1	13 (30,95%)	1	19 (45,23%)	1	14 (33,33%)	14	0 (0%)
2	2 (4,76%)	2	10 (23,8%)	2	9 (21,42%)	2	11 (26,19%)	15	0 (0%)
3	2 (4,76%)	3	10 (23,8%)	3	1 (2,38%)	3	4 (9,52%)	16	0 (0%)
4	5 (11,9%)	4	1 (2,38%)	4	1 (2,38%)	4	2 (4,76%)	17	1 (2,38%)
5	13 (30,95%)	5	5 (11,9%)	5	5 (11,9%)	5	3 (7,14%)	18	2 (4,76%)
6	14 (33,33%)					6	0 (0%)	19	0 (0%)
7	5 (11,9%)					7	4 (9,52%)	20	39 (92,85%)
n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)
\bar{x}	5,095238095	\bar{x}	2,19047619	\bar{x}	1,642857143	\bar{x}	2,357142857	\bar{x}	19,83333333
%PC	5.09/7 (72.7%)	%OI	2.19/5 (43.8%)	%EC	1.64/5 (32.8%)	%RA	2.35/7 (33.5%)	-	-
s	1,47845542	s	1,418313929	s	1,495055731	s	1,997821113	s	0,62143291

Resultados Pretest Grupo G2:

Preguntas contestadas		Operac. interpretadas		Ejecuciones correctas		Respuestas acertadas		Duración en la prueba	
Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	t (min)	Frec.
Ninguna	0 (0%)	Ninguna	3 (7,14%)	Ninguna	4 (9,52%)	Ninguna	4 (9,52%)	13	6 (14,28%)
1	0 (0%)	1	3 (7,14%)	1	6 (14,28%)	1	4 (9,52%)	14	1 (2,38%)
2	1 (2,38%)	2	5 (11,9%)	2	10 (23,8%)	2	3 (7,14%)	15	1 (2,38%)
3	1 (2,38%)	3	12 (28,57%)	3	13 (30,95%)	3	9 (21,42%)	16	0 (0%)
4	2 (4,76%)	4	13 (30,95%)	4	5 (11,9%)	4	9 (21,42%)	17	0 (0%)
5	7 (16,66%)	5	6 (14,28%)	5	4 (9,52%)	5	8 (19,04%)	18	0 (0%)
6	16 (38,09%)					6	2 (4,76%)	19	0 (0%)
7	15 (35,71%)					7	3 (7,14%)	20	34 (80,95%)
n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)
\bar{x}	5,928571429	\bar{x}	3,119047619	\bar{x}	2,5	\bar{x}	3,476190476	\bar{x}	18,73809524
%PC	5.92/7 (84.5%)	%OI	3.11/5 (62.2%)	%EC	2.5/5 (50.0%)	%RA	3.47/7 (49.5%)	-	-
s	1,155957151	s	1,400389858	s	1,401218982	s	1,92845313	s	2,650904345

Resultados Prueba Final Grupo G1:

Preguntas contestadas		Operac. interpretadas		Ejecuciones correctas		Respuestas acertadas		Duración en la prueba	
Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	t (min)	Frec.
Ninguna	0 (0%)	Ninguna	1 (2,38%)	Ninguna	6 (14,28%)	Ninguna	8 (19,04%)	23	0 (0%)
1	0 (0%)	1	4 (9,52%)	1	10 (23,8%)	1	16 (38,09%)	24	0 (0%)
2	0 (0%)	2	7 (16,66%)	2	8 (19,04%)	2	4 (9,52%)	25	0 (0%)
3	0 (0%)	3	8 (19,04%)	3	4 (9,52%)	3	7 (16,66%)	26	0 (0%)
4	2 (4,76%)	4	3 (7,14%)	4	5 (11,9%)	4	4 (9,52%)	27	0 (0%)
5	12 (28,57%)	5	3 (7,14%)	5	2 (4,76%)	5	1 (2,38%)	28	1 (2,38%)
6	10 (23,8%)	6	6 (14,28%)	6	3 (7,14%)	6	2 (4,76%)	29	0 (0%)
7	10 (23,8%)	7	3 (7,14%)	7	2 (4,76%)	7	0 (0%)	30	41 (97,61%)
8	8 (19,04%)	8	4 (9,52%)	8	2 (4,76%)	8	0 (0%)		
		9	3 (7,14%)	9	0 (0%)				
		10	0 (0%)	10	0 (0%)				
n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)
\bar{x}	6,238095238	\bar{x}	4,404761905	\bar{x}	2,761904762	\bar{x}	1,857142857	\bar{x}	29,95238095
%PC	6.23/8 (77.8%)	%OI	4.40/10 (44.0%)	%EC	2.76/10 (27.6%)	%RA	1.85/8 (26.4%)	-	-
s	1,205870272	s	2,585697281	s	2,324939273	s	1,646451151	s	0,3086067

Resultados Prueba Final Grupo G2:

Preguntas contestadas		Operac. interpretadas		Ejecuciones correctas		Respuestas acertadas		Duración en la prueba	
Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	Cant.	Frec.	t (min)	Frec.
Ninguna	0 (0%)	Ninguna	1 (2,38%)	Ninguna	3 (7,14%)	Ninguna	5 (11,9%)	23	0 (0%)
1	1 (2,38%)	1	5 (11,9%)	1	5 (11,9%)	1	5 (11,9%)	24	0 (0%)
2	1 (2,38%)	2	2 (4,76%)	2	3 (7,14%)	2	11 (26,19%)	25	2 (4,76%)
3	1 (2,38%)	3	2 (4,76%)	3	5 (11,9%)	3	6 (14,28%)	26	0 (0%)
4	4 (9,52%)	4	3 (7,14%)	4	4 (9,52%)	4	6 (14,28%)	27	0 (0%)
5	5 (11,9%)	5	6 (14,28%)	5	8 (19,04%)	5	6 (14,28%)	28	1 (2,38%)
6	8 (19,04%)	6	4 (9,52%)	6	2 (4,76%)	6	2 (4,76%)	29	1 (2,38%)
7	6 (14,28%)	7	3 (7,14%)	7	7 (16,66%)	7	0 (0%)	30	38 (90,47%)
8	16 (38,09%)	8	7 (16,66%)	8	4 (9,52%)	8	1 (2,38%)		
		9	4 (9,52%)	9	0 (0%)				
		10	5 (11,9%)	10	1 (2,38%)				
n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)	n	42 (100%)
\bar{x}	6,30952381	\bar{x}	5,80952381	\bar{x}	4,404761905	\bar{x}	2,833333333	\bar{x}	29,69047619
%PC	6.30/8 (78.7%)	%OI	5.80/10 (58.0%)	%EC	4.40/10 (44.0%)	%RA	2.83/8 (35.3%)	-	-
s	1,827808167	s	3,022178953	s	2,623156949	s	1,898865729	s	1,115043371

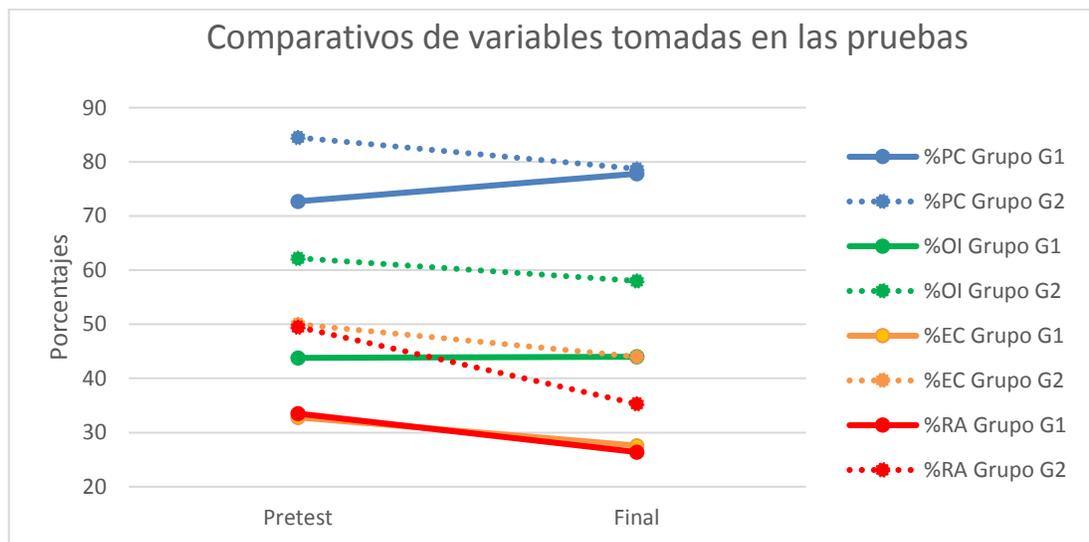
A partir de los resultados obtenidos del porcentaje de variables cumplidos, se realizó el siguiente resumen tabulado y se generó un gráfico comparativo por variables.

Tabla 11. Resumen de resultados de los porcentajes de variables en las pruebas de los estudiantes de los dos grupos

Variable (%)	Grupo G1 (Experimental)		Grupo G2 (Control)	
	Pretest	Final	Pretest	Final
%PC	72.7	77.8	84.5	78.7
%OI	43.8	44.0	62.2	58.0
%EC	32.8	27.6	50.0	44.0
%RA	33.5	26.4	49.5	35.3

Fuente: Autor del proyecto

Figura 4. Gráfico comparativo de los porcentajes de variables en las pruebas de los estudiantes de los dos grupos



Fuente: Autor del proyecto

Se pudo observar que el grupo de control tuvo en todas las variables un porcentaje completado más alto que el grupo experimental, en el pretest y en la prueba final. Pero se pudo notar un comportamiento diferente al comparar las variables de preguntas contestadas (PC) y operaciones interpretadas (OI), dado

que mientras que en el grupo experimental G1 los valores aumentaron, en el grupo de control disminuyeron.

4.3 ANÁLISIS ESTADÍSTICO

4.3.1 Fiabilidad de las variables

Antes de generar la comparación estadística de los datos se procedió a realizar un análisis de fiabilidad de las variables medidas en el pretest (PC, OI, EC, RA, TD) y de la variable de calificación (Cf), determinando promedios de correlaciones entre ellas. Se generó la revisión usando la herramienta IBM SPSS v.20, observando que 5 de las 6 variables medidas tienen un alto nivel de correlación entre ellas, sin embargo la variable de tiempo de desarrollo (TD) parecía no presentar una correlación adecuada con las otras variables, dado que muchos de los datos correspondían a los tiempos límite asignados, por lo que se planteó una variable nueva que determinó si los estudiantes completaban la actividad en menos tiempo que el límite establecido o no (TML). Al integrar este nuevo conjunto de datos, se observó que presentaba una correlación adecuada con respecto a las otras variables.

Tabla 12. Resultado de correlación entre las variables y la calificación final mediante alfa de Cronbach

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,911	,915	6

Fuente: Autor del proyecto

4.3.2 Análisis de resultado de la calificación final

El grupo de control obtuvo mejores calificaciones que el grupo experimental en ambas pruebas, pero en ambos grupos disminuyó la media de las calificaciones obtenidas en la prueba final respecto a las obtenidas en el pretest, más en el grupo de control que en el grupo experimental. Con el fin de determinar si todas

estas variaciones fueron significativas, estos hechos se procedieron a revisar estadísticamente, realizando inicialmente los comparativos en forma transversal entre los resultados del grupo experimental y el grupo de control tanto en el pretest como en la prueba final y tomando como base las siguientes hipótesis nulas y alternativas:

Ho ₁ : NO se apreció diferencia estadísticamente significativa entre las calificaciones entre los estudiantes del grupo G1 y los estudiantes del grupo G2 antes de la intervención.	Ho ₁ : $\mu_1(\text{preG1}) = \mu_2(\text{preG2})$
Ha ₁ : Se apreció diferencia estadísticamente significativa entre las calificaciones entre los estudiantes del grupo G1 y los estudiantes del grupo G2 antes de la intervención.	Ha ₁ : $\mu_1(\text{preG1}) \neq \mu_2(\text{preG2})$

Tabla 13. Análisis estadístico mediante prueba T resultados pretest (Comparación G1 - G2)

Variable	Supuestos	Prueba Levene calidad varianzas		Prueba t para la igualdad de medias						
		F	Sig.	T	gl	Sig. (bilat)	Dif. medias	Dif. error std.	95% intervalo confianza dif.	
									Inf.	Sup.
Calificación final	Varianzas iguales	,394	,532	-3,716	82	,000	-1,84269	,49587	-2,82912	-,85625
	No varianzas iguales			-3,716	80,824	,000	-1,84269	,49587	-2,82934	-,85603

Fuente: Autor del proyecto

Para el análisis transversal con los datos de las calificaciones obtenidas en el pretest se realizó una prueba t de Student para muestras independientes, determinando que se rechaza la hipótesis nula planteada Ho₁ ($p < 0.05$), aceptando la hipótesis alternativa Ha₁. Es decir, se apreciaron evidencias que revelaron una diferencia estadísticamente significativa entre las calificaciones entre los estudiantes del grupo G1 y los estudiantes del grupo G2 antes de la intervención, con una diferencia de medias de 1.84 puntos en las calificaciones finales del grupo de control G2 sobre el grupo experimental G1. Para determinar si efectivamente se logró generar un mejoramiento en las capacidades de resolución de problemas matemáticos luego de la intervención se esperó obtener como resultado que los

estudiantes del grupo experimental pudieran igualar o incluso superar a los estudiantes del grupo de control en el promedio de calificaciones finales obtenidas en la prueba final. Sin embargo se observó que los resultados en el grupo G1 se mantuvieron inferiores a los datos presentados a G2 luego de la aplicación del curso aunque en menor grado. Los resultados mostraron una diferencia de medias presentada de 1.05 puntos en las calificaciones de la prueba final del grupo de control G2 sobre el grupo experimental G1. Para determinar si estas variaciones continuaron siendo significativas en la prueba final, se procedieron a demostrar estadísticamente, realizando el comparativo entre los resultados de la prueba final del grupo experimental y el grupo de control y tomando como base las siguientes hipótesis nulas y alternativas:

Ho ₂ : NO se apreció diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo G1 y los estudiantes del grupo G2 después de la intervención.	Ho ₂ : $\mu_1(\text{postG1}) = \mu_2(\text{postG2})$
Ha ₂ : Se apreció diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo G1 y los estudiantes del grupo G2 después de la intervención.	Ha ₂ : $\mu_1(\text{postG1}) \neq \mu_2(\text{postG2})$

Tabla 14. Análisis estadístico mediante prueba T resultados prueba final (Comparación G1 - G2)

Variable	Supuestos	Prueba Levene calidad varianzas		Prueba t para la igualdad de medias							
		F	Sig.	Revisión de prueba final (Comparación G1 - G2)							
				T	gl	Sig. (bilat)	Dif. medias	Dif. error std.	95% intervalo confianza dif.		
									Inf.	Sup.	
Calificación final	Varianzas iguales	1,416	,237	-2,196	82	,031	-1,05729	,48137	-2,01490	-,09969	
	No varianzas iguales			-2,196	79,163	,031	-1,05729	,48137	-2,01541	-,09917	

Fuente: Autor del proyecto

Se realizó nuevamente una prueba t de Student para muestras independientes con los datos de las calificaciones obtenidas en la prueba final, determinando que se rechaza la hipótesis nula planteada Ho₁ (p<0.05), aceptando la hipótesis alternativa Ha₁, es decir, se apreció evidencia que indicó una diferencia

estadísticamente significativa entre las calificaciones entre los estudiantes del grupo G1 y los estudiantes del grupo G2 después de la intervención, concluyendo así que los resultados NO evidenciaron un cambio estadísticamente significativo en los resultados del grupo experimental comparado contra el grupo de control antes y después de la intervención. No obstante se observó un desmejoramiento mayor en las calificaciones del grupo de control comparado con el grupo experimental. Con el fin de determinar si las variaciones presentadas independientemente por grupo fueron significativas, se procedió a realizar un análisis longitudinal de los datos de las calificaciones entre los resultados del pretest y en la prueba final tanto en el grupo experimental como en el grupo de control y tomando como base las siguientes hipótesis nulas y alternativas:

Ho ₃ : NO se apreció diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo experimental G1 antes y después de la intervención.	Ho ₃ : $\mu_1(\text{preG1}) = \mu_1(\text{postG1})$
Ha ₃ : Se apreció diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo experimental G1 antes y después de la intervención.	Ha ₃ : $\mu_1(\text{preG1}) \neq \mu_1(\text{postG1})$
Ho ₄ : NO se apreció diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo de control G2 antes y después de la intervención.	Ho ₄ : $\mu_1(\text{preG2}) = \mu_1(\text{postG2})$
Ha ₄ : Se apreció diferencia estadísticamente significativa en las calificaciones de los estudiantes del grupo de control G2 antes y después de la intervención.	Ha ₄ : $\mu_1(\text{preG2}) \neq \mu_1(\text{postG2})$

Tabla 15. Análisis estadístico mediante prueba T (Comparación de calificaciones por grupos)

Comparación Pretest – Prueba Final Posttest		Diferencias relacionadas					T	gl	Sig. (bilateral)
		Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
					Inferior	Superior			
Par 1	PuntajeG1Pre PuntajeG1Fin	- ,12309	2,77566	,42829	-,74187	,98804	,287	41	,775
Par 2	PuntajeG2Pre PuntajeG2Fin	- ,90848	3,34677	,51642	-,13445	1,95141	1,759	41	,086

Fuente: Autor de proyecto

Para el análisis longitudinal de los datos de las calificaciones obtenidas se realizó una prueba t de Student para muestras relacionadas, determinando que NO se

rechaza la hipótesis nula planteada H_{03} ($p > 0.05$), es decir, se concluyó que NO se apreciaron evidencias que revelaran diferencias estadísticamente significativas en las calificaciones de los estudiantes del grupo G1 antes y después de la intervención. Además, NO se rechaza la hipótesis nula planteada H_{04} ($p > 0.05$), es decir, se determinó que NO hay evidencia que afirme una diferencia estadísticamente significativa entre las calificaciones de los estudiantes del grupo G2 antes y después de la intervención.

De acuerdo con los resultados obtenidos en las pruebas pretest y final en el grupo experimental y en las comparaciones con el grupo de control, no hay evidencia sistemática suficiente que sustente una variación significativa esperada entre el grupo experimental y el grupo de control para determinar si efectivamente se generó un mejoramiento de las habilidades matemáticas de resolución de problemas mediante la enseñanza de la programación. No obstante se notó académicamente una diferencia notoria entre los dos grupos, dado que los estudiantes del grupo experimental se lograron mantener de una manera más estable en las dos pruebas (disminución de 0.12 puntos) mientras que los del grupo de control no lo lograron hacer (disminución de 0.90 puntos). Por esta razón se procedió a realizar un análisis secundario de las variables usadas para obtener la calificación, con el fin de comprender mejor el comportamiento observado.

4.3.3 Análisis secundario por variable

Revisando los datos de la prueba inicial se pudo observar que los valores de las medias en las variables de preguntas contestadas (PC), operaciones interpretadas (OI), ejecuciones correctas (EC), respuestas acertadas (RA) y los valores de las calificaciones en el grupo G1 fueron menores a los datos presentados a G2 antes de la aplicación del curso, además que hubo más estudiantes en el grupo G2 que terminaron en un tiempo menor al tiempo límite (TML), que los estudiantes en el grupo G1. Por tal razón se procedió a determinar estadísticamente las variaciones significativas de las variables entre los grupos G1 y G2.

Tabla 16. Análisis estadístico mediante prueba T variables evaluadas en pretest (Comparación G1 - G2)

	Asunciones	Prueba Levene calidad varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilat)	Dif. medias	Dif. error std.	95% intervalo confianza diferencia	
									Inf.	Sup.
Preguntas contestadas	Varianzas iguales	1,396	,241	-2,878	82	,005	-,833	,290	-1,409	-,257
	No varianzas iguales			-2,878	77,491	,005	-,833	,290	-1,410	-,257
Operaciones interpretadas	Varianzas iguales	,098	,756	-3,019	82	,003	-,929	,308	-1,540	-,317
	No varianzas iguales			-3,019	81,987	,003	-,929	,308	-1,540	-,317
Ejecuciones correctas	Varianzas iguales	,005	,944	-2,711	82	,008	-,857	,316	-1,486	-,228
	No varianzas iguales			-2,711	81,658	,008	-,857	,316	-1,486	-,228
Respuestas acertadas	Varianzas iguales	,001	,976	-2,612	82	,011	-1,119	,428	-1,971	-,267
	No varianzas iguales			-2,612	81,898	,011	-1,119	,428	-1,971	-,267
Termina antes del tiempo límite	Varianzas iguales	11,744	,001	-1,623	82	,108	-,119	,073	-,265	,027
	No varianzas iguales			-1,623	70,765	,109	-,119	,073	-,265	,027

Fuente: Autor del proyecto

Al realizar el análisis estadístico de los datos de la primera prueba se determinó que los resultados evidenciaron una diferencia significativa en los valores de contestación de preguntas, interpretación y ejecución de operaciones, y sustentación de resultados en los estudiantes del grupo experimental con respecto a los estudiantes del grupo de control antes de aplicar la intervención, pero no revelaron una diferencia significativa en la capacidad de responder a las preguntas en tiempos similares.

Para determinar si efectivamente se logró generar un mejoramiento en las capacidades de resolución de problemas matemáticos luego de la intervención se esperó obtener como resultado que los estudiantes del grupo experimental pudieran igualar o incluso superar a los estudiantes del grupo de control en los

aspectos de contestación de preguntas, interpretación y ejecución de operaciones, y sustentación de resultados. Sin embargo, al hacer la revisión los datos de la prueba final se observó que los valores de las medias en las variables de preguntas contestadas (PC), operaciones interpretadas (OI), ejecuciones correctas (EC), y respuestas acertadas (RA) en el grupo G1 seguían manteniéndose inferiores a los datos presentados a G2 antes de la aplicación del curso. Adicionalmente, se mantuvo la tendencia en los estudiantes en el grupo G2 a terminar en un tiempo menor al tiempo límite (TML), que los estudiantes en el grupo G1.

Tabla 17. Análisis estadístico mediante prueba T variables evaluadas en prueba final (Comparación G1 - G2)

	Asunciones	Prueba Levene calidad varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	GI	Sig. (bilat)	Dif. medias	Dif. error std.	95% intervalo confianza diferencia	
									Inf.	Sup.
Preguntas contestadas	Varianzas iguales	5,939	,017	-,211	82	,833	-,071	,338	-,744	,601
	No varianzas iguales			-,211	71,006	,833	-,071	,338	-,745	,602
Operaciones interpretadas	Varianzas iguales	0,877	,352	-2,289	82	,025	-1,405	,614	-2,626	-,184
	No varianzas iguales			-2,289	80,083	,025	-1,405	,614	-2,626	-,183
Ejecuciones correctas	Varianzas iguales	,898	,346	-3,037	82	,003	-1,643	,541	-2,719	-,567
	No varianzas iguales			-3,037	80,834	,003	-1,643	,541	-2,719	-,567
Respuestas acertadas	Varianzas iguales	,747	,390	-2,517	82	,014	-,976	,388	-1,748	-,205
	No varianzas iguales			-2,517	80,386	,014	-,976	,388	-1,748	-,204
Termina antes del tiempo límite	Varianzas iguales	8,374	,005	-1,383	82	,171	-,071	,052	-,174	,031
	No varianzas iguales			-1,383	61,618	,172	-,071	,052	-,175	,032

Fuente: Autor del proyecto

Estas variaciones se procedieron a demostrar estadísticamente, realizando el comparativo entre los resultados de la prueba final del grupo experimental y el grupo de control. Al realizar el análisis estadístico de los datos de la prueba final se observó que se presentaron evidencias que reflejaron aún una diferencia significativa en la capacidad de interpretación y ejecución de operaciones, y sustentación de resultados en los estudiantes del grupo experimental con respecto a los estudiantes del grupo de control luego de aplicar la intervención. Sin embargo, se apreció un cambio en los resultados, mostrando que luego de aplicar la intervención NO se generaron evidencias que reflejaron diferencia significativa en cuanto al aspecto de contestación de preguntas entre los estudiantes del grupo experimental y los del grupo de control.

Para comprender el comportamiento de esta variable se procedió a realizar un análisis longitudinal de los resultados con el fin de determinar variaciones significativas en las habilidades de contestación de preguntas, interpretación y ejecución de operaciones, y sustentación de las respuestas entre las pruebas pretest y la prueba final, analizando cada grupo por separado. Dado que los datos de medición se establecieron mediante cantidades exitosas, se transformaron los datos en términos de porcentajes exitosos para obtener un parámetro similar de comparación. Al revisar esta nueva información se observó que los niveles en las variables de contestación de preguntas e interpretación de operaciones cambiaron en el grupo experimental, donde se apreció un ligero incremento en el promedio, pero en el grupo de control estos niveles disminuyeron.

Tabla 18. Análisis estadístico mediante prueba T (Comparación de variables por grupos)

Grupo experimental		G1	Prueba de muestras relacionadas				T	gl	Sig. (bilat.)	
			Diferencias relacionadas							
			Dif. Medias Δx	Desv. típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
Inf.	Sup.									
Par 1	PCpre - PCpost	-	-,05187	,24382	,03762	-,12785	,02411	-1,379	41	,175
Par 2	Olpre - Olpost	-	-,00238	,35372	,05458	-,11261	,10785	-,044	41	,965
Par 3	ECpre - ECpost	-	,05238	,33731	,05205	-,05273	,15749	1,006	41	,320
Par 4	RApre - RApost	-	,10459	,31923	,04926	,00511	,20407	2,123	41	,040

Prueba de muestras relacionadas									
Grupo G2 control		Diferencias relacionadas					T	gl	Sig. (bilat.)
		Dif. Medias Δx	Desv. típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
					Inf.	Sup.			
Par 1	PCpre - PCpost	,05825	,28661	,04423	-,03107	,14756	1,317	41	,195
Par 2	Olpre - Olpost	,04286	,43401	,06697	-,09239	,17810	,640	41	,526
Par 3	ECpre - ECpost	,05952	,38702	,05972	-,06108	,18013	,997	41	,325
Par 4	RApre - RApost	,14243	,37966	,05858	,02412	,26074	2,431	41	,020

Fuente: Autor del proyecto

Este dictamen se procedió a comprobar estadísticamente por medio de una prueba t de Student para muestras relacionadas; en este caso se relacionaron los resultados de las pruebas inicial (pretest) y la final en el mismo grupo. La prueba se realizó con las 4 variables usadas para obtener la calificación final (%PC, %OI, %EC, %RA).

Al realizar el análisis estadístico de los datos de las pruebas NO se evidenció una diferencia significativa en el nivel de capacidad de contestación de preguntas e interpretación de operaciones tanto en el pretest como en la prueba final en los estudiantes del grupo experimental. También se observó que NO hubo evidencia que permita establecer una diferencia significativa en el nivel de capacidad de contestación de preguntas e interpretación de operaciones tanto en el pretest como en la prueba final en los estudiantes del grupo control. No obstante, los resultados muestran también un comportamiento atípico: al revisar los valores de las medias en la comparación de las variables de preguntas contestadas y operaciones interpretadas, se observó un incremento obtenido en el grupo experimental para la prueba final, aunque estas variaciones no fueran estadísticamente significativas (PC: $\Delta x = -0.05187$, $p = 0.175$; OI: $\Delta x = -0.00238$, $p = 0.965$), pero no se mantuvo esta misma igualdad en el grupo de control, por lo que se presentó una disminución no significativa (PC: $\Delta x = 0.05825$, $p = 0.195$; OI: $\Delta x = 0.04286$, $p = 0.526$).

4.4 RESULTADO DE LA REVISIÓN DE COMPETENCIAS

El curso aplicado se basó inicialmente en un enfoque basado en objetivos, el cual ha sido usado tradicionalmente en el tratamiento de la asignatura de tecnología e informática. Con el fin de convertir esta propuesta a un enfoque basado en competencias, se realizó una revisión de cuáles competencias debían adquirir los estudiantes según lo estipulado en los estándares que presenta el Ministerio de Educación Nacional con respecto al área de tecnología e informática (Ministerio de Educación Nacional, 2008). Al revisar dicho documento, se encontró que los estándares presentados se refieren a criterios que debe alcanzar el estudiante en un grado determinado, además las competencias que se encontraron fueron referidas en el documento como “indicadores”. No obstante no se encontraron algunas que se relacionen con la adquisición de competencias de desarrollo del pensamiento algorítmico.

Revisando además los estándares de matemáticas para el grado 5° (Ministerio de Educación Nacional, 2003), se encontró que la enseñanza de los algoritmos no es un tema relevante a ser incluido como una competencia, es decir, la competencia referente a la comprensión o aplicación del pensamiento algorítmico no existe en este momento, ni en los estándares de tecnología ni en los estándares de matemáticas. Es por esta razón que se determinó tomar como base las competencias presentadas en el estándar de tecnología para reformular y proponer nuevas competencias que tengan que ver con el pensamiento algorítmico.

Los estándares e indicadores que superficialmente se podrían usar como referencia al desarrollo del pensamiento algorítmico se consignaron en la tabla siguiente.

Tabla 19. Estándares e indicadores existentes identificados para referenciar competencias

Componente	Apropiación y uso de la Tecnología
Estándar	E_01: Describo y explico las características y el funcionamiento de algunos artefactos, productos, procesos y sistemas de mi entorno y los uso en forma segura y apropiada.
Indicadores	I_01: Sigo instrucciones sobre el uso adecuado de artefactos y procesos, que están en

	<p>manuales y otros documentos.</p> <p>I_02: Describo y utilizo, adecuadamente, las tecnologías de la información y la comunicación disponibles en mi entorno, para el desarrollo de diversas actividades (comunicación, entretenimiento, aprendizaje, búsqueda y validación de información, investigación,...).</p> <p>I_03: Describo el funcionamiento y las características de artefactos, procesos y sistemas tecnológicos usando diferentes formas de representación (esquemas, dibujos, diagramas).</p> <p>I_04: Ensambo artefactos y dispositivos sencillos siguiendo instrucciones de texto o esquemáticas.</p>
--	---

Componente	Solución de problemas con tecnología
Estándar	E_02: Describo y analizo las ventajas y desventajas de la utilización de artefactos y procesos, y los empleo para solucionar problemas de la vida cotidiana.
Indicadores	<p>I_05: Diseño y construyo soluciones tecnológicas expresadas en maquetas o modelos que funcionan y cumplen con propósitos previamente establecidos.</p> <p>I_06: Frente a un problema propongo varias soluciones posibles indicando como llegué a ellas, sus ventajas y las dificultades de cada una.</p> <p>I_07: Identifico y comparo ventajas y desventajas de distintas soluciones tecnológicas a un mismo problema.</p> <p>I_08: Identifico fallas sencillas en un artefacto o proceso, actúo en forma segura frente a ésta fallas y realizo propuestas de reparación.</p> <p>I_09: Describo y argumento mis propuestas y decisiones para la solución de problemas.</p>

Fuente: (Ministerio de Educación Nacional, 2008)

Estos estándares e indicadores fueron la base para poder generar una propuesta de nuevas competencias en los que sí se permita medir el desarrollo del pensamiento algorítmico.

Adicionalmente se logró establecer las competencias a partir del objetivo general planteado para el curso. De acuerdo con Miranda Justo de la Rosa, de la Asociación Mundial de Educadores Infantiles (Justo de la Rosa, 2009), se puede generar una programación curricular basada en competencias basado en el siguiente orden:

- Objetivos
- Competencias
- Contenidos
- Actividades
- Sistemas de evaluación

Justo de la Rosa establece también que es posible presentar estos elementos en un orden diferente en el que los sistemas de evaluación se estipulen luego de determinar las competencias. Dado que ya se cuenta con el objetivo del curso, es posible utilizar también como base dicho objetivo para el establecimiento de las competencias a proponer, y a partir de allí determinar sus criterios de evaluación.

4.4.1 Propuesta de competencias a aplicar en un curso de tecnología referente al desarrollo del pensamiento algorítmico

Se establecieron un conjunto de 7 competencias. Las competencias propuestas se siguen agrupando en dos de los componentes presentados en el área de tecnología por el Ministerio de Educación:

- Apropiación y Uso de la Tecnología
- Solución de Problemas con Tecnología

Además se siguen manteniendo los estándares generales de cada uno de los componentes.

Tabla 20. Competencias propuestas por componente

Componente	Apropiación y uso de la Tecnología
Competencia general	E_01: Describo y explico las características y el funcionamiento de algunos artefactos, productos, procesos y sistemas de mi entorno y los uso en forma segura y apropiada.
Competencias específicas propuestas	<p>CP_01: Sigo instrucciones sobre el uso adecuado de aplicaciones software que están en manuales y otros documentos.</p> <p>CP_02: Describo el funcionamiento y las características de algunas secuencias de órdenes y procesos algorítmicos usando diferentes formas de representación (esquemas, dibujos, diagramas, texto).</p>

	CP_03: Diseño y construyo algoritmos expresados en secuencias de órdenes siguiendo instrucciones de texto o esquemáticas, que funcionan y cumplen con propósitos previamente establecidos.
--	---

Componente	Solución de problemas con tecnología
Competencia general	E_02: Describo y analizo las ventajas y desventajas de la utilización de artefactos y procesos, y los empleo para solucionar problemas de la vida cotidiana.
Competencias específicas propuestas	<p>CP_04: Identifico información como un conjunto de datos en diversos contextos que requieran el uso del pensamiento algorítmico para la resolución de un problema</p> <p>CP_05: Frente a un problema propongo varios diseños de algoritmos posibles indicando como llegué a ellas para solucionar un mismo problema, sus ventajas y las dificultades de cada una.</p> <p>CP_06: Identifico fallas sencillas en un algoritmo y realizo propuestas de modificación para solucionar las fallas.</p> <p>CP_07: Aplico el desarrollo del pensamiento algorítmico para la solución de problemas cotidianos</p>

Fuente: Autor del proyecto

Cada una de estas competencias propuestas puede ser fácilmente medible utilizando para ello los objetivos establecidos previamente en cada tema planteado en el plan inicial, y añadiendo nuevos objetivos los cuales pueden presentarse en forma de logros. Se establece la propuesta de la siguiente forma:

Tabla 21. Logros propuestos para medición de competencias

COMPONENTE: APROPIACIÓN Y USO DE LA TECNOLOGÍA		
Competencias propuestas	Logros	Ámbitos temáticos (temas donde se mide)
CP_01: Sigo instrucciones sobre el uso adecuado de aplicaciones software que están en manuales y otros documentos.	<p>L_01: Que el estudiante comprenda el funcionamiento del juego Lightbot para la resolución de las actividades del tema.</p> <p>L_02: Que el estudiante comprenda el funcionamiento del juego Blockly para la resolución de las actividades del tema</p>	Temas 3, 7 y 10

CP_02: Describo el funcionamiento y las características de algunas secuencias de órdenes y procesos algorítmicos usando diferentes formas de representación (esquemas, dibujos, diagramas, texto).	L_03: Que el estudiante comprenda el concepto de algoritmo como una secuencia finita de pasos formados mediante instrucciones definidas y representadas en un formato particular para el cumplimiento de un objetivo	Tema 2
CP_03: Diseño y construyo algoritmos expresados en secuencias de órdenes siguiendo instrucciones de texto o esquemáticas, que funcionan y cumplen con propósitos previamente establecidos.	L_04: Que el estudiante pueda construir un algoritmo lógico para la consecución de un objetivo deseado L_05: Que el estudiante pueda reutilizar instrucciones en un algoritmo lógico por medio de implementaciones de funciones de tipo subrutina L_06: Que el estudiante pueda establecer un conjunto de instrucciones que puedan ser utilizadas a modo de órdenes iterativas con un elemento de control L_07: Que el estudiante pueda establecer un conjunto de instrucciones que puedan definirse dentro de controles condicionales para la consecución de un objetivo específico.	Temas 3, 5, 7 y 8

COMPONENTE: SOLUCIÓN DE PROBLEMAS CON TECNOLOGÍA		
Competencias propuestas	Logros	Ámbitos temáticos (temas donde se mide)
CP_04: Identifico información como un conjunto de datos en diversos contextos que requieran el uso del pensamiento algorítmico para la resolución de un problema	L_08: Que el estudiante pueda identificar datos para ser aplicados en conjunto con las instrucciones adecuadas para la consecución de un objetivo determinado	Tema 10
CP_05: Frente a un problema propongo varios diseños de algoritmos posibles indicando como llegué a ellas para solucionar un mismo problema, sus ventajas y las dificultades de cada una.	L_09: Que el estudiante pueda sugerir otra alternativa de solución en el desarrollo de un algoritmo para la resolución del problema planteado	Temas 5, 7, 8
CP_06: Identifico fallas sencillas en un algoritmo y realizo propuestas de modificación para solucionar las fallas.	L_10: Que el estudiante pueda determinar el resultado de un algoritmo lógico, detectar y corregir errores para la consecución de un	Tema 4

	objetivo deseado	
CP_07: Aplico el desarrollo del pensamiento algorítmico para la solución de problemas cotidianos	<p>L_11: Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo como una secuencias de pasos formados mediante instrucciones definidas para el cumplimiento de un objetivo</p> <p>L_12: Que el estudiante comprenda el concepto de dato, identifique datos en la vida diaria, y pueda identificar los tipos de datos más comunes (textos, números, combinados, lógicos)</p> <p>L_13: Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo mediante instrucciones y datos definidos para el cumplimiento de un objetivo</p>	Temas 6, 9, 11

Fuente: Autor del proyecto

Una vez obtenido el conjunto de logros a obtener se puede establecer las actividades y herramientas de evaluación de estos logros. De acuerdo con Clavijo (Clavijo Clavijo, 2008), dentro de los ámbitos en el estudiante que son objeto de evaluación, el docente debe valorar los aprendizajes referentes a los contenidos en un sentido amplio, es decir, referidos a conceptos y valores (ámbito conceptual), procedimientos (ámbito procedimental) y actitudes (ámbito actitudinal). Dentro de los métodos de evaluación que se pueden utilizar para evaluar el aprendizaje por cada ámbito se proponen los siguientes:

Ámbito de evaluación	Logro (Dimensiones a evaluar)	Herramientas
Conceptual	<p>L_03: Que el estudiante comprenda el concepto de algoritmo como una secuencia finita de pasos formados mediante instrucciones definidas y representadas en un formato particular para el cumplimiento de un objetivo. (Datos, principios)</p> <p>L_12: Que el estudiante comprenda el concepto de dato, identifique datos en la vida diaria, y pueda identificar los tipos de datos más comunes como textos, números, combinados y lógicos. (Datos, principios)</p>	<p>Preguntas F/V y de apareamiento (relación de conceptos)</p> <p>Interpretación de secuencias en diagrama o texto</p>
Procedimental	<p>L_01: Que el estudiante comprenda el funcionamiento del juego Lightbot para la resolución de las actividades del tema. (Adquisición de información del procedimiento)</p> <p>L_02: Que el estudiante comprenda el funcionamiento del juego</p>	<p>Observación indirecta por el docente</p> <p>Solicitud de explicación de pasos de</p>

	<p>Blockly para la resolución de las actividades del tema. (Adquisición de información del procedimiento)</p> <p>L_04: Que el estudiante pueda construir un algoritmo lógico para la consecución de un objetivo deseado. (Ejecución de procedimiento)</p> <p>L_05: Que el estudiante pueda reutilizar instrucciones en un algoritmo lógico por medio de implementaciones de funciones de tipo subrutina. (Ejecución de procedimiento)</p> <p>L_06: Que el estudiante pueda establecer un conjunto de instrucciones que puedan ser utilizadas a modo de órdenes iterativas con un elemento de control. (Ejecución de procedimiento)</p> <p>L_07: Que el estudiante pueda establecer un conjunto de instrucciones que puedan definirse dentro de controles condicionales para la consecución de un objetivo específico. (Ejecución de procedimiento)</p> <p>L_08: Que el estudiante pueda identificar datos para ser aplicados en conjunto con las instrucciones adecuadas para la consecución de un objetivo determinado. (Ejecución de procedimiento)</p> <p>L_09: Que el estudiante pueda sugerir otra alternativa de solución en el desarrollo de un algoritmo para la resolución del problema planteado. (Implicación del estudiante)</p> <p>L_10: Que el estudiante pueda determinar el resultado de un algoritmo lógico, detectar y corregir errores para la consecución de un objetivo deseado. (Implicación del estudiante)</p> <p>L_11: Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo como una secuencias de pasos formados mediante instrucciones definidas para el cumplimiento de un objetivo. (Ejecución de procedimiento)</p> <p>L_13: Que el estudiante pueda aplicar en actividades de la vida diaria los conceptos de algoritmo mediante instrucciones y datos definidos para el cumplimiento de un objetivo. (Ejecución de procedimiento)</p>	<p>procedimientos</p> <p>Observación y análisis de resultados por el estudiante</p>
Actitudinal	-	-

No se tuvo en cuenta en este punto la evaluación en el ámbito actitudinal, ya que ésta implica el conocimiento y respeto a la diversidad personal, incluso con los

propios sesgos del evaluador (Clavijo Clavijo, 2008). Además, se necesitaría más tiempo para evidenciar en el estudiante un cambio de actitud, el cual sería difícil de observar en un entorno con las limitaciones existentes de intensidad horaria.

Con el resultado de esta propuesta de competencias, se logra asociar los contenidos y las actividades plasmadas en el primer plan curricular que se propuso en la institución, generando una nueva propuesta de currículo basado en competencias, el cual se anexa al final del informe.

5. DISCUSIÓN Y CONCLUSIONES

5.1 HALLAZGOS EN LOS RESULTADOS OBTENIDOS

Esta investigación buscó como resultado generar un modelo inicial de trabajo para enseñar a niños de grado 5° de educación básica primaria la elaboración de macroinstrucciones mediante el uso de las tecnologías de la información y de la comunicación con el fin de vincular el estudio de la lógica algorítmica, pretendiendo de esta forma mejorar sus competencias para la resolución de problemas de tipo matemático.

De acuerdo con los resultados obtenidos en las calificaciones finales de las pruebas generadas antes (pretest) y después de la intervención (final) en un grupo experimental de niños de 5° grado de educación básica primaria y al realizar comparaciones de resultados con un grupo de control de estudiantes, no hay evidencia para acotar una variación significativa generada por la intervención. No obstante, los resultados por variables arrojaron tres variaciones importantes:

- En el pretest se presentaron evidencias que revelaron diferencias significativas entre los estudiantes de los dos grupos en el aspecto de contestación de preguntas, pero en la prueba final no se apreciaron estas evidencias que generen diferencia significativa, mostrando que se logró alcanzar un nivel académicamente similar entre los estudiantes de los grupos luego de aplicar la intervención. Es posible que en algunos casos la intervención haya mejorado la autoconfianza en la resolución de problemas de tipo matemático debido a su experiencia previa, sin embargo, no se puede generalizar esta apreciación porque varias de las preguntas contestadas en varias pruebas revisadas no presentaron una sustentación de la operación adecuada, así como hubo algunos casos en los que no se presentó ningún tipo de operación ejecutada, viéndose esto reflejado en los bajos valores presentados de interpretación y ejecución de operaciones, comparando con la cantidad de preguntas contestadas.

- Aunque se mantiene un nivel más bajo en cuanto a la capacidad de interpretación de operaciones por parte del grupo experimental con respecto al grupo de control, se presentó un incremento en el promedio en los datos de la prueba final comparado con el nivel presentado en los datos del pretest. A pesar que los resultados no evidenciaron una diferencia significativa, es necesario acotar esta variación dado que en la prueba esta variable se refirió justamente a la capacidad de poder determinar un conjunto de datos iniciales, determinar una serie de operaciones matemáticas y establecer un orden en la ejecución de las operaciones. Es posible que la intervención haya permitido mantener esta habilidad en los estudiantes, dado que no se presentó ningún tipo de jornada de refuerzo ni actividad extraclase en ninguno de los dos grupos, además se corroboró que se dictaron las mismas temáticas de matemáticas a los dos grupos de estudiantes en las mismas semanas, y que durante el momento de la aplicación del proyecto, los estudiantes no estaban desarrollando ejercicios de repaso de operaciones básicas en la asignatura de matemáticas puesto que debían ver conceptos de geometría. La disminución de la práctica que se evidenció en los estudiantes pudo ser una razón muy importante por la cual los estudiantes disminuyeron la capacidad de interpretación de operaciones, la cual se pudo haber solventado en parte con la aplicación del curso en el grupo experimental.
- En cuanto a la capacidad de interpretación de operaciones todavía se encontró que varios estudiantes no determinaban cuáles debían ser los datos a utilizar ni las operaciones matemáticas correctas y sus secuencias de aplicación, pero algo que causó curiosidad fue la incapacidad, encontrada en ambos grupos, de ejecutar correctamente operaciones matemáticas como la resta o la multiplicación a pesar que esta habilidad debía poseerse ya en los niños de estas edades, por lo que es posible que la intervención no haya generado ningún efecto en la capacidad de ejecución de operaciones matemáticas.

No obstante, otros factores que no se tuvieron en cuenta, como variables externas (entorno familiar, situación ambiental, problemas sociales o académicos) pudieron afectar el rendimiento en la aplicación del pretest o de la prueba final.

En esta investigación se evidenciaron resultados académicos que muestran que el uso de las tecnologías mediante la enseñanza de la programación podría mejorar significativamente las habilidades de resolución de problemas matemáticos en los niños de 5° grado, dado que se pudo notar que académicamente se presentó una mejora en los valores que medían la habilidad en los estudiantes para determinar que se debe llegar a una respuesta particular interpretando la secuencia y los datos correctos para determinar las operaciones requeridas con el fin de solucionar un problema matemático que requiere más pasos de desarrollo, comparado contra un grupo de control en el que no se aplicó la enseñanza de la programación, pudiendo ser posible que en los niños intervenidos que se haya generado una analogía en la forma como adquirieron la capacidad de determinar un conjunto de actividades secuenciales novedosas para ellos y la capacidad de determinar un conjunto de operaciones en situaciones nuevas. Sin embargo, los resultados obtenidos en las pruebas no lograron generar una evidencia sistemática suficiente que permitiera mostrar una variación estadísticamente significativa entre el grupo experimental y el grupo de control con el fin de determinar estadísticamente si hubo eficacia en la aplicación del plan propuesto que involucró la enseñanza de la programación para mejorar las habilidades matemáticas de resolución de problemas. Cabe aclarar que todavía no es concluyente esta afirmación, teniendo en cuenta que solamente se contaba con un corto tiempo de intervención (solo 1 hora semanal de dedicación durante 10 semanas), y además hubo un conjunto de variables externas que no se controlaron y que pudieron afectar el desempeño de los estudiantes en las pruebas.

Por otra parte, además de lograr desarrollar e implementar un plan curricular en la institución, fue posible plantear adicionalmente un conjunto de competencias con las cuales se pueden determinar logros en los estudiantes referentes a la adquisición de las habilidades de pensamiento algorítmico. Esta habilidad que se podría adquirir al trabajar con algoritmos de programación es una competencia que no se ha implementado en los estándares de tecnología del Ministerio de Educación. Incluso, a pesar que en el estándar de matemáticas se menciona muy brevemente la formulación algorítmica, tampoco se encuentran competencias en dichos estándares para medir esta habilidad.

5.2 DISCUSIONES GENERALES

5.2.1 Uso de videojuegos como técnica para enseñar programación a niños

La revisión de la literatura permitió proponer una técnica que implicó el uso de aplicaciones como Blockly y Lightbot para la enseñanza de macroinstrucciones con el fin de desarrollar secuencias órdenes lógicas que pudieron brindar una generación del pensamiento lógico para la resolución de problemas. La intervención desarrollada en la institución educativa mostró que es posible implementar en forma exitosa esta técnica incluso teniendo como factores limitantes (1) la poca intensidad horaria actual asignada en el área de tecnología e informática en los planes curriculares en instituciones oficiales de educación básica primaria en nuestro país, (2) la falta de una especialización docente del área de tecnología en la enseñanza de la programación y en la preparación de recursos digitales específicos en esta temática, (3) la poca importancia que se le brinda a la asignatura tecnología e informática en los planes educativos institucionales, y (4) la carencia de competencias tan importantes como la habilidad de pensamiento algorítmico en los planes curriculares.

En efecto, hay que tener en cuenta que en Colombia a diferencia de otros países se ha enfocado el área de tecnología e informática más hacia la enseñanza de otros conceptos como máquinas, procesos, materiales, y al uso de la tecnología como apoyo al desarrollo de proyectos y otras actividades como búsqueda de información, simulación, diseño asistido, manufactura, representación gráfica, comunicación de ideas, trabajo colaborativo (Ministerio de Educación Nacional, 2008) pero ha desplazado la enseñanza de la programación en sí. Por ello no se han establecido en los perfiles docentes para estas áreas, la capacidad de crear soluciones a situaciones planteadas a partir del establecimiento de secuencias lógicas de órdenes. Esto se comprobó claramente al revisar las preguntas establecidas en las convocatorias del Magisterio para selección docente en el área de tecnología e informática, realizadas en junio del año 2013, en la cual participé. Al hacer una consulta en varias instituciones educativas oficiales, se observó que la intensidad horaria dedicada al área de tecnología e informática es en general de 1 hora a la semana, habiendo encontrado en muy pocos casos un máximo de 2 horas semanales. Todo esto hace que en este momento y en este entorno actual

sea algo complicado acceder a recursos más especializados como personal con experiencia en programación o que implique una mayor dedicación horaria por parte del personal docente, y por ello se ofreció esta técnica que es muy sencilla de aprender, de preparar y de aplicar con los niños, además que la cantidad de tiempo para invertir en la preparación y en la aplicación no es tan considerable como en la aplicación de otras técnicas.

La educación actual colombiana está exigiendo un cambio radical en los procesos de enseñanza en el aula. Es responsabilidad de los educadores y personas que aportamos en el proceso de enseñanza a desarrollar e implementar criterios de calidad para mejorar los procesos educativos. Es ahí donde entra en juego un elemento importante como lo es la utilización del enfoque por competencias en los desarrollos curriculares. Sin embargo no solo entra a participar el estudiante para cumplir estos nuevos retos. Los docentes, las instituciones y el estado colombiano deben buscar las formas más adecuadas de generar situaciones didácticas que impliquen verdaderos desafíos para sus estudiantes y que puedan explotar estas habilidades que han sido descuidadas, con el fin de que puedan alcanzar los logros en los que realmente demuestren que pueden dar soluciones óptimas a problemas, incluso no solo de índole matemática.

Por otro lado, los juegos como Blockly o LightBot pueden proveer un ambiente de trabajo que permite la enseñanza de la programación a los niños, al igual que el experimento de Rogozhkina & Kushnirenko (2011) con el uso de Piktomir, ya que también les permite conocer conceptos como secuencias, subrutinas, repeticiones y los primeros usos de las variables. Adicionalmente el incremento en la motivación al utilizar juegos hizo que los estudiantes pusieran más de su parte para aprender de estos nuevos temas y superar los retos que presentaban los niveles de los juegos.

5.2.2 Enseñanza de la programación como refuerzo en la habilidad de interpretación de operaciones matemáticas

Al revisar los resultados obtenidos en las pruebas se observó un incremento promedio en la variable que evalúa la capacidad de determinar las operaciones matemáticas, los datos y las secuencias de desarrollo de operaciones para dar

solución a un problema en un grupo de estudiantes que fueron sometidos a una intervención en el curso propuesto de programación, en comparación con estudiantes que no fueron intervenidos, los cuales no lograron mantener estos valores de la misma forma, a pesar que no se presentó una variación estadísticamente significativa en esta capacidad en el grupo experimental. Es posible que los estudiantes intervenidos hubieran reforzado su capacidad de determinar las operaciones adecuadas al haber desarrollado y ejercitado un proceso mental de depuración paso a paso, que pudiera brindarles pistas que les ayudaran a comprender, localizar la información y usar los procesos adecuados para resolver el problema, tal como se presentó en la experiencia de Sipitakiat & Nusén (2012). Grover (2009) afirmó también que el pensamiento algorítmico que se pueda desarrollar en los estudiantes, les permite definir más claramente un problema, separarlo en problemas más pequeños y manejables y definir de mejor forma un conjunto de pasos para resolverlo. Paulo Blikstein, citado en (National Academy of Sciences, 2010, pág. 34) destacó que puede haber una oportunidad de utilizar el pensamiento algorítmico para representar procesos y relaciones complejas incluso de manera más comprensible que las matemáticas, dado que tanto las matemáticas y el pensamiento computacional son herramientas usadas para representación.

En particular se esperaba que hubiera una variación estadísticamente significativa en la habilidad de interpretación de operaciones matemáticas en los estudiantes, por ello no se puede argumentar que la intervención en el curso haya sido la única causante de esta variación, dado que es posible que otras variables externas fuera del alcance de este experimento hayan intervenido en el desempeño de los estudiantes al aplicar la prueba final, así como también es posible que el corto tiempo de aplicación tanto en intensidad horaria como en tiempo total no haya sido suficiente para generar un mejoramiento significativo. La intervención de Burke y Kafai (2010), por ejemplo, fue una de las experiencias en las que menos tiempo se trabajó con los niños (dos sesiones semanales después de clases), mayor al tiempo de intervención de este proyecto. Otro ejemplo fue el tiempo de trabajo de Javidi (2009) con los niños, el cual tomó 10 a 15 sesiones los días sábados desde las 9:00am hasta las 3:00pm. Debido a esto se plantea una nueva hipótesis para un trabajo posterior, el cual permitiría determinar si el incremento en el tiempo de intervención en este tipo de enseñanza pueda mejorar de manera significativa la

habilidad de interpretación de operaciones matemáticas, tratando además de establecer otras posibles variables que pudieran afectar este desempeño.

5.2.3 No afectación en la habilidad de ejecución de operaciones matemáticas

Aunque se vio una leve discrepancia en los valores que representaron la habilidad de interpretación de operaciones en un problema matemático, no se obtuvieron resultados satisfactorios en la habilidad de ejecución de los cálculos, a pesar que se determinaba correctamente las operaciones matemáticas a realizar, la secuencia de operación y los datos correctos. Particularmente, cálculos como restas o multiplicaciones no se ejecutaban correctamente, viéndose entre los errores más frecuentes el intercambiando el orden de los dígitos entre el minuendo y el sustraendo para las restas, olvidando el proceso de “restar llevando” y realizando multiplicaciones sin colocar ordenadamente los números que se adquieren en el proceso con cada dígito del multiplicador. La falla en esta habilidad es más operativa que lógica, es por ello que no se pudo generar una afectación en esta capacidad al aplicar la intervención. No obstante podría recomendarse la implementación de nuevas aplicaciones de tipo juego que permitan, mediante el establecimiento de secuencias de pasos, comprender los procesos operativos más complejos como las restas de números grandes y las multiplicaciones con el fin de mejorarles esta habilidad.

5.3 CONCLUSIONES

Esta investigación inició basada en la siguiente hipótesis: ¿De qué manera es posible mejorar el desarrollo de las capacidades de razonamiento para resolución de problemas, en los niños de grado 5° de educación básica primaria, en algunas instituciones de educación de Bucaramanga mediante el uso de las Tecnologías de la Información y de la Comunicación?. A partir de este gran interrogante, surgieron otras preguntas más específicas al inicio de esta experiencia, las cuales

pudieron ser ahora contestadas a raíz de los resultados obtenidos con el fin de dar respuesta a la hipótesis planteada.

El resultado de la primera fase de este estudio determinó que es posible presentar la temática de la lógica y la secuenciación para la resolución de problemas basados en la enseñanza de los mismos temas que comprenden la generación de algoritmos, como lo son la secuenciación de órdenes, la comprensión y uso de los datos y variables, el uso de estructuras de control tanto condicionales como iterativas y el uso de subrutinas o funciones de reutilización. Dentro de las metodologías y técnicas de enseñanza adecuadas para orientar en esta temática en particular se encontró la técnica relacionada con el uso de herramientas de tipo videojuego como Blockly y LightBot, las cuales son de fácil acceso y usabilidad, y permiten mediante la organización de instrucciones por medio de íconos, generar que ciertos elementos en el juego ejecuten acciones de acuerdo a la secuencia presentada por los usuarios, siendo estas herramientas adecuadas para la enseñanza de estos temas (*resultado del objetivo 1*). Esta técnica ha sido utilizada en varios países como Estados Unidos, Rusia y otros países aventajados en el área de las tecnologías de la información. A pesar que otras técnicas han sido también implementadas en dichos países, se concluyó que la técnica del uso de los videojuegos, dada su facilidad de aplicación y su tiempo corto requerido para preparación, hace que sea una metodología cuya implementación en un entorno educativo puede ser viable para aplicar en el plan de área de tecnología e informática, sin necesidad de generar mayores inversiones en adquisición de personal, en inversión de grandes cantidades de tiempo o recurso especializado, y sin salirse del cumplimiento de los estándares de competencias establecidos para esta área (*resultado del objetivo 2*).

Se determinó también por medio de pruebas comparativas entre un grupo experimental y un grupo de control de estudiantes de 5° grado de educación básica primaria de un colegio oficial en el municipio de Girón que la enseñanza en la lógica de la programación basada en el uso de videojuegos por medio de la aplicación de un curso durante un tiempo corto (*resultado del objetivo 3*) no generó una evidencia sistemática que muestre un mejoramiento estadísticamente significativo en la habilidad de resolución de problemas matemáticos en estudiantes de este nivel. Esta metodología por sí sola no es suficiente para mejorar esta habilidad en su totalidad como se pudo apreciar en los datos

obtenidos (*resultado del objetivo 4*), pero académicamente se observó que los estudiantes del grupo experimental mantuvieron valores estables en las calificaciones, mientras que el grupo de control bajó sus calificaciones en las pruebas. Estos resultados reflejaron que académicamente la intervención pudo haber logrado que los estudiantes mantuvieran su habilidad de resolución de problemas, dejando abierta la posibilidad para generar más estudios que indiquen si incrementando en el tiempo de aplicación en este tipo de enseñanza pueda ayudar a mejorar de manera significativa sus habilidades matemáticas, pudiendo de esta manera detectar otros factores y ayudar así a los niños de estas edades a intensificar en parte su capacidad de razonamiento para la resolución de problemas matemáticos.

5.4 LIMITANTES Y RECOMENDACIONES PARA TRABAJO FUTURO

Dentro de las limitantes que se tuvieron en cuenta en la aplicación de este proyecto, se destaca que el estudio comprendió la generación de un plan curricular que abarcó solamente un bimestre académico para un grupo de estudiantes de nivel 5° del Colegio San Juan de Girón Sebe Primaria “Eloy Valenzuela” del municipio de Girón, Santander, con condiciones óptimas de infraestructura tecnológica en el colegio, con recursos suficientes para permitir el uso del computador de a 2 estudiantes por equipo, con un tiempo de dedicación de una hora semanal para la clase de Tecnología para poder desarrollar la intervención; y aplicando un plan de área diseñado para personal docente que no contara con conocimientos avanzados de programación. Como trabajo futuro se pretenderá comprobar si es posible que el incremento en el tiempo de intervención en la enseñanza de la programación es un factor determinante que permita mejorar de manera significativa la habilidad de interpretación de operaciones matemáticas. Adicionalmente se pretenderá establecer otras posibles variables que pudieran afectar el desempeño en las pruebas de medición con el fin de generar planes de contingencia para controlar dichas variables.

La metodología enfocada en el uso de los videojuegos para la enseñanza de macroinstrucciones puede implementarse en el plan de área de tecnología e

informática con el fin de ayudar al reforzamiento en la habilidad de resolver problemas matemáticos. No obstante se recomienda tener en cuenta que debe complementarse con otras actividades adicionales que permitan mejorar en su capacidad de resolver correctamente las secuencias de operaciones matemáticas planteadas, dado que no es suficiente determinar los pasos para su solución sino que también es necesaria su adecuada ejecución. Esto podría lograrse mediante la implementación de nuevas aplicaciones que permitan afianzar los procesos operativos más complejos en los niños como las restas de números grandes y las multiplicaciones.

Se recalca nuevamente que no se pretendió hacer un recorrido por todas las técnicas existentes para la enseñanza de la programación ni hacer un debate sobre cuáles son las mejores sino que se seleccionó la más apropiada, basada en investigaciones previas y restringida por las limitaciones particulares que se presentaron en este proyecto. Por ello, como trabajo futuro se pretenderá también trabajar con otras técnicas para la enseñanza de la programación diferentes al uso de los videojuegos, con el fin de generar un comparativo en la eficacia de estas, midiendo nuevamente sus efectos en la habilidad de resolución de problema/s de tipo matemático en los estudiantes.

6. BIBLIOGRAFÍA

- Abelson, H., Sussman, G. J., & Sussman, J. (1996). *Structure and Interpretation of Computer Programs*. Boston: The MIT Press.
- Abramovich, S. (Enero de 2013). Computers in Mathematics Education: An Introduction. *Computers in the Schools*, 30(1-2), 4-11.
- Ackermannn, E. (2002). *Piaget's Constructivism, Papert's Constructionism: What's the difference?* . MIT Media Lab.
- Agina, A. (Julio de 2012). The Effect of Nonhuman's External Regulation on Young Children's Creative Thinking and Thinking Aloud Verbalization During Learning Mathematical Tasks. *Computers in Human Behavior*, 28(4), 1213-1226.
- Akihabara News. (04 de 07 de 2013). *Japanese Robots: Kids' Summer School for Robotics & Engineering in Rural Japan*. Obtenido de NPO Hito Project's Robot Summer School: <http://en.akihabaranews.com/136473/toy/japanese-robots-kids-summer-school-for-robotics-engineering>
- Albornoz, M. E. (2007). *El aprendizaje según Piaget*. Obtenido de Mayeutica Educativa:
http://mayeuticaeducativa.idoneos.com/index.php/348494#El_Aprendizaje_seg%C3%BAAn_Piaget
- Alt, C., Astrachan, O., Forbes, J., Lucic, R., & Rodger, S. (2006). Social Networks Generate Interest in Computer Science. *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (págs. 438-442). New York: ACM.
- Amézquita Zárate, P., Contreras Pineda, J., & Pardo Romero, M. A. (1997). *La comunidad educativa frente al neoliberalismo*. Bogotá: Centro de Estudios Por La Nueva Cultura.
- Armor Games. (2005). *armorgames.com*. Recuperado el 31 de Enero de 2014, de Página principal: <http://armorgames.com/>

- Barrera Osorio, F., & Leigh L., L. (2009). *The Use and Misuse of Computers in Education: Evidence from a Randomized Experiment in Colombia*. World Bank Human Development Network Education Team.
- Bauer, J., & Kenton, J. (2005). Toward Technology Integration in the Schools: Why It Isn't Happening. *Journal of Technology and Teacher Education*, 519-546.
- Burke, Q., & Kafai, Y. B. (2010). Programming & Storytelling: Opportunities for Learning About Coding & Composition. *Proceedings of the 9th International Conference on Interaction Design and Children* (págs. 348-351). Barcelona: ACM.
- Cárdenas, M. (2001). *Economic Growth in Colombia: A Reversal of 'Fortune'?*. CID Working Paper No. 83. Cambridge: Center for International Development at Harvard University.
- Clavijo Clavijo, G. A. (2008). *La evaluación del proceso de formación*. Cartagena. Recuperado el 28 de julio de 2014, de http://www.colombiaaprende.edu.co/html/productos/1685/articles-178627_ponen7.pdf
- Code.org. (2013). *Code.org*. Obtenido de Página institucional: <http://code.org/>
- Coderise.org. (2012). *Coderise.org*. Obtenido de Página institucional: <http://coderise.org/index.html>
- Comisión Económica Para América Latina y el Caribe - CEPAL. (2012). *Contribución al crecimiento económico de las tecnologías de la información y las comunicaciones y de la productividad en la Argentina, el Brasil, Chile y México - Serie Estudios Estadísticos y Prospectivos*. CEPAL - División de Desarrollo Económico.
- Computer Science Education Week. (2014). *csedweek.org*. Obtenido de Página institucional: <http://csedweek.org/>
- Corporación para la Nutrición Infantil - CONIN. (1988). Desarrollo cerebral en el niño. *Revista Creces - Ciencia y Tecnología*.

- Delgado, J., Güell, J., García, J., Conde, M., & Casado, V. (2013). Aprendizaje de la programación en el Cutilab. *Revista Iberoamericana de Ciencia, Tecnología y Sociedad*, 123-133.
- Doerschuk, P., Liu, J., & Mann, J. (2012). An INSPIRED game programming academy for high school students. *Proceedings - Frontiers in Education Conference, FIE*.
- Eggers, W. D. (2005). *Government 2.0: Using Technology to Improve Education, Cut Red Tape, Reduce Gridlock, and Enhance Democracy*. Maryland: Rowman & Littlefield Publishers, Inc.
- Felleisen, M., Fidler, R., Flatt, M., & Krishnamurthi, S. (2009). A functional I/O system*: Or, fun for freshman kids. *Proceedings of the ACM SIGPLAN International Conference on Functional Programming* (págs. 47-58). ICFP.
- Fidge, C., & Teague, D. (2009). Losing Their Marbles: Syntax-free Programming for Assessing Problem-solving Skills. *Proceedings of the Eleventh Australasian Conference on Computing Education*. 95, págs. 75-82. Darlinghurst: Australian Computer Society, Inc.
- Fundación Gabriel Piedrahita Uribe. (2013). *Proyecto Scratch Motorola IV - Informe Final Abril 2013*.
- Giannakos, M., & Jaccheri, L. (2013). What Motivates Children to Become Creators of Digital Enriched Artifacts? *Proceedings of the 9th ACM Conference on Creativity & Cognition* (págs. 104-113). New York: ACM.
- Gomez, P. (2010). *Diseño curricular en Colombia: el caso de las matemáticas*. Granada: Universidad de Granada. Obtenido de <http://funes.uniandes.edu.co/651/>
- Great Schools. (02 de 2013). *Is the next second language JavaScript?* Obtenido de Should all kids learn to code?: <http://www.greatschools.org/parenting/learning-development/5894-javascript-class-learn.gs?page=all>

- Grover, S. (Noviembre de 2009). Computer Science Is Not Just for Big Kids. *Learning & Leading with Technology*, 37(3), 27-29.
- Hug, S., Guenther, R., & Wenk, M. (2013). Cultivating a K12 Computer Science Community: A Case Study. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (págs. 275-280). New York: ACM.
- Instituto Colombiano para la Evaluación de la Educación - ICFES. (2010). *Colombia en PISA 2009 - Síntesis de Resultados*. Bogotá: Cadena.
- Instituto Colombiano para la Evaluación de la Educación - ICFES. (2013). *Resultados del grado Quinto en el área de Matemáticas*. Obtenido de Resultados históricos 2002 - 2005 - 2009 - 2012 - 2013: <http://www.icfessaber.edu.co/historico.php/graficar/nacion/id/1/grado/5/tipo/2>
- Instituto de Nuestra Señora de la Asunción - INSA. (2012). *Currículo INSA de Informática 2012*. Obtenido de Eduteka: <http://www.eduteka.org/tag/inicio/insa/1>
- Janalta Interactive Inc. (2010). *Macroinstructon*. Recuperado el 4 de Diciembre de 2013, de Techopedia.com: <http://www.techopedia.com/definition/24802/macro-instruction>
- Javidi, G., & Sheybani, E. (Enero de 2009). Digispired: Digital Inspiration for Interactive Game Design and Programming. *Journal of Computing Sciences in Colleges*, 24(3), 144-150.
- Justo de la Rosa, M. (2009). Competencias en Educación Infantil [Video]. *Congreso Internacional "Fortaleciendo las Competencias: Nuevas estrategias, nuevos aprendizajes"*. Madrid. Recuperado el 27 de julio de 2014, de <https://www.youtube.com/watch?v=HIHZHourDh4&index=9&list=PL2E599E1C578E229E>
- Kafai, Y. B. (1996). Software by kids for kids. *Communications of the ACM*, 39(4).

- Kelleher, C., & Pausch, R. (Junio de 2005). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Computing Surveys*, 37(2), 83-137.
- Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (págs. 1455-1464). New York: ACM.
- Kirriemuir, J., & McFarlane, A. (2004). *Futurelab Series. Report 8: Literature Review in Games and Learning*. Bristol: Futurelab.
- Kliegman, R., Stanton, B., Schor, N., St. Geme III, J., & Behrman, R. (2011). *Nelson Textbook of Pediatrics*. Philadelphia: Elsevier Saunders Inc.
- Lameras, P., Smith, D., Moumoutzis, N., Christodoulakis, S., Ovcin, E., & Stylianakis, G. (2010). Transforming teaching and learning: Changing the pedagogical approach to using educational programming languages. *17th Association for Learning Technology Conference (ALT-C 2010)*. Nottingham: ALT-C.
- Lee, M., & Ko, A. (2011). Personifying Programming Tool Feedback Improves Novice Programmers' Learning. *Proceedings of the Seventh International Workshop on Computing Education Research* (págs. 109-116). New York: ACM.
- Lin, C.-C., Zhang, M., Beck, B., & Olsen, G. (2009). Embedding Computer Science Concepts in K-12 Science Curricula. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (págs. 539-543). New York: ACM.
- Liu, C.-C., Cheng, Y.-B., & Huang, C.-W. (Noviembre de 2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, 57(3), 1907-1918.
- Maloney, J., Rusk, N., Burd, L., Silverman, B., Kafai, Y., & Resnick, M. (2004). Scratch: A sneak preview. *Second International Conference on Creating, Connecting and Collaborating Through Computing*.

- Martín Fraile, B. (2011). Teorías educativas que subyacen en las prácticas docentes. (U. d. Salamanca, Ed.) *Teoría de la Educación*, 23(1), 45-70.
- Meyers, A., Cole, M., Korth, E., & Pluta, S. (2009). Musicomputation: Teaching Computer Science to Teenage Musicians. *Proceedings of the Seventh ACM Conference on Creativity and Cognition* (págs. 29-38). New York: ACM.
- Ministerio de Educación Nacional. (1994). *Ley General de Educación - Ley 115 de Febrero 8 de 1994*. Bogotá: Congreso de la República de Colombia.
- Ministerio de Educación Nacional. (2001). *El Constructivismo como modelo pedagógico*. Recuperado el 04 de Diciembre de 2013, de Colombia Aprende: http://www.colombiaprende.edu.co/html/docentes/1596/articulos-169653_archivo.doc
- Ministerio de Educación Nacional. (2003). *Estándares Básicos de Competencias de Matemáticas*. Bogotá.
- Ministerio de Educación Nacional. (2007). *Foro Educativo Nacional: Enfrentar un problema es encontrar un mundo de soluciones. Rueda de experiencias*. Bogotá: Ministerio de Educación Nacional.
- Ministerio de Educación Nacional. (2008). *Ser competente en tecnología: ¡Una necesidad para el desarrollo! Orientaciones generales para la educación en tecnología*. Bogotá: Imprenta Nacional.
- Ministerio de Educación Nacional. (2010). *Encuesta Nacional de Deserción Escolar (ENDE)*. Bogotá: Ministerio de Educación Nacional.
- Ministerio de Educación Nacional. (2013). *Objetivos Misionales*. Recuperado el 25 de mayo de 2013, de Página Institucional: <http://www.mineduacion.gov.co/1621/w3-article-85244.html>
- Ministerio de Educación Nacional. (2013). *Plan de estudios*. Recuperado el 11 de marzo de 2014, de Glosario: <http://www.mineduacion.gov.co/1621/article-79419.html>

- Ministerio de Tecnologías de la Información y las Comunicaciones. (2011). *Talento Digital*. Recuperado el 10 de Feb de 2014, de Sitio web de Talento Digital, Programa Gobierno en línea: <http://www.talentodigital.gov.co/>
- Miranda, M. J. (2007). *Culturas juveniles y nuevas tecnologías*. San Nicolás: Instituto Superior de Formación Docente No. 127.
- MIT Media Lab. (2013). *Scratch Project*. Obtenido de Página institucional: <http://scratch.mit.edu/>
- Mitchell, A., & Savill-Smith, C. (2004). *The use of computer and video games for learning. A review of the literature*. London: Learning and Skills Development Agency.
- National Academy of Sciences. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. Washington, D.C.: National Academies Press.
- Nickerson, R. S., & Zoghates, P. P. (1988). *Technology in Education: Looking Toward 2020*. New York: Routledge.
- OECD Programme for International Student Assessment. (2010). *PISA 2009 Results: What Students Know and Can Do: Student Performance in Reading, Mathematics and Science (Volume I)*. Paris: OECD.
- Organización de las Naciones Unidas Para la Agricultura y la Alimentación. (1999). *Elaboración Participativa de Planes de Estudios para la Educación y Capacitación Agrícola*. Roma: Departamento de Desarrollo Sostenible FAO.
- Osborne, J., & Hennessy, S. (2003). *Futurelab Series. Report 6: Literature Review in Science Education and the Role of ICT: Promise, Problems and Future Directions*. Bristol: Futurelab.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc.
- Pulido, D., & Velasco, L. (2009). *Proyecto de Vida: Una alternativa para la prevención de la deserción escolar*. Bogotá: Universidad de la Sabana.

- Real Academia Española. (2001). *Algoritmo*. Recuperado el 4 de 12 de 2013, de Diccionario de la lengua española: <http://lema.rae.es/drae/?val=algoritmo>
- Repenning, A., & Ioannidou, A. (Marzo de 2008). Broadening Participation through Scalable Game Design. *SIGCSE Bulletin*, 40(1), 305-309.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (Noviembre de 2009). Scratch: Programming for All. (ACM, Ed.) *Communications of the ACM*, 52(11), 60-67.
- Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (Enero de 2011). A CS0 Course Using Scratch. *Journal of Computing Sciences in Colleges*, 26(3), 19-27.
- Rodger, S., Bashford, M., Dyck, L., Hayes, J., Liang, L., Nelson, D., & Qin, H. (2010). Enhancing K-12 Education with Alice Programming Adventures. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education* (págs. 234-238). New York: ACM.
- Rogozhkina, I., & Kushnirenko, A. (2011). PiktoMir: Teaching programming concepts to preschoolers with anew tutorial environment. *World Conference on Educational Technology Research. Procedia - Social and Behavioral Sciences*. Moscu.
- Sancho Gil, J. M., & et.al. (2006). *Tecnologías para transformar la educación*. Madrid: Universidad Internacional de Andalucía.
- Schwartz, J., Stagner, J., & Morrison, W. (2006). Kid's Programming Language (KPL). *ACM SIGGRAPH 2006 Educators Program* (págs. 52.1-52.4). Boston: ACM.
- Sipitakiat, A., & Nusen, N. (2012). Robo-Blocks: Designing Debugging Abilities in a Tangible Programming System for Early Primary School Children. *Proceedings of the 11th International Conference on Interaction Design and Children* (págs. 98-105). New York: ACM.
- Skinner, B. (1977). *Sobre el conductismo*. Barcelona: Fontanella.

- Tapscott, D. (2008). *Grown Up Digital*. New York: Mc.Graw-Hill.
- Tarkan, S., Sazawal, V., Druin, A., Golub, E., Bonsignore, E., Walsh, G., & Atrash, Z. (2010). Toque: Designing a Cooking-based Programming Language for and with Children. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (págs. 2417-2426). New York: ACM.
- Tomcsányiová, M., & Tomcsányi, P. (2011). Little beaver - A new bebras contest category for children aged 8-9. *5th International Conference on Informatics in Schools: Situation, Evolution and Perspectives, ISSEP 2011*. Bratislava: ISSEP.
- Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (Noviembre de 2010). Alice, Greenfoot, and Scratch - A Discussion. *ACM Transactions on Computing Education*, 10(4), 17.1 - 17.11.
- Valente, A. (2004). Exploring theoretical computer science using paper toys (for kids). *ICALT '04 Proceedings of the IEEE International Conference on Advanced Learning Technologies*. Washington.
- Velez, E., Schiefelbein, E., & Valenzuela, J. (1994). Factores que Afectan el Rendimiento Académico en la Educación. Revisión de la Literatura de América Latina y El Caribe. *Revista Latinoamericana de Innovaciones Educativas*.
- Walton-Hadlock, M. (2008). Tots to Tweens: Age-Appropriate Technology Programming for Kids. (A. L. Association, Ed.) *Children & Libraries: The Journal of the Association for Library*, 6(3), 52-55.
- Wen-Yu Lee, S., & Tsai, C.-C. (2013). Technology-supported Learning in Secondary and Undergraduate Biological Education: Observations from Literature Review. (Springer, Ed.) *Journal of Science Education and Technology*, 22(2), 226-233.
- Wolz, U., Leitner, H., Malan, D., & Maloney, J. (2009). Starting with scratch in CS 1. *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*.

Zuckerman, O., Arida, S., & Resnick, M. (2005). Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (págs. 859-868). New York: ACM.

ANEXOS

Anexo 1. Formato de carta de consentimiento presentada a los padres de familia

Yo _____ identificado(a) con C.C. _____ expedida en _____, como representante y en condición de padre () madre () acudiente (), autorizo a la Universidad Autónoma de Bucaramanga - UNAB, representada únicamente para propósitos de esta actividad por el Ing. Carlos Andrés Palma Suárez, estudiante de maestría de la UNAB, al Colegio San Juan de Girón Sede B – Concentración Escolar Eloy Valenzuela y demás participantes en el pilotaje de Implementación de Propuesta Curricular del Área de Tecnología, Computación e Informática para el grado 5° de Educación Básica Primaria, para que se realice registro fílmico, fotográfico y/o valorativo de mi hijo(a) _____ identificado con R.C.() T.I. () _____ expedido en _____ con el propósito de sustentar y documentar el desarrollo de las actividades que se vienen realizando en el marco del proyecto Metodología para la Enseñanza Inicial de Macroinstrucciones y Procesos Lógico-Matemáticos a Niños de Grado Quinto en Instituciones Oficiales de Educación Básica Primaria. Del mismo modo autorizo para que los registros se incluyan de forma parcial o total en cualquier soporte audiovisual para efectos de reproducción y comunicación pública para los fines y dentro de los propósitos establecidos en el proyecto.

En virtud de este documento, declaro que renuncio a recibir contraprestación alguna con ocasión de la utilización de los registros fílmicos, fotográficos o valorativos de mi hijo(a), y garantizo que puedo otorgar la presente autorización sin ningún tipo de limitación y responderé por cualquier reclamo que se pueda presentar por falta de competencia para suscribir el presente documento, exonerando de cualquier responsabilidad a las personas y entidades participantes en el proyecto. La autorización que aquí se concede sobre los registros es exclusiva de las entidades participantes en el proyecto en mención y podrán ser difundidos en cualquier medio de comunicación a nivel nacional o internacional con fines sin ánimo de lucro por los sistemas de medios escritos, radio o televisión abierta o cerrada en el territorio nacional o en el exterior si así se requiriese.

Atentamente,

Firma del padre o madre: _____

C.C.: _____

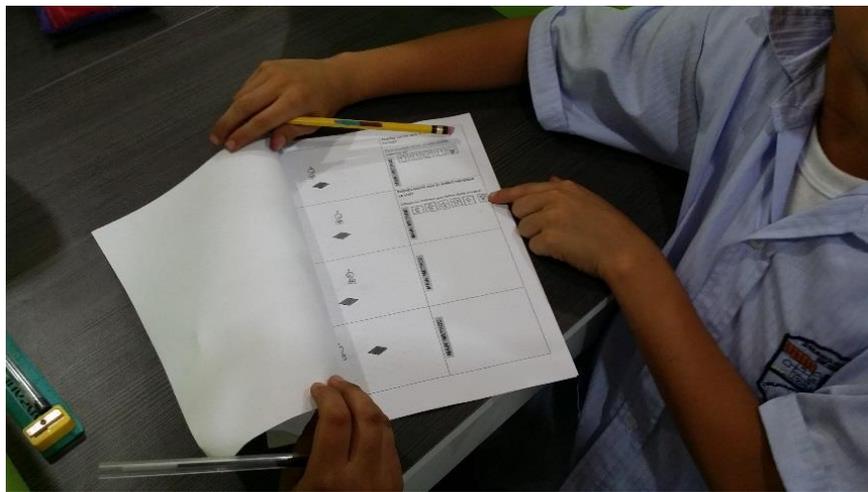
Dirección: _____

Teléfono / Celular: _____

Correo electrónico: _____

Anexo 2. Fotografías tomadas durante la aplicación del curso en el Colegio San Juan de Girón







Fuente: Autor del proyecto

Anexo 3. Propuesta de plan curricular enfocado en competencias

Se entrega documento en archivo digital aparte.

Nombre del archivo: Plan Curricular Tecnología y Computación 5 competencias final.doc