

CHILDPROGRAMMING-C: EXTENDIENDO CHILDPROGRAMMING DESDE LA PERSPECTIVA DE LA INGENIERÍA DE LA COLABORACIÓN



Universidad
del Cauca

ANA MARÍA CHIMUNJA GONZÁLEZ

**Director: Ph.D. César Alberto Collazos Ordoñez
Co-Director: Ph D. Julio Ariel Hurtado Alegría**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Maestría en Computación
Grupo de Investigación IDIS
Áreas de Investigación: Ingeniería de la Colaboración
Popayán, abril de 2018**

ANA MARÍA CHIMUNJA GONZÁLEZ

CHILDPROGRAMMING-C: EXTENDIENDO
CHILDPROGRAMMING DESDE LA PERSPECTIVA DE LA
INGENIERÍA DE LA COLABORACIÓN

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Magister en: Computación

Director: Ph.D. César Alberto Collazos
Co-Director: Ph D. Julio Ariel Hurtado Alegría

Popayán

2018

A Dios por darme la sabiduría para investigar y plasmar aquí todo mi trabajo.

*A mi familia, en especial a mi madre que cada día me ha
enseñado lo maravillosa*

*que es la vida cuando tenemos sueños y vivimos para
lograrlos sin importar las dificultades.*

*A mis amigos que incondicionalmente
siempre han creído en mí.*

Agradecimientos

En este momento llegan a mi mente muchos de los recuerdos cuando estaba iniciando mi carrera sin tener claro cuál iba a ser mi rumbo dentro de la Ingeniería de Sistemas, hoy después de muchos años no me arrepiento del camino que elegí al continuar por el camino de la investigación; esta maestría ha sido uno de los mejores regalo que hasta ahora me ha dado la vida, por eso quiero agradecerle a Dios porque a pesar de los inconvenientes de salud este sueño es realidad, también le agradezco a mi papá, mi mamá por su apoyo incondicional y ser ese motor que me impulsa a seguir mis metas y cumplir mis sueños.

Mis más grandes agradecimientos a esa persona que ha sido un gran ejemplo a seguir, nada de esto sería posible sin la guía del PhD. César Collazos quien me orientó, aminó y me llamó la atención cuando fue necesario, gracias porque con usted aprendí que no tenemos límites y debemos perder el miedo al riesgo si queremos lograr los sueños, de igual manera quiero agradecerle al Doctor Julio Ariel Hurtado por toda su comprensión, enseñanzas y sobre todo por su valiosa amistad, gracias por escucharme y guiarme cuando lo necesité; durante estos años de maestría ustedes han hecho de mí una mejor persona e ingeniera, y han sido las luces del camino que me han inspirado para ser mejor.

De igual manera agradezco al grupo de Investigación y Desarrollo en Ingeniería del Software (IDIS), al comité de la Maestría en Computación y la Vicerrectoría Académica de Investigaciones de la Universidad del Cauca, por todo su apoyo en los eventos y propuestas presentadas.

A Mauricio Muñoz, Fernando Meriño, Rony Veloza y Maria Elena Chimunja, por toda la experiencia compartida de sus muchos años como docentes de niños, y sus aportes en los estudios de caso realizados en esta investigación, finalmente agradezco a las directivas y estudiantes de la Institución Educativa Agroempresarial Huasanó, por permitirme incentivar la programación de software en mis estudiantes y poder iniciar un semillero de programación con los niños de la institución.

Tabla de contenido

1. Capítulo 1	13
1.1 Introducción	13
1.2 Descripción del problema.....	15
1.3 Objetivos.....	18
1.4 Hipótesis	18
1.5 Aporte investigativo.....	19
2. Base conceptual	21
2.1 Ingeniería de la Colaboración	21
2.2 Trabajo colaborativo	22
2.3 Principios básicos del trabajo colaborativo.....	23
2.4 Aprendizaje colaborativo.....	23
2.5 Patrones y subpatrones de colaboración	24
2.5.1 Subpatrones de colaboración	25
2.6 Thinklets de colaboración	26
2.6.1 Mapa de selección de thinklets	30
2.7 Modelo ChildProgramming.....	31
2.7.1 Gamificación en ChildProgramming.....	33
2.8 Metodologías utilizadas para el desarrollo del proyecto	33
2.8.1 Método CSACE (Case Study based Analysis in Collaboration Engineering)....	33
2.8.2 Metodología para el desarrollo de proceso Colaborativos.....	35
2.9 Trabajos relacionados.....	36
3. Capítulo 3. Identificando actividades colaborativas	43
3.1 Estudios de casos exploratorios.....	43

3.2	Resultados de los estudios de casos exploratorios	44
3.2.1	Estudio de caso 1	44
3.2.2	Estudio de caso 2.	46
3.3	Formulación de ChildProgramming como un proceso colaborativo	48
3.3.1	Identificando las actividades del modelo ChildProgramming que requieren trabajo colaborativo para su ejecución.....	48
3.4	Definición de actividades colaborativas de ChildProgramming.....	60
3.4.1	Asociación de patrones de colaboración a las subactividades	62
3.5	Patrones de colaboración propuestos	69
4.	Capítulo 4. Modelo ChildProgramming – C	73
4.1	Mejoras identificadas del modelo ChildProgramming	73
4.1.1	Sensibilizar a los estudiantes sobre la importancia del trabajo colaborativo.....	73
4.1.2	Mejor organización de los equipos de trabajo.	75
4.1.3	Mayor importancia del tablero de tareas	75
4.1.4	Mayor usabilidad del modelo para los usuarios (Docentes).	76
4.2	Generación de thinklets (Childthinklets)	81
4.3	Descripción de ChildProgramming - C	83
4.3.1	Modelo conceptual de ChildProgramming.....	83
4.3.2	Actores del Modelo ChildProgramming	86
4.3.3	Descripción de las fases y actividades del modelo.....	87
4.3.4	Prácticas propuestas para childprogramming-C.....	90
4.3.5	Modelo ChildProgramming desde la ingeniería de la colaboración	92
5.	Capítulo 5. evaluación del modelo	103
5.1	Estudios de caso confirmatorios	103
5.1.1	Contexto de la investigación	104
5.1.2	Indicadores de evaluación seleccionados	105
5.1.3	Instrumentos de evaluación utilizados.....	109
5.1.4	Caso de estudio 1	113
5.1.5	Estudio de caso 2	120
6.	Capítulo 6. Conclusiones y trabajos futuros	129
6.1	Trabajos futuro.....	132

6.2 Publicaciones y reconocimientos	133
Referencias.....	135

Lista de tablas

Tabla 1. Patrones definidos en la Ingeniería de la Colaboración [30].....	24
Tabla 2. Subpatrones del patrón generación [30].....	25
Tabla 3. Subpatrones del patrón reducción [30].....	25
Tabla 4. Subpatrones del patrón clarificación [30].....	25
Tabla 5. Subpatrones del patrón organización [30]	26
Tabla 6. Subpatrones del patrón evaluación [30]	26
Tabla 7. Subpatrones del patrón construcción de consenso [30]	26
Tabla 8. Documentación de un thinklet [30]	27
Tabla 9. Comparación de modelos relacionados.	38
Tabla 10. Valoración preliminar de los equipos de trabajo –Primer estudio de Caso	46
Tabla 11. Valoración de los equipos de trabajo –Estudio de Caso con dinámicas y mecánicas de juego.....	47
Tabla 12. Descripción del proceso objeto de estudio.	53
Tabla 13. Descripción de actividades generales ChildProgrammig [14].	55
Tabla 14. Descomposición de actividades de ChildProgrammig [14].	57
Tabla 15. Análisis de criterios para definir si una actividad requiere trabajo colaborativo para su ejecución.....	59
Tabla 16. Criterios seleccionados para definir si una actividad llevada a cabo por niños requiere trabajo colaborativo para su ejecución.....	59
Tabla 17. Subactividades de ChildProgramming que requiere trabajo colaborativo para su ejecución	61
Tabla 18. Patrones definidos en la Ingeniería de la Colaboración [30].....	63
Tabla 19. Asociación de patrones a las subactividades [30].	63
Tabla 20. Descripción de subactividad 1.1 3 Definir las reglas de juego.	64
Tabla 21. Descripción de subactividad 1.3.2: Identificación de tareas.....	65
Tabla 22. Descripción de subactividad 1.3.3: Identificación y priorización de tareas	65
Tabla 23. Descripción de subactividad 2.1.1 Definición de la estrategia.	66
Tabla 24. Descripción de subactividad 2.1.1 Definición de tareas.....	66
Tabla 25. subactividad 2.2 Aplicar la Estrategia.....	67

Tabla 26. Descripción de subactividad 2.3 Revisar la estrategia.....	68
Tabla 27. Descripción de Subactividad 2.4 Analizar la estrategia	69
Tabla 28. Técnicas colaborativas de discusión	78
Tabla 29. Técnicas de enseñanza recíproca entre compañeros	79
Tabla 30. Técnicas colaborativas para resolver problemas.....	79
Tabla 31. Técnicas colaborativas de organización gráfica	80
Tabla 32. Childthinklet Analysis in Pairs.....	82
Tabla 33. Childthinklet Divide et impera.....	82
Tabla 34. Actores del Modelo ChildProgramming –C.....	87
Tabla 35. Descripción del modelo Childprogramming-C	88
Tabla 36. Descripción del modelo Childprogramming-C	91
Tabla 37. Relación de thinklets subactividad 2.4.....	95
Tabla 38. Agenda detallada de actividades colaborativas.....	97
Tabla 39. Estudios de caso realizados.....	104
Tabla 40. Diseño del estudio de caso 1	113
Tabla 41. Descripción Actividad No. 1 tomada y adaptada de [14].....	115
Tabla 42. Descripción Actividad No. 2	118
Tabla 43. Resultados indicadores del estudio de caso 1.....	119
Tabla 44. Resultados de indicadores del estudio de caso 1 – Docente e investigador...	119
Tabla 45. Promedio de valoración de indicadores del estudio de caso 1	120
Tabla 46. Diseño del estudio de caso confirmatorio.....	121
Tabla 47. Diseño del estudio de caso confirmatorio.....	123
Tabla 48. Resultados del estudio de caso 2.....	124
Tabla 49. Resultados del estudio de caso 2 – docente e investigador	125
Tabla 50. Promedio de valoración de indicadores del estudio de caso 2	126
Tabla 51. Resultados finales de criterios evaluados.	127

Lista de figuras

Figura 1. Patrón reducción y thinklets	28
Figura 2. Patrón generación y thinklets	28
Figura 3. Patrón clarificación y thinklets	29
Figura 4. Patrón organización y thinklets	29
Figura 5. Patrón evaluación y thinklets	29
Figura 6. Patrón construcción de consenso y thinklets.....	30
Figura 7. Fragmento de mapa de selección	30
Figura 8. Dimensiones de Childprogramming [13]	31
Figura 9. Ciclo de vida del proceso de ChildProgramming inicial [44]	32
Figura 10. Ciclo de vida del proceso de ChildProgramming gamificado (Childprogramming-G) [46]	32
Figura 11. Estructura fundamental del método CSACE	34
Figura 12. Metodología para el Desarrollo de proceso Colaborativos [52].	35
Figura 13. Primera fase del método CSACE [7]	43
Figura 14. Evidencias tofográficas estudio de caso 1 – actividad 1	45
Figura 15. Evidencias tofográficas estudio de caso 1 – Actividad 2	45
Figura 16. Análisis primer estudio de caso preliminar	46
Figura 17. Evidencias fotográficas estudio de caso 2 – Taskboard y equipos	47
Figura 18. Análisis segundo estudio de caso	48
Figura 19. Integración del método CSACE y la Metodología para el Desarrollo de procesos Colaborativos	49
Figura 20. Ciclo de vida del proceso ChildProgramming-G	51
Figura 21. Modelo Conceptual ChildProgramming - G.	52
Figura 22. Patrón liderar	70
Figura 23. Patrón propagación de conocimiento	70
Figura 24. Mejoras identificadas al modelo Childprogramming	73
Figura 25. Estrategias para sensibilizar a los niños sobre la importancia de la colaboración	74
Figura 26. Proceso de sensibilización propuesto	75

Figura 27. Técnicas de colaboración para apoyar el proceso Childprogrammig –C basada de [87]	77
Figura 28. Propuesta de Incorporación de técnicas colaborativas a la Metodología para el Desarrollo de proceso Colaborativos	80
Figura 29. Modelo conceptual del Chiildprogrammig-C [14][45]	83
Figura 30 Modelo ChildProgramming –C. fuente propia.....	85
Figura 31. Modelo de facilitación de procesos – fase Prejuego.....	98
Figura 32. Modelo de facilitación de procesos Actividades colaborativas Childprogramming-C.....	100
Figura 33. Fase 3 del método CSACE	103
Figura 34. Estudios de caso definidos para la evaluación.....	103
Figura 35. Indicadores para valorar el nivel de colaboración.....	106
Figura 36. Folleto para actividad del poliedro.....	115
Figura 37. Evidencia fotográfica - Actividad 1	116
Figura 38. Guía para retos de programación.	118
Figura 39. Evidencia fotográfica - Actividad 2	118
Figura 40. Gráfica Indicadores de colaboración - estudio de caso 1.	120
Figura 41: Actividad de sensibilización colaborativa 1 – Escribir juntos	122
Figura 42. Actividad de sensibilización colaborativa 2 – Competencia tarros.....	122
Figura 43. Actividad de sensibilización colaborativa 3 – Competencia con pitillos	122
Figura 44. Actividad de sensibilización colaborativa 4- competencia con bombas	122
Figura 45. Actividad de sensibilización colaborativa 5 organización de números	123
Figura 46. Actividad de sensibilización colaborativa 6 – mi guía	123
Figura 47. Actividad de sensibilización 1 - Poliedro	124
Figura 48. Desarrollo de aplicaciones software.....	124

1.Capítulo 1

1.1 Introducción

“La colaboración es un fenómeno crítico en la vida de las organizaciones” [1], las decisiones importantes se toman por un grupo de personas y la solución de problemas complejos se lleva a cabo por medio de aportes o experiencias de varios individuos que contribuyen a lograr mejores resultados [2]. Por esta razón, “la sociedad siempre ha buscado la formación de grupos de trabajo que actúan como una sola unidad, con el propósito de resolver problemas o ejecutar tareas específicas” [3]. La importancia de la colaboración no ha sido ajena al proceso de desarrollo de software, es así como el manifiesto de las metodologías ágiles de desarrollo de software expone la colaboración como un factor humano clave que debe existir entre los equipos ágiles [4].

Aunque al colaborar los individuos pueden lograr mejores resultados de los que se podían individualmente, el logro de una eficaz colaboración en equipo sigue siendo un reto [1]. Por lo anterior, es fundamental proponer soluciones desde el enfoque de la Ingeniería de la Colaboración que contribuyan al alcance de un proceso colaborativo y más en el ámbito del aprendizaje del desarrollo del software llevado a cabo por niños, donde se puede incentivar la resolución de problemas complejos y el desarrollo del pensamiento computacional mientras colaboran y aprenden, aprovechando las ventajas que ofrece el trabajo colaborativo, entre las cuales se tiene [5]:

- Promover el logro de objetivos más ricos en contenido, dado que reúne propuestas y soluciones de varias personas del grupo.
- Valorar el conocimiento y opiniones de los demás miembros del grupo.
- Aumentar la motivación por el trabajo individual y grupal.
- Aumentar las habilidades sociales, de interacción y comunicación.
- Disminuir el temor a la realimentación.

De este modo, es primordial considerar que al integrar aspectos de trabajo colaborativo a un proceso determinado, el objetivo no es solo la mejora de la comunicación, sino una mayor participación y compromiso entre los integrantes de un grupo que trabajan en torno a una actividad común, lo que conlleva a mejorar la calidad del producto elaborado [6]. En ChildProgramming dicho producto es el software desarrollado por los niños en el proceso orientado por los docentes y un investigador. Esta investigación realiza un valioso aporte a los aspectos relacionados con la Ingeniería de la Colaboración, la Ingeniería del Software, la Mejora de procesos de Software y el trabajo Colaborativo.

El presente trabajo propone un conjunto de mejoras al actual modelo ChildProgrammig, relacionadas con el desarrollo y evaluación de procesos colaborativos, dicha investigación está orientado mediante el método CSACE (Case Study based Analysis in Collaboration Engineering) [7], y la Metodología para el Desarrollo de Procesos colaborativos [8], basado en estudios de caso para establecer las necesidades de colaboración desde las interacciones del equipo, así como la evaluación de la efectividad del proceso colaborativo resultante. Para el desarrollo de los casos de estudio se utilizó la herramienta SCRATCH y participaron niños entre los 10 y 12 años de edad.

El presente documento de trabajo de grado se divide en los capítulos que se describen a continuación:

Capítulo 1: este capítulo incluye el planteamiento del problema que dio origen al proyecto, los objetivos propuestos, la hipótesis y los aportes investigativos del presente trabajo.

Capítulo 2: este capítulo presenta el marco conceptual que soporta esta investigación y los trabajos relacionados.

Capítulo 3: en el capítulo tres se presentan los estudios de caso que permitieron plantear las mejoras del modelo ChildProgramming desde la Ingeniería de la Colaboración y el Aprendizaje Colaborativo, así como la identificación de las actividades colaborativas del modelo y la propuesta de dos patrones de colaboración.

Capítulo 4: presenta la descripción y componentes del modelo ChildProgramming–C además de los principios colaborativos que lo soportan.

Capítulo 5: describe el diseño, la ejecución y los resultados del estudio de caso realizado con el fin de comprobar la hipótesis.

Capítulo 6: este capítulo expone las conclusiones y trabajos futuros derivados del presente estudio, la divulgación que se ha realizado para la investigación y finalmente los referentes de la investigación.

1.2 Descripción del problema

Wing[9] ha definido el pensamiento computacional como una aproximación para resolver problemas, diseñar soluciones y entender el comportamiento humano. La autora se ha convertido en un referente para quienes se interesan en el estudio del pensamiento computacional [10], el cual lo define, como una habilidad que debe desarrollar todo el mundo y no únicamente los científicos de la computación. Wing [11] hace una referencia explícita al papel de la educación en este desarrollo, cuando invita a los docentes a incluir el análisis como parte de su quehacer en el aula, argumentando que para desarrollar el pensamiento computacional en sus propias palabras, se deben tener en cuenta ciertas habilidades para: resolver problemas, diseñar y desarrollar sistemas, entender el comportamiento humano utilizando conceptos informáticos y una serie de herramientas mentales que reflejan la amplitud de las aplicaciones informáticas [11].

El pensamiento computacional es una competencia universal que debe añadirse a la capacidad analítica de cada niño como un componente fundamental de su aprendizaje escolar [12]. Particularmente, en el desarrollo del pensamiento computacional existe la preocupación sobre el cómo capacitar a los niños y hacer que ellos aprendan a construir adecuadamente aplicaciones de la vida real, utilizando un lenguaje de programación [12]. Por ello para enseñar y generar habilidades de computación en niños, han surgido una gran variedad de experiencias y herramientas software, como por ejemplo: Scratch, Kodu, Alice, Squeak, Toontalk, Kidpad, LOGO, entre otras [13].

Del mismo modo, para realizar una enseñanza de la programación y facilitar la práctica del desarrollo de software en niños, el grupo de Investigación y Desarrollo de Ingeniería del Software de la Universidad del Cauca (IDIS), ha propuesto un modelo llamado ChildProgramming [14], el cual plantea estrategias de trabajo colaborativo y aprovecha los enfoques ágiles para el desarrollo de software basado en paradigmas modernos, así como

prácticas cognitivas basando su ciclo de vida de la metodología de desarrollo de software Scrum [13].

ChildProgramming se compone de tres dimensiones [14]: la dimensión cognitiva, considerada como el esfuerzo que realizará un niño para comprender, analizar y apropiarse situaciones presentes en las tareas definidas en el modelo. La dimensión ágil, basada en la promulgación de los valores del manifiesto de las metodologías ágiles de desarrollo de software; y la dimensión colaborativa, que intenta incrementar la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos mediante el desarrollo de aplicaciones software [14]. El modelo define tres fases: pre-juego, juego y post-juego, y ha mejorado en aspectos de gamificación hasta obtener la versión Childprogramming-G.

Sin embargo, al realizar un análisis desde la perspectiva de la colaboración de las actividades del proceso y las prácticas colaborativas propuestas por ChildProgramming y su versión gamificada, se encontraron debilidades en su estructuración dado que no cuenta con una formalización desde la Ingeniería de la Colaboración y a pesar de identificarse patrones y Thinklets de colaboración, estos se presentan desligados de las actividades propuestas por el modelo, y no muestra claridad de cómo ejecutar una actividad colaborativa dentro de ChildProgramming; por lo tanto, se evidencia que hay carencia de estructuración de las actividades desde el enfoque de Ingeniería de Colaboración donde “se diseñan procesos repetitivos colaborativos, los cuales se pueden transferir a grupos, usando técnicas y tecnología de colaboración”[1] y pretende guiar la ejecución de tareas colaborativas, de forma predecible y exitosa entre los miembros de un equipo de trabajo [1]. Del mismo modo no hay una orientación sobre qué técnicas colaborativas o recursos de colaboración utilizar en un proceso de desarrollo de software llevado a cabo por niños.

Así mismo, y considerando la importancia de “la Interdependencia positiva, la igual participación y la responsabilidad individual, como elementos imprescindibles que se deben presentar en la colaboración para que el aprendizaje tenga lugar de manera efectiva” [15], preliminar al planteamiento del presente proyecto, se realizaron dos estudios de caso exploratorios con estudiantes de una institución educativa, donde se analizó cómo los niños colaboraban al desarrollar las actividades propuestas en el modelo ChildProgramming. De esta experiencia se logró observar que la mayoría de los niños se muestran motivados al

desarrollar el software pero, a pesar de explicar previamente las prácticas colaborativas descritas en el modelo, algunos grupos no mostraron interés por las actividades colaborativas relacionadas con la selección del nombre del equipo, la asignación de roles y responsabilidades, la elección del líder, la definición y ejecución de estrategia para resolver el problema; por lo tanto, varios equipos presentaron improvisaciones, discusiones, y se evidenció la poca participación de algunos integrantes al verse opacados por otros compañeros.

Analizando la situación presentada en el desarrollo de los estudios de caso, se pudo apreciar que la interdependencia positiva, la igual participación y la responsabilidad individual se ven afectadas al no tener un compromiso de todos los participantes, y no definir una estrategia clara para solucionar el problema y olvidar la asignación de roles y responsabilidades. Así mismo, al evaluar los productos entregados y el cumplimiento del tiempo asignado para la ejecución de la actividad, solo dos de seis equipos lograron el desarrollo del producto en el tiempo esperado y cumpliendo los requerimientos planteados.

La estrategia de trabajar y aprender en conjunto ha sido bastante usada y difundida [6], *“sin embargo, trabajar realmente de forma colaborativa no es fácil, no basta con disponer a un grupo de personas entorno a una actividad y esperar a que el aprendizaje llegue”*[6], *“es necesario estructurar actividades para alcanzar ese objetivo”* [16]. En el marco de esta situación y considerando las dificultades evidenciadas en los estudios de caso, surgen las siguientes preguntas de investigación:

¿Cómo diseñar y estructurar un modelo colaborativo para el desarrollo de software llevado a cabo por niños a partir del modelo ChildProgramming?

¿Estructurar las actividades del modelo ChildProgramming desde el enfoque la Ingeniería de la Colaboración, incrementa la colaboración en un proceso de desarrollo de software llevado a cabo por niños?

El poder dar respuesta a estas preguntas a través de la construcción de un modelo de desarrollo de software implementado por niños desde la perspectiva de la Ingeniería de la Colaboración, permitirá diseñar y estructurar actividades desde el enfoque colaborativo con el objetivo de guiar la ejecución de las tareas en un entorno donde los profesionales que

coordinan el desarrollo de software generalmente son docentes y no necesariamente son ingenieros y/o expertos en colaboración, además se busca que los niños adquieran conocimientos, habilidades sociales y demás beneficios del trabajo colaborativo mediante el desarrollo de software.

1.3 Objetivos

Objetivo general

- Definir un modelo colaborativo para soportar el desarrollo de software llevado a cabo por niños, basado en el modelo ChildProgramming.

Objetivos específicos

- Identificar las actividades del modelo ChildProgramming que requieren trabajo colaborativo para su ejecución.
- Rediseñar el modelo ChildProgramming desde el enfoque de la Ingeniería de la Colaboración.
- Evaluar la colaboración generada durante el proceso de desarrollo de software por parte de los niños, al utilizar el modelo ChildProgramming-C.

1.4 Hipótesis

De acuerdo a la descripción del problema y el estado del arte consultado, a continuación, se define la hipótesis de investigación y la hipótesis nula:

Hipótesis de investigación: plantear las actividades del proceso de ChildProgramming desde el enfoque de la Ingeniería de la Colaboración, incrementa la colaboración durante el desarrollo de software llevado a cabo por niños.

Hipótesis nula: plantear las actividades del proceso de ChildProgramming desde el enfoque de la Ingeniería de la Colaboración, no incrementa la colaboración durante el desarrollo de software llevado a cabo por niños.

De esta forma, se espera que al finalizar el trabajo de investigación mediante la aplicación de estudios de caso la hipótesis sea satisfecha.

1.5 Aporte investigativo

El presente trabajo se encuentra enmarcado dentro de las líneas de la Ingeniería del Software, la mejora de procesos de software y la Ingeniería de la Colaboración que desarrolla el grupo IDIS [17]. A continuación, se exponen los aportes esperados del presente trabajo de investigación, gran parte de ellos reflejados en tres artículos publicados:

- Conjunto de criterios para evaluar actividades llevadas a cabo por niños, que requieren trabajo colaborativo durante un proceso de desarrollo de software.
- Propuesta de dos patrones de colaboración identificados en el proceso de desarrollo de software llevado a cabo por niños y aplicables a otros contextos.
- Propuesta de dos thinklets de colaboración identificados en las actividades realizadas por los niños.
- Propuesta de estrategias para sensibilizar a los niños sobre la importancia del trabajo colaborativo antes de iniciar procesos de desarrollo de software.
- Un modelo desde el enfoque de la Ingeniería de la Colaboración para el desarrollo de software por parte de los niños a partir del modelo ChildProgramming. Los procesos que conformen el modelo propuesto serán diseñados sistemáticamente aplicando el enfoque de la Ingeniería de la Colaboración, para el caso particular de procesos realizados por niños.
- Conjunto de prácticas de apoyo del modelo Childprogramming –C.
- Monografía del trabajo de grado. Corresponde al presente documento, donde se describe el proceso seguido en el desarrollo del proyecto, los aportes más sobresalientes, las conclusiones y las recomendaciones para el desarrollo de futuras investigaciones en el área.

2. Base conceptual

En esta sección se describen algunos conceptos necesarios para la contextualización del presente trabajo. El primero de ellos es la Ingeniería de la Colaboración, adscrita a la Ingeniería del Software como área que promueve una filosofía organizacional del proceso de desarrollo y que fomenta el trabajo en equipo, a partir del reconocimiento de las habilidades y aptitudes individuales del grupo de trabajo [18], posteriormente se presentan los principios del trabajo colaborativo, los patrones y thinklets de colaboración relevantes en este trabajo, también se presenta el modelo Childprogramming y su versión gamificada. En última instancia se realiza una descripción de las metodologías utilizadas para el desarrollo del proyecto y los trabajos relacionados.

2.1 Ingeniería de la Colaboración

A partir de la necesidad de diseñar, ejecutar y estructurar procesos colaborativos al interior de diferentes grupos, surge la Ingeniería de la Colaboración [19], la cual es “un enfoque sistemático para el diseño de procesos de colaboración repetibles, que se pueden utilizar para aumentar la eficiencia y la eficacia humana en las organizaciones” [20]. La Ingeniería de la Colaboración es un acercamiento al diseño de procesos colaborativos reutilizables, por lo tanto, los procesos colaborativos necesitan ser explícitamente diseñados, estructurados y manejados, para poder ser transferidos a grupos, usando técnicas y tecnologías de colaboración [1].

El campo de la Ingeniería de la Colaboración ha surgido como un punto focal para la investigación sobre el diseño y la implementación de procesos de colaboración que son recurrentes en la naturaleza y ejecutados por profesionales en las organizaciones, en lugar de profesionales de colaboración [21]. Al colaborar, los individuos logran mayores resultados de los que podrían individualmente, sin embargo, alcanzar una eficaz colaboración en equipo sigue siendo un reto [1].

2.2 Trabajo colaborativo

El trabajo colaborativo ofrece una opción para superar limitaciones del aprendizaje tradicional, bajo este esquema debe tenerse en cuenta la presencia de los diferentes actores y su coordinación, toda vez que las situaciones obligan a tener colaboración, comunicación e intercambio de información [22], al integrar aspectos de trabajo colaborativo a un proceso determinado el objetivo no es sólo la mejora de la comunicación, sino también lograr mayor participación y compromiso, entre los integrantes de un grupo que trabajan en torno a una actividad común, lo que conlleva a una mejor calidad del producto elaborado [23].

Johnson & Johnson han desarrollado varios análisis estadísticos con estudiantes de diferentes edades, niveles educativos y sociales, y han demostrado el efecto positivo que el aprendizaje colaborativo ha tenido en su éxito académico y en sus logros sociales [24], de este modo definen el trabajo colaborativo como [24]: "el conjunto de métodos de instrucción o entrenamiento para el uso en grupos pequeños, así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del grupo es responsable tanto de su aprendizaje como del de los restantes miembros del grupo".

Según Turban [3], el trabajo grupal cuenta con una serie de ventajas sobre el trabajo individual entre ellas se tiene: un grupo entiende mejor un problema que una sola persona, existe una responsabilidad compartida, facilita la detección de errores, un grupo presenta un mayor conocimiento que una sola persona, lo cual ofrece mejores alternativas para la resolución de problemas, se presenta efectividad y calidad de producción, la efectividad y calidad de la producción de un grupo es mayor que la suma de lo que pueda producir cada miembro en forma individual, dado que se potencia este conocimiento individual con el grupal obteniendo mejores resultados; de este modo la colaboración es uno de los principales componentes del modelo Childprogramming que busca ser mejorado.

2.3 Principios básicos del trabajo colaborativo

En [25] se proponen algunos principios básicos para que un grupo pueda desempeñarse bajo el enfoque del trabajo colaborativo:

- Con el objetivo de alcanzar metas comunes, es necesario que todos los integrantes brinden apoyo y ayuda en el cumplimiento de las tareas y el trabajo.
- Debe propenderse por el respeto de los diferentes puntos de vista entre los integrantes.
- La formación de grupos es intencional y basada en la heterogeneidad. Los grupos se constituyen con base a las diferencias de habilidades, así como de características de personalidad y género de los integrantes.
- Es necesario que cada uno de los integrantes contribuya de un modo particular al logro de las metas del grupo. Nadie gana méritos “a costa” del trabajo de los demás.
- La responsabilidad debe ser igual para cada integrante del equipo.
- Son indispensables las habilidades interpersonales tales como: la confianza mutua, la comunicación clara y sin ambigüedades, el apoyo mutuo y la resolución constructiva de conflictos, durante el trabajo en equipo.

2.4 Aprendizaje colaborativo

El Aprendizaje Colaborativo se define [24] como: “un conjunto de métodos de instrucción para la aplicación en grupos pequeños, de entrenamiento y desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del grupo es responsable tanto de su aprendizaje como del de los restantes miembros del grupo”. Para lo cual se basan en tres teorías: La interdependencia social, la perspectiva evolutiva cognitiva y la perspectiva conductista del aprendizaje.

El énfasis del Aprendizaje Colaborativo está en el pensamiento crítico, la resolución de problemas y la construcción del conocimiento [26]. Es, ante todo, un sistema de interacciones cuidadosamente diseñado, que organiza e induce la influencia recíproca entre los integrantes de un equipo. También es un proceso en el que se va desarrollando gradualmente, entre los integrantes de dicho equipo, el concepto de ser “mutuamente responsables del aprendizaje de cada y uno, y de los demás”[27].

El aprendizaje colaborativo es una teoría que data de años atrás y que vuelve a cobrar vigencia desde la aparición de los modelos educativos en línea y con los estudiantes actuales que son nativos digitales [28]. Muchos autores que defienden esta teoría, argumentan que el intercambio de forma activa, el debate y la negociación de ideas entre pares, aumenta el interés en el aprendizaje [29].

2.5 Patrones y subpatrones de colaboración

Los *patrones de colaboración* se definen en términos del “movimiento del grupo desde su estado inicial hasta su estado final” [36]. Cada patrón asocia subpatrones que se pueden relacionar con las actividades. La Tabla 1 presenta los *patrones de colaboración* y su respectiva descripción.

PATRONES DE COLABORACIÓN	DESCRIPCIÓN
Patrón Generación	Es un patrón a partir del cual el grupo crea contenido. Consiste en pasar de tener pocos a muchos conceptos que son compartidos por el grupo.
Patrón Reducción	El objetivo de este patrón es mantener sólo la información que cumple con un determinado criterio o criterios. Consiste en pasar de tener muchos conceptos a unos pocos que el grupo considere que requieren mayor atención.
Patrón Clarificación	El objetivo de este patrón es lograr el entendimiento común de conceptos manejados por el grupo, consiste en pasar de tener un menor a un mayor conocimiento compartido de los conceptos, las palabras y frases usadas para expresarlos.
Patrón Organización	Consiste en pasar de tener un menor a un mayor conocimiento de las relaciones entre los conceptos que el grupo esté considerando.
Patrón Evaluación	Consiste en pasar de un menor a un mayor conocimiento del valor relativo de los conceptos bajo consideración. Este patrón tiene como efectos apoyar a la toma de decisiones y a la comunicación del grupo.
Patrón Construcción de consenso	Moverse de tener pocos a muchos miembros del grupo quienes estarán dispuestos a comprometerse para un objetivo.

Tabla 1. Patrones definidos en la Ingeniería de la Colaboración [30]

2.5.1 Subpatrones de colaboración

En las tablas 2 – 7, se describen los subpatrones de cada uno de los siguientes patrones: generación, reducción, clarificación, organización y evaluación [31].

PATRONES DE COLABORACIÓN		
Generación	Subpatrones asociados:	<p>Reunir: coleccionar y compartir conceptos entre los miembros del grupo.</p> <p>Crear: producir y compartir nuevas ideas que no fueron previamente conocidas por los miembros del grupo.</p> <p>Elaborar: adicionar detalles a los conceptos que ya fueron compartidos por los miembros del grupo.</p> <ul style="list-style-type: none"> • Descomponer: caracterizar un concepto en términos de sus componentes o subcomponentes. • Expandir: adicionar detalles para una explicación más completa o para describir un concepto.

Tabla 2. Subpatrones del patrón generación [30]

PATRONES DE COLABORACIÓN		
Patrón Reducción	Subpatrones asociados:	<p>Seleccionar: escoger un subconjunto de conceptos existentes.</p> <p>Abstractar: derivar conceptos más generales desde instancias específicas, en el conjunto existente.</p> <p>Resumir: capturar la esencia de los conceptos, sin eliminar conceptos únicos.</p>

Tabla 3. Subpatrones del patrón reducción [30]

PATRONES DE COLABORACIÓN		
Patrón Clarificación	Subpatrón asociado:	Describir: proponer explicaciones y descripciones alternativas de un concepto.

Tabla 4. Subpatrones del patrón clarificación [30]

PATRONES DE COLABORACIÓN		
Patrón Organización	Subpatrones asociados:	<p>Clasificar: ordenar conceptos en categorías.</p> <p>Estructurar: crear arreglos espaciales entre conceptos para representar sus relaciones conceptuales.</p>

Tabla 5. Subpatrones del patrón organización [30]

PATRONES DE COLABORACIÓN		
Patrón Evaluación	Subpatrones asociados:	<p>Votar: evaluar la opinión del grupo, respecto a los conceptos.</p> <p>Categorizar: identificar un orden de preferencia entre los conceptos.</p> <p>Valorar: Especificar y detallar en el valor de los conceptos.</p>

Tabla 6. Subpatrones del patrón evaluación [30]

PATRONES DE COLABORACIÓN		
Patrón Construcción de consenso	Subpatrones asociados:	<p>Medir: Evaluar el grado en que los participantes están dispuestos a comprometerse con una propuesta.</p> <p>Diagnosticar: Buscar la comprensión de las causas de desacuerdos.</p> <p>Apoyar: Tratar de convencer a otros a aceptar y adoptar una posición.</p> <p>Resolver: Buscar la forma de superar las causas de desacuerdos.</p>

Tabla 7. Subpatrones del patrón construcción de consenso [30]

2.6 Thinklets de colaboración

Los thinklets son “técnicas de facilitación repetibles, transferibles y predecibles para asistir a un grupo en alcanzar su objetivo acordado” [32]. La descripción de los patrones de

colaboración no presenta una forma detallada para guiar a un equipo de forma precisa a través de la ejecución de un proceso y debe ser posible seleccionar bloques de construcción existentes y unirlos para especificar cómo un determinado patrón debe realizarse [33]. Un thinklet “constituye la unidad más pequeña del capital intelectual necesario para crear un patrón de colaboración repetible y predecible entre las personas que trabajan hacia un objetivo” [34]. Un thinklet debe documentarse con la siguiente información [35][36]:

Información para documentar un thinklets
Un nombre metafórico o representativo relacionado con los patrones que crea el thinklet.
Criterios para decidir cuándo escoger o no el thinklet.
Información general del thinklet, como entradas y salidas. Las entradas son variables que deben ser instanciadas al tiempo que el thinklet es usado y las salidas son los entregables que deben generarse
Modo de uso del thinklet (pasos que conforman el thinklet que muestran cómo usar el thinklet), una historia exitosa que ayuda a clarificar las circunstancias bajo las cuales el Thinklet es útil y una explicación del nombre, la cual hace más fácil recordar el nombre del thinklet.

Tabla 8. Documentación de un thinklet [30]

Una de las grandes ventajas de los thinklets es que los diseñadores de procesos colaborativos pueden emplearlos para escoger soluciones conocidas y no invertir esfuerzos en inventar y probar nuevas. Esto puede reducir tanto el esfuerzo como el riesgo, durante el desarrollo de procesos colaborativos [34].

Una de las grandes ventajas de los thinklets es que al diseñar procesos colaborativos se pueden emplear soluciones conocidas y no invertir esfuerzos en inventar y probar nuevas [37], por lo tanto, el enfoque de trabajo cambia de inventar y probar soluciones, a escoger soluciones conocidas y probadas, lo que puede reducir tanto el esfuerzo como el riesgo de desarrollar e implementar procesos grupales [38]. Por ejemplo, un thinklet en el que un

grupo vota sobre un conjunto de ideas puede crear un modelo de evaluación y dará lugar a una clasificación del conjunto de ideas basadas en el criterio de votación. Un thinklet en el que el grupo selecciona una serie de ideas que debe abordarse en una discusión de grupo posterior, crea un patrón de evaluación, así como un patrón de reducción [39].

A continuación se presentan los thinklet relacionados a cada patrón de colaboración de acuerdo a [40][41], y en el anexo A se hace una descripción más amplia de cada uno de ellos:

En la figura 1 se presenta el thinklet BronWagon que hace parte del patrón reducción.

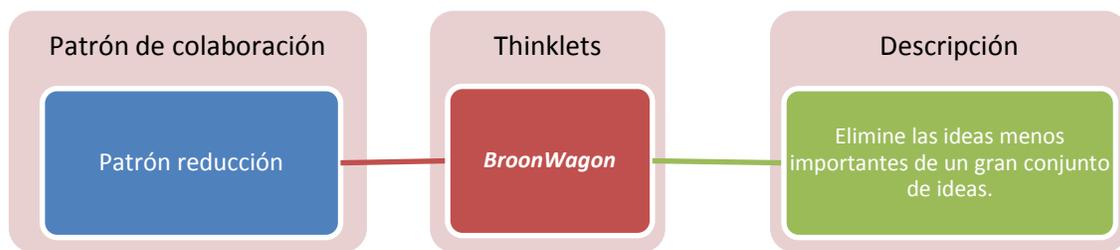


Figura 1. Patrón reducción y thinklets

En la siguiente figura se presenta los thinklets Onepage, LeafHopper, FreeBrainstorm que hace parte del patrón generación.

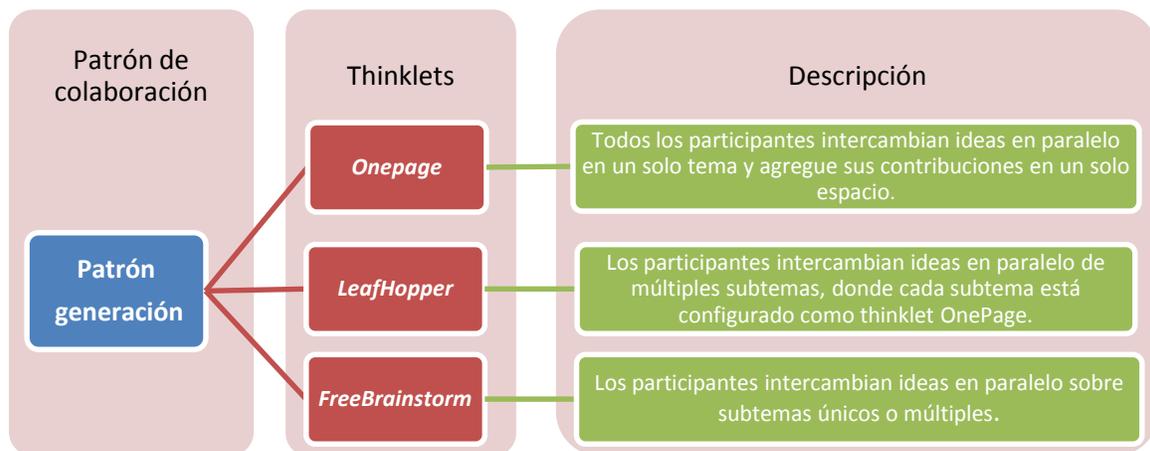


Figura 2. Patrón generación y thinklets

En la siguiente figura se presenta el thinklet Pin the tail on the donkey que hace parte del patrón clarificación.



Figura 3. Patrón clarificación y thinklets

En la figura 4 se presentan los thinklets Concentration y RichRelations que están asociados al patrón organización.

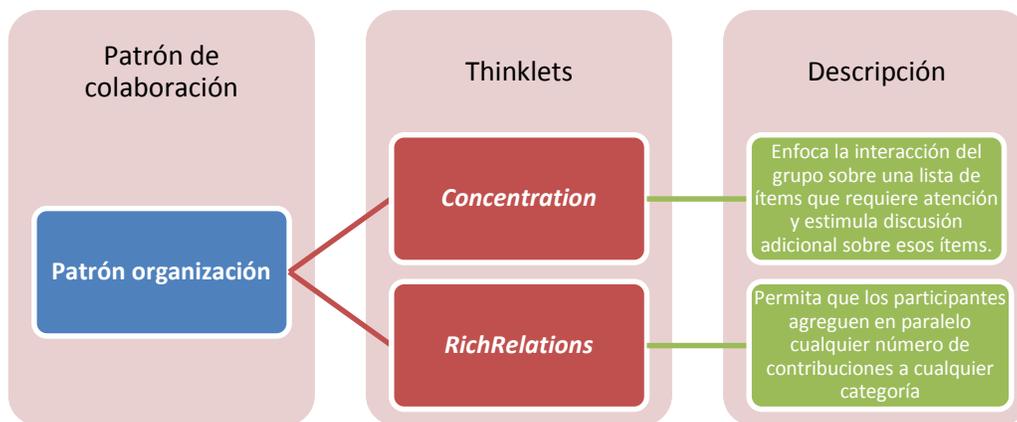


Figura 4. Patrón organización y thinklets

En la figura 5 se presentan el thinklet StrawPoll que están asociados al patrón evaluación.

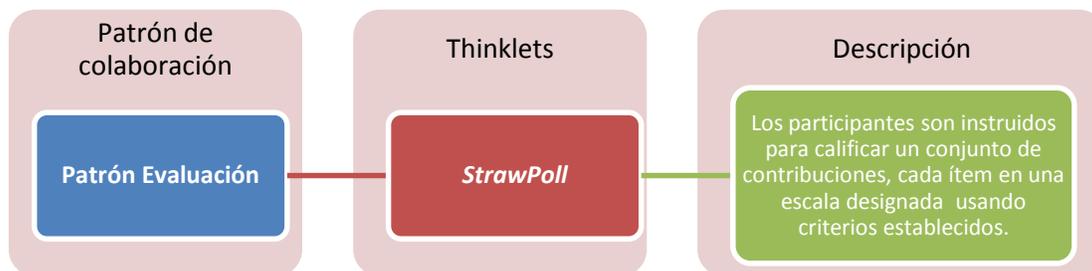


Figura 5. Patrón evaluación y thinklets

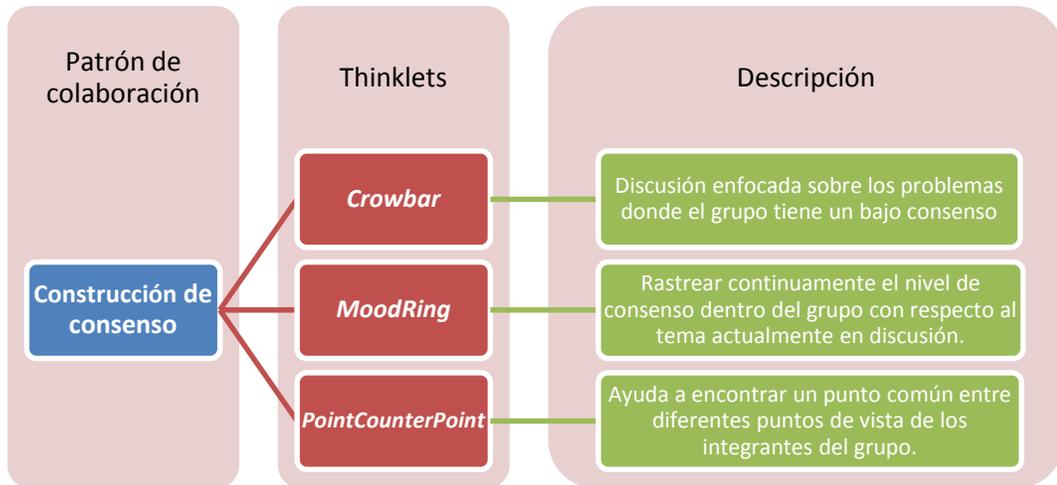


Figura 6. Patrón construcción de consenso y thinklets

2.6.1 Mapa de selección de thinklets

El mapa de selección de thinklets indica el tipo de relación entre los thinklets. Este aspecto es útil ya que permite identificar los thinklets que tienen una excelente relación, lo cual da mayor certeza en que se obtendrán los resultados adecuados con su ejecución, además sirve para descartar las relaciones que se han establecido como imposibles [8].

Los tipos de relación se representan mediante colores: el verde indica si la relación es excelente; naranja si la relación es posible pero difícil y rojo si la relación es imposible. La Figura 7 presenta un fragmento del mapa de selección.

La relación de estos thinklets es:	FreeBrainstorm	OnePage	Comparative Brainstorm	LeafHopper	DealersChoice	PlusMinusInteresting
Excelente						
Possible pero difícil						
Imposible						
Punto de partida?						
FreeBrainstorm						
OnePage						
Comparative Brainstorm						
LeafHopper						
DealersChoice						
PlusMinusInteresting						

Figura 7. Fragmento de mapa de selección

2.7 Modelo ChildProgramming

Algunas discusiones en el área de la programación de software en los niños se han centrado en diferentes temas como: identificar la forma de crear lenguajes para programación con niños, o si los niños pueden aprender temas particulares en la programación de software[42]; sin embargo, otras cuestiones se han dirigido en encontrar una estrategia para que los pequeños puedan crear sus propios programas, este es el caso del modelo ChildProgramming, el cual nace como una idea de proyecto en el grupo de Investigación y Desarrollo de Ingeniería del Software (IDIS) de la Universidad del Cauca, y formaliza un modelo para soportar el desarrollo de software orientado a los niños, a partir de un conjunto de experiencias prácticas fundamentadas en estrategias de aprendizaje colaborativo y prácticas ágiles de desarrollo de software [8]. ChildProgramming se compone de tres dimensiones [8], como se muestra en la figura 8 y se explica a continuación.



Figura 8. Dimensiones de Childprogramming [13]

La dimensión cognitiva, considerada como el esfuerzo que realizará un niño por comprender, analizar y apropiar situaciones presentes en las tareas definidas por el modelo. La dimensión ágil: basada en la promulgación de los valores del manifiesto de las metodologías ágiles de desarrollo de Software, esta dimensión aporta a ChildProgramming una forma acorde de trabajo para alcanzar los objetivos donde se evidencie el trabajo en equipo que permita a los integrantes del mismo permanecer juntos a lo largo de la actividad.

Al tener en cuenta que el aprendizaje colaborativo es un conjunto de métodos de instrucción o entrenamiento para uso en grupos pequeños [8], Childprogramming propone la dimensión colaborativa, que intenta incrementar la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos mediante el desarrollo de software [2].

El proceso ChildProgramming está basado en la aplicación de prácticas ágiles, colaborativas y cognitivas, exploradas y empíricamente evaluadas con equipos de niños de edad escolar. Su ciclo de vida se basa en un marco de trabajo que comprende las fases de prejuego, juego y postjuego [8][9], como se aprecia en la figura 9.

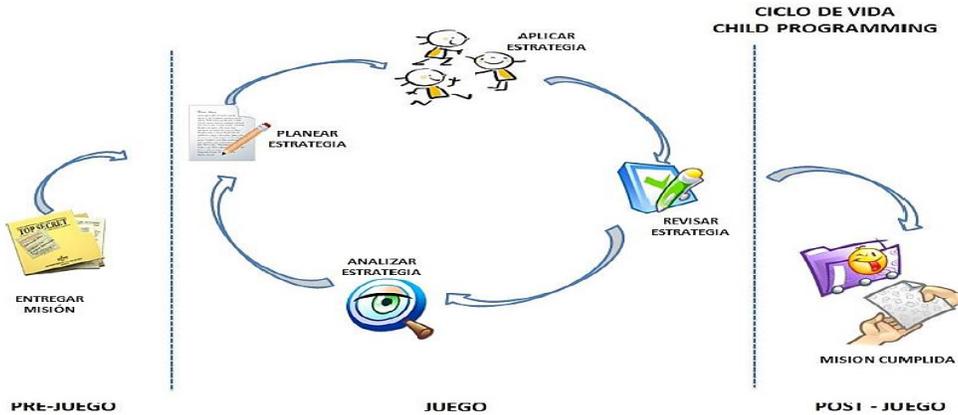


Figura 9. Ciclo de vida del proceso de ChildProgramming inicial [43]

El modelo ha ido evolucionando y su última versión ha incorporado mejoras relacionadas con la gamificación, obteniendo el modelo Childprogramming – G [44], ver figura 10, donde se puede apreciar una mejora significativa respecto a las tareas de equipo y el rol tutor.

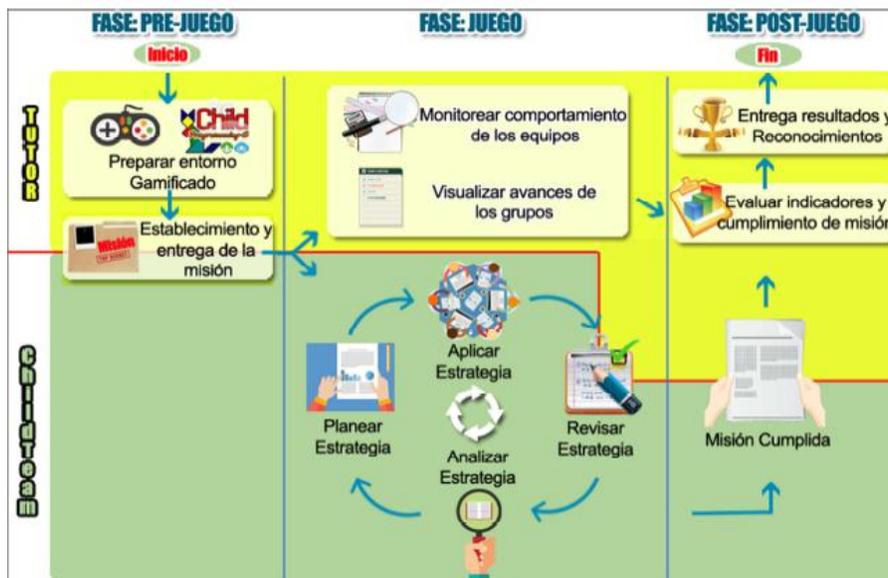


Figura 10. Ciclo de vida del proceso de ChildProgramming gamificado (Childprogramming-G) [45]

2.7.1 Gamificación en ChildProgramming

La gamificación se define como la aplicación de elementos básicos que hacen divertidos y atractivos a las cosas que normalmente no son consideradas un juego [46]. De igual manera se refiere a “La adopción de la tecnología de juegos y los métodos de diseño de juegos fuera de la industria de los videojuegos” [47]. “El proceso de utilizar pensamiento de juego y mecánicas de juego para resolver problemas y atraer a los usuarios” [48][47].

Childprogramming en su versión gamificada ha tenido en cuenta la gamificación orientada en un ambiente educativo y colaborativo, incorporando elementos de juego en un contexto de aula con el objetivo de comprometer a los estudiantes con el aprendizaje por medio de actividades que les brinden experiencias divertidas propias de los juegos para los niños [49]. Dentro de la gamificación los siguientes conceptos juegan un papel muy importante:

- Mecánicas de juego: reglas que pretenden incrementar la motivación y el compromiso de los jugadores mediante la consecución de objetivos y con la finalidad de obtener reconocimiento [43]
- Dinámicas de juego: son las necesidades e inquietudes humanas que motivan a las personas y son el resultado de utilizar las mecánicas de juego [43]

2.8 Metodologías utilizadas para el desarrollo del proyecto

2.8.1 Método CSACE (Case Study based Analysis in Collaboration Engineering)

El método *CSACE (Case Study based Analysis in Collaboration Engineering)* [7], el cual está basado en estudios de caso para establecer las necesidades de procesos colaborativos desde las interacciones del equipo, así como la evaluación empírica de la efectividad del proceso colaborativo resultante [7].

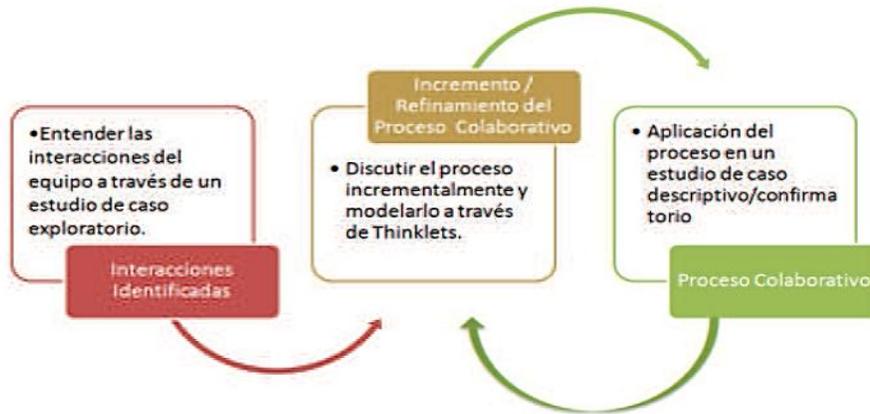


Figura 11. Estructura fundamental del método CSACE

Según Runeson et al. [50], aunque los estudios de casos se utilizan comúnmente en áreas como la psicología, la sociología, la ciencia política, trabajo social y en área de negocios, la aceptación de trabajos empíricos en ingeniería de software y sus contribuciones a aumentar el conocimiento está en continuo crecimiento. Runeson et al. describe que el estudio de caso es una metodología de investigación adecuada para la ingeniería del software debido a que estudia un fenómeno contemporáneo en su contexto real, buscando mantener la integridad y las características significativas de los eventos; además se lleva a cabo cuando el investigador tiene poco control sobre los eventos y cuando los objetos de estudio son más fáciles de observar en grupo que de manera aislada.

Los estudios de caso son una metodología de investigación que ha demostrado ser útil para la Ingeniería del Software en el análisis de sujetos de estudio, que son más fáciles de observar en grupo que de forma aislada [50], por lo cual son un enfoque viable para el análisis y evaluación de los procesos colaborativos [7].

CSACE describe tres fases entre las cuales se debe entender las interacciones del equipo a través de un estudio de caso exploratorio, de igual forma al conocer las interacciones, se debe discutir el proceso incrementalmente y modelarlo a través de thinklets, para posteriormente aplicarlo y evaluarlo en un caso de estudio descriptivo/confirmatorio.

2.8.2 Metodología para el desarrollo de proceso Colaborativos

La metodología para el desarrollo de procesos colaborativos consta de las siguientes fases [8]: diagnóstico de la tarea, evaluación de la actividad, descomposición de la actividad, relación de Thinklets, documentación de diseño y validación del diseño. En la Figura 12 las cajas en gris de la parte superior, son las entradas que se requieren en cada una de las fases y las cajas en la parte inferior, describen la salida de cada fase [8].

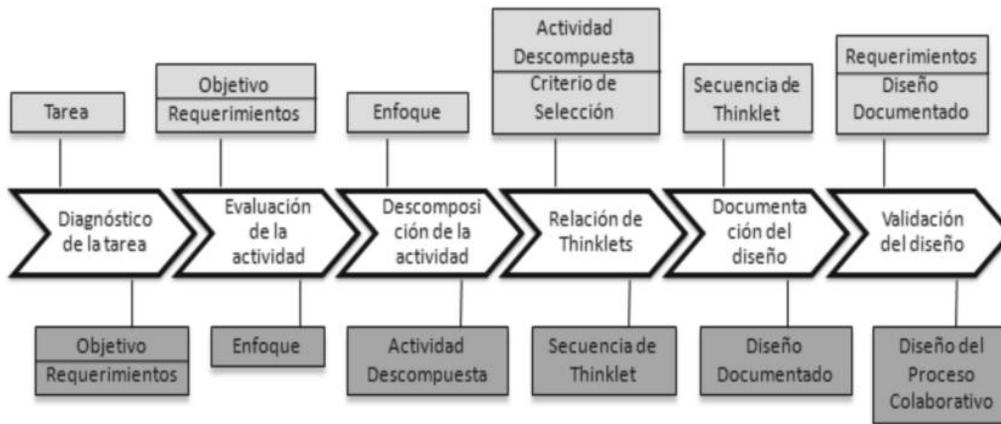


Figura 12. Metodología para el Desarrollo de proceso Colaborativos [51].

Descripción de las fases:

Fase 1: Diagnóstico de la tarea: Según la metodología, en esta fase se describe de forma detallada el proceso objeto de estudio. Incluye los entregables, recursos, participantes, ventajas y desventajas [37].

Fase 2: Evaluación de la actividad: Para el proceso objeto de estudio, son identificadas las actividades generales que lo componen y la secuencia entre ellas.

Fase 3: Descomposición de la actividad En esta fase son especificadas las subactividades que componen cada una de las actividades generales identificadas en la fase anterior (Fase 2). Luego de obtener las subactividades, son definidas cuales se realizarían de forma colaborativa, y a cada subactividad definida como colaborativa se le asocia uno o más patrones de colaboración.

Fase 4: Relación de Thinklets En esta fase son asociados los Thinklets a las subactividades colaborativas que componen el proceso objeto de estudio. Los Thinklets identificados deben adecuarse a los procesos que conforman las subactividades, a los recursos, al grupo y hasta a las propias habilidades de las personas involucradas en la ejecución de los procesos colaborativos [52].

Fase 5: Documentación de la técnica: a partir de la información obtenida en las fases anteriores, en esta fase deben generarse los siguientes elementos definidos en Ingeniería de Colaboración[53]: modelo de facilitación del proceso y agenda detallada.

- Modelo de Facilitación del Proceso (MFP): es utilizado para mostrar el flujo del proceso y los elementos tales como thinklet, número de secuencia, patrón de colaboración y nombre, relacionados con cada una de las actividades que conforman el proceso.
- Agenda detallada: documento que presenta de manera clara y detallada la información de las actividades que forman parte del proceso diseñado.

2.9 Trabajos relacionados

En esta sección se presentan los trabajos relacionados con los temas base de esta investigación los cuales son relevantes para el desarrollo del presente trabajo, entre ellos se tienen: modelos colaborativos de enseñanza de programación de software, diseño y estructuración de procesos colaborativos.

En primera instancia se presentan algunos estudios enfocados en modelos colaborativos para la enseñanza de la programación de software, entre ellos se tiene ChildProgramming [14] el cual expone el resultado de una propuesta de aprendizaje en equipo que permite a los niños, asimilar conceptos de programación mediante la aplicación de prácticas ágiles, cognitivas y colaborativas, para soportar el desarrollo de software. Se compone de tres dimensiones [14]: la dimensión cognitiva, la dimensión ágil, y la dimensión colaborativa. Este trabajo se convierte en el punto de referencia para el desarrollo de ChildProgramming-C. Por otra parte, existe una extensión con técnicas de gamificación [44], llamada ChildProgramming-G incorporando conceptos de la teoría de la gamificación, como una estrategia clave para incrementar el desempeño, la persistencia, iniciativa y atención a los

detalles de los equipos de desarrollo de software, e incentivar el aprendizaje de los niños que se ven identificados con dichas mecánicas, y evidencia la importancia del trabajo colaborativo para que los participantes logren alcanzar sus objetivos mientras juegan y aprenden, sin embargo no se presenta estructuración de las actividades colaborativas.

De igual manera, en [54] se propone un modelo híbrido que integra el potencial del Aprendizaje Basado en Problemas (ABP) como un medio para desarrollar habilidades de trabajo cooperativo en la búsqueda de soluciones, y la eficacia de la programación en parejas, como estrategia de aprendizaje colaborativo y herramienta para mejorar las habilidades de programación en estudiantes universitarios.

En [55] se presenta un modelo didáctico para la enseñanza de la programación de software con niños entre los 6 a 12 años de edad, al respecto conviene decir, que su origen se da al considerar el potencial que tiene la programación de computadoras en el desarrollo de habilidades de pensamiento, y propone utilizar la heurística de Poyla, la cual consiste en una serie de pasos y preguntas para analizar el texto de un problema [56]. Este modelo plantea las siguientes fases: segmentación del grupo, concepto, pensamiento computacional y algoritmo, heurística de Poyla, diseño de algoritmo, práctica de programación y exposición magistral.

Al llegar a este punto cabe que mencionar el modelo expuesto en [57], donde se busca determinar los elementos que intervienen mediante la aplicación de aprendizaje colaborativo en E-Learning, con el fin de aumentar la motivación de los estudiantes de secundaria de Indonesia, este modo describe las competencias del estudiante en cada uno de los grados de escolaridad y muestra algunas herramientas y lenguajes de programación que podrían utilizarse de acuerdo a los nivel de educación de dicho país. Plantea tres componentes principales: las habilidades técnicas de los usuarios de E-learning, los contenidos digitales y la tecnología.

En la tabla 9 se presentan y analizan las propuestas descritas en esta sección, respecto a seis aspectos de comparación.

Propuesta	Dirigido a Niños	Estrategia o método de colaboración	Estructura actividades desde la Ingeniería de la Colaboración	Evalúa la Colaboración	Entorno de programación utilizado o lenguaje de programación	Metodología de estudio
Child Programming [13].	Si	Propone algunas prácticas colaborativas e identifica algunos patrones de colaboración	No	No	Scratch	Estudio de caso
Child Programming-G [44].	Si	Sigue prácticas de Child Programming.	No	No	Scratch	Estudio de caso
ChildThink: Propuesta de un Modelo Didáctico para la Enseñanza de la Programación en Niños [55].	Si	Segmentación del grupo.	No	No	Scratch, Alice y Lego	Práctica
A Collaborative Approach to Learning Programming: A Hybrid Learning Model [58].	No	Programación en parejas	No	No	Java	Práctica
E-Learning Model for Teaching Programming Language for Secondary School Students in Indonesia [57].	Si	Interacción con estudiantes y docentes mediante redes sociales	No	No	Propone varios entornos, mas no se ha aplicado	No especificado

Tabla 9. Comparación de modelos relacionados.

Es de notar que en los modelos presentados se evidencian aportes a los procesos de desarrollo de software y la enseñanza de la programación principalmente en niños, aunque la mayoría de las propuestas plantean la organización de equipos de trabajo, las actividades expuestas no son orientadas desde el enfoque de la Ingeniería de la Colaboración por lo

cual no se ofrece una estructura que guíe el proceso de desarrollo de las actividades colaborativas, ni evalúan la colaboración generada en los equipos de trabajo.

En segunda instancia, respecto a la enseñanza de la programación de software, Pears [58] ofrece una revisión de la literatura sobre la enseñanza de la introducción a la programación, y muestra una visión general de temas tales como: planes de estudios, lenguajes y herramientas existentes para apoyar el aprendizaje del desarrollo de software y el pensamiento computacional, no obstante concluye indicando que a pesar del gran volumen de literatura en esta área, hay poca evidencia sistemática para apoyar un determinado enfoque.

Del mismo modo, se encuentran diversas experiencias con el uso de herramientas de desarrollo de software para niños y jóvenes. Es así como en [59] se presenta una estrategia para involucrar estudiantes de secundaria en entornos recreativos, que permita el aprendizaje de la lógica de programación, mediante el uso del kit de robótica Lego Mindstorms NXT 2.0, dicha propuesta nace a partir del informe de varias universidades brasileñas, donde el número de fallos en el área de lógica de programación alcanzan el 60%. Por otra parte, en [60] se presenta la experiencia de utilizar Alice para enseñar un curso de Introducción a la informática mediante la estrategia de programación por parejas, al final del módulo los datos cualitativos indicaron que los estudiantes, entendieron conceptos básicos de programación y algoritmos. Ahora bien, en [61] se muestra como estudiantes entre 10 y 12 años de edad, aprendieron a usar Stagecast Creator, HANDS y Visual Basic para crear juegos simples y animaciones, los estudiantes trabajaron individualmente en sus propios equipos la mayor parte del tiempo, aun así se unieron para hacer proyectos más grande, obteniendo mejores resultados.

De igual forma Crook [62], muestra el uso de Scratch en el aula como parte del plan de estudios de las TIC, para los niños de siete y ocho años en una escuela primaria en Brighton en el Reino Unido. En este caso Scratch se convierte en un elemento central para enseñar y ampliar el currículo de las TIC más allá de la enseñanza de habilidades básicas de ofimática, y ofrecer un entorno creativo para acercar a los niños a la programación de computadores. Los investigadores concluyen que Scratch ofrece una oportunidad para enseñar a los niños, además de participar en los procesos de colaboración en la planificación, diseño, creación y fijación de una aplicación informática, mejorando así las habilidades de comunicación entre los niños. Cabe notar que Scratch se ha convertido en

uno de los entornos más utilizados de desarrollo de software para niños, es así como programadores de todo el mundo suben en línea diariamente más de 1.500 nuevos proyectos al sitio oficial, con el código fuente libremente disponible para compartir y reutilizar, el público principal en el sitio se encuentra entre las edades de ocho a dieciséis años [63].

En tercera instancia, respecto a los estudios relacionados con el diseño y aspectos claves en la estructuración de actividades colaborativas, se tienen trabajos como [64] donde se considera que el entendimiento de la tarea es un determinante importante para el rendimiento de un equipo, y más aún cuando estos son heterogéneos. “Los miembros del grupo podrían estar utilizando las mismas palabras para los diferentes conceptos o palabras diferentes para referirse a un mismo concepto sin darse cuenta” [21]. Este trabajo contribuye a la construcción de conocimiento compartido en grupos heterogéneos, mediante el uso de herramientas de la ingeniería de la Colaboración como lo son los thinklet.

De igual modo, en [65] se presentan los factores y subfactores que definen la colaboración y el trabajo colaborativo, dicha investigación hace parte del proyecto integrado europeo CoSpace. A través de la revisión de la literatura y del trabajo empírico con empresas asociadas en los sectores: aeroespacial, automoción, la construcción, se obtienen siete categorías principales de factores que intervienen en la colaboración: el contexto, el apoyo, las tareas, los procesos de interacción, los equipos, las personas y factores fundamentales, los cuales se resumieron en un marco de referencia [65].

Por otra parte, en [37] se presenta cómo la Metodología para el Desarrollo de Procesos Colaborativos [66], puede ser utilizada para incorporar aspectos colaborativos en las actividades que conforman procesos complejos, que requieren la participación de varias personas. Así mismo, en [67] se presenta la *Metodología para el Diseño de Métodos de Evaluación de Usabilidad Colaborativos* obtenida a partir de la Metodología para el Desarrollo de Procesos Colaborativos, ahí se muestra cómo el diseño colaborativo aporta significativamente en la evaluación de la usabilidad cuando participan varias personas y ayuda a mejorar un proceso. En [68] se presentan algunos objetivos de la colaboración en la Ingeniería del Software, así mismo, se indaga sobre las herramientas de colaboración

existentes enfocadas en artefactos simples o en el proceso de desarrollo de una aplicación. También se menciona que la Ingeniería del Software es una actividad altamente cooperativa y promete continuar de esta forma.

3. Capítulo 3. Identificando actividades colaborativas

En este capítulo se presenta dos estudios de caso que permitieron comprender las interacciones colaborativas dentro de los equipos conformados por niños. Posteriormente se identifican las actividades que requieren trabajo colaborativo partiendo de un conjunto de criterios seleccionados con un equipo de nueve expertos en ingeniería de la Colaboración. Finalmente, en este capítulo se presentan la relación de los patrones de colaboración con cada una de las actividades colaborativas seleccionadas, y se proponen dos patrones de colaboración que fueron identificados a partir de las experiencias de los estudios de caso y del análisis conceptual del modelo ChildProgramming.

3.1 Estudios de casos exploratorios

Para el desarrollo del proyecto ChildProgramming – C se ha seguido el modelo SCASE, en la figura 13 se destaca la primera etapa del modelo CSACE que propone entender las interacciones de los equipos.



Figura 13. Primera fase del método CSACE [7]

Para identificar y entender las interacciones de los equipos mientras desarrollan software guiados con el modelo Childprogramming y su versión gamificada, se ejecutaron dos estudios de caso exploratorios donde niños entre 10 y 12 años desarrollaron aplicaciones apoyados en la herramienta de desarrollo Scratch 2.0 y la herramienta de seguimiento del proceso Gamitool, dichos estudios de caso se presentan en los Anexo B y Anexo C.

En los estudios de caso se analizó la colaboración del proceso de desarrollo de software desde cuatro aspectos, como elementos imprescindibles que se deben presentar en la colaboración para que el aprendizaje tenga lugar de manera efectiva[69], [15] y [70], ellos son:

- La interdependencia positiva.
- La igual participación.
- La responsabilidad individual.
- La definición de estrategias del equipo.

Así mismo se observó el desempeño del líder al considerarse una característica primordial en los equipos de trabajo conformados por niños, esta característica fue propuesta por los equipos de docentes que apoyaron esta investigación desde su experiencia.

3.2 Resultados de los estudios de casos exploratorios

3.2.1 Estudio de caso 1

El diseño y desarrollo del estudio de caso 1 se presenta en el Anexo B. La unidad de análisis para este estudio de caso fueron los equipos de trabajo. El estudio de fue de tipo embebido considerando 5 unidades de análisis con 24 sujetos de investigación y se desarrollaron dos actividades.



Figura 14. Evidencias tofográficas estudio de caso 1 – actividad 1

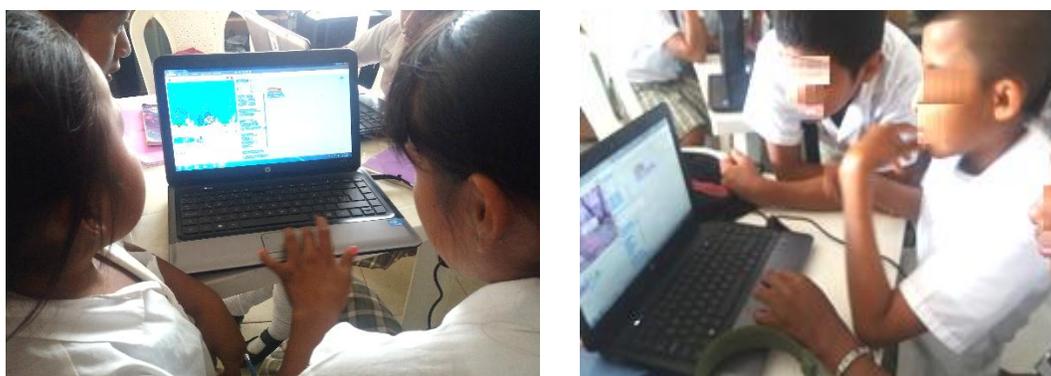


Figura 15. Evidencias tofográficas estudio de caso 1 – Actividad 2

De este primer estudio de caso se logró observar que la mayoría de los niños muestran motivación durante el proceso de desarrollo de software de forma colaborativa, pero a pesar de explicar previamente las prácticas colaborativas descritas en el modelo, algunos grupos no mostraron interés por las actividades colaborativas relacionadas con la selección del nombre del equipo, la asignación de roles y responsabilidades, la elección del líder, la definición y ejecución de una estrategia para resolver el problema; por lo tanto en algunos equipos se presentaron improvisaciones, discusiones, y se evidenció la poca participación de algunos integrantes al verse opacados por otros compañeros.

La tabla 10 presenta la valoración preliminar de los equipos de trabajo que se realizó al inicio de la investigación en el primer estudio de caso aplicando el modelo Childprogramming.

Evaluación de características de la colaboración en estudio de caso 1					
equipos	Interdependencia Positiva	Igual participación	Responsabilidad individual	Definición de estrategia	Trabajo del líder
E1	3	2	3	1	2
E2	2	2	1	1	3
E3	4	2	3	2	4
E4	3	3	2	2	2
E5	2	2	2	2	1

Tabla 10. Valoración preliminar de los equipos de trabajo –Primer estudio de Caso

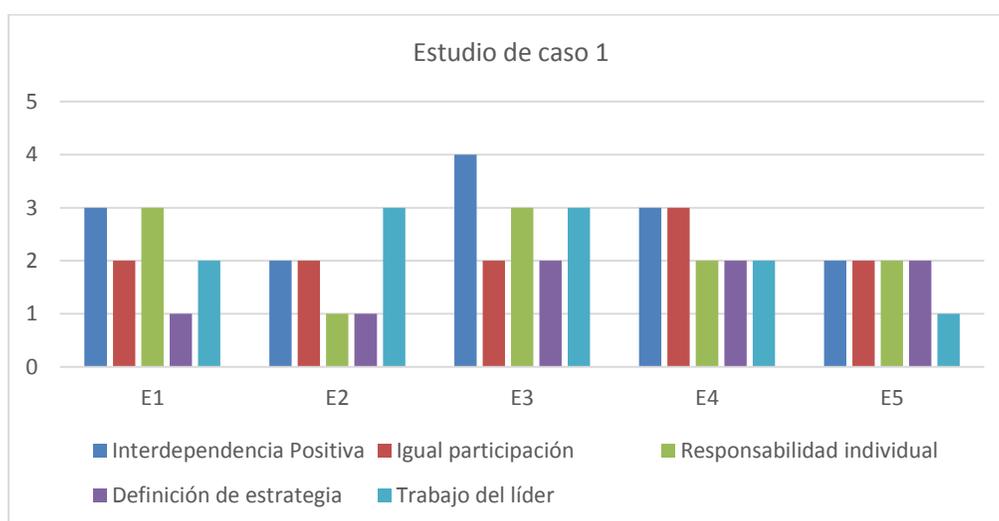


Figura 16. Análisis primer estudio de caso preliminar

3.2.2 Estudio de caso 2.

En el estudio de caso 2 se aplicó el modelo Childprogrammig-G, ver figura 17. Para este estudio de caso se seleccionó como dinámica de juego la recompensa, definida como una de las dinámicas de la gamificación importante para mantener motivados y comprometidos a los niños, dependiendo del comportamiento o de los puntos brindados [44]. Como mecánica de juego se seleccionó la tabla de clasificación la cual, tiene como objetivo proporcionar el deseo de aspiración y aportar una comparativa entre grupos de niños que

lleve a una visión general del desarrollo de las actividades y de los estados de los equipos [44].

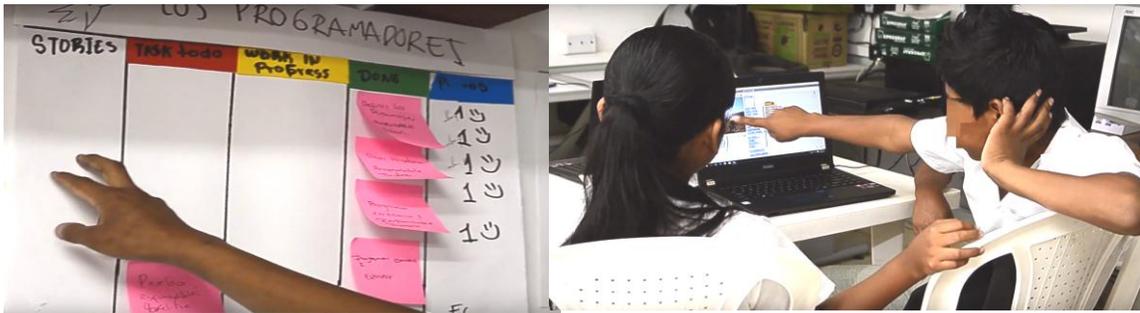


Figura 17. Evidencias fotográficas estudio de caso 2 – Taskboard y equipos

La tabla 11, muestra la valoración de los equipos de trabajo realizada en el estudio de caso utilizando el modelo Childprogramming–G, dando mayor relevancia a las dinámicas y mecánicas de juego.

Evaluación de características de la colaboración en estudio de caso 2					
Equipos	Interdependencia Positiva	Igual participación	Responsabilidad individual	Definición de estrategia	Trabajo del líder
E1	2	4	4	3	4
E2	4	3	3	3	3
E3	4	3	4	2	3
E4	4	2	4	5	3
E5	3	2	3	3	2

Tabla 11. Valoración de los equipos de trabajo –Estudio de Caso con dinámicas y mecánicas de juego.

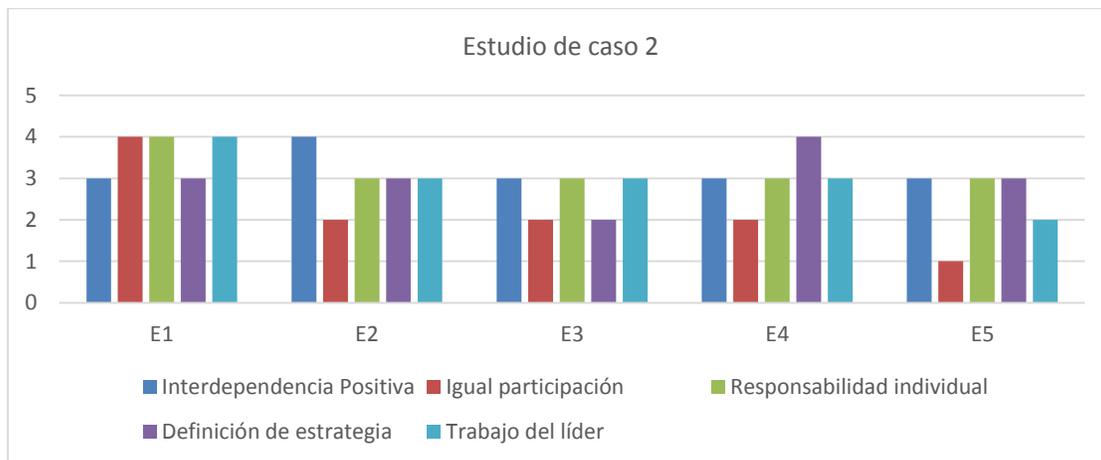


Figura 18. Análisis segundo estudio de caso

Comparando las tablas 10 y 11, y las figuras 16 y 18 se puede apreciar una mejora en la evaluación de las características de la colaboración, siendo la responsabilidad individual y la definición de estrategias las valoraciones que han mejorado significativamente. Los líderes de los equipos manifestaron que las dinámicas y mecánicas de juego fueron una gran herramienta para motivar a sus equipos de trabajo a mantenerse unidos, y concentrados durante el desarrollo de los retos. Cabe resaltar que también se evidenció la motivación al autoaprendizaje, pues el deseo de ganar los retos hacía que los niños al ver al docente ocupado atendiendo dudas de otros grupos, buscaran ayuda en video tutoriales, foros y otras páginas de Scratch, para no depender tanto del docente y poder avanzar en la búsqueda de sus objetivos.

Sin embargo, en este estudio de caso se notó la importancia de guiar al docente en la planeación y ejecución de las actividades colaborativas del modelo y realizar seguimiento y control durante el proceso, además, la necesidad de mejorar en aspectos colaborativos.

El estudio de caso 1 y 2 permitió entender las interacciones de los equipos, así como posibles mejoras al proceso. Cabe mencionar que estos primeros estudios de caso fueron parte del proyecto de aula “programando retos” del área de informática en la institución educativa Agroempresarial Huasano de Caloto, y fue presentado en la convocatoria: “*V Encuentro Nacional de Experiencias Significativas con uso pedagógico de TIC*”, organizada por el Ministerio de Educación de Colombia en el mes de junio del 2017, siendo una de las mejores experiencias en el país, por buscar el desarrollo del pensamiento computacional colaborativamente en niños de zona rurales.

3.3 Formulación de ChildProgramming como un proceso colaborativo

3.3.1 Identificando las actividades del modelo ChildProgramming que requieren trabajo colaborativo para su ejecución.

Promover la interacción, valorar el aporte individual de cada estudiante en las tareas y asumir responsabilidades, desarrollando habilidades personales y de trabajo en grupo tales como debatir, cuestionar, tomar decisiones, liderar, coordinar y evaluar, son algunas de las metas del Aprendizaje Colaborativo en la enseñanza de la Ingeniería del Software [71].

Para la evaluación de ChildProgramming como un proceso colaborativo se aplicó la segunda fase del método CSACE, la cual consiste en discutir el proceso incrementalmente y modelarlo a través de thinklets; para dicho modelado se tuvo en cuenta las fases del uno a la tres de la Metodología para el Desarrollo de Procesos Colaborativos, que permitió identificar las actividades colaborativas y las posibles mejoras para realizar un refinamiento del proceso.

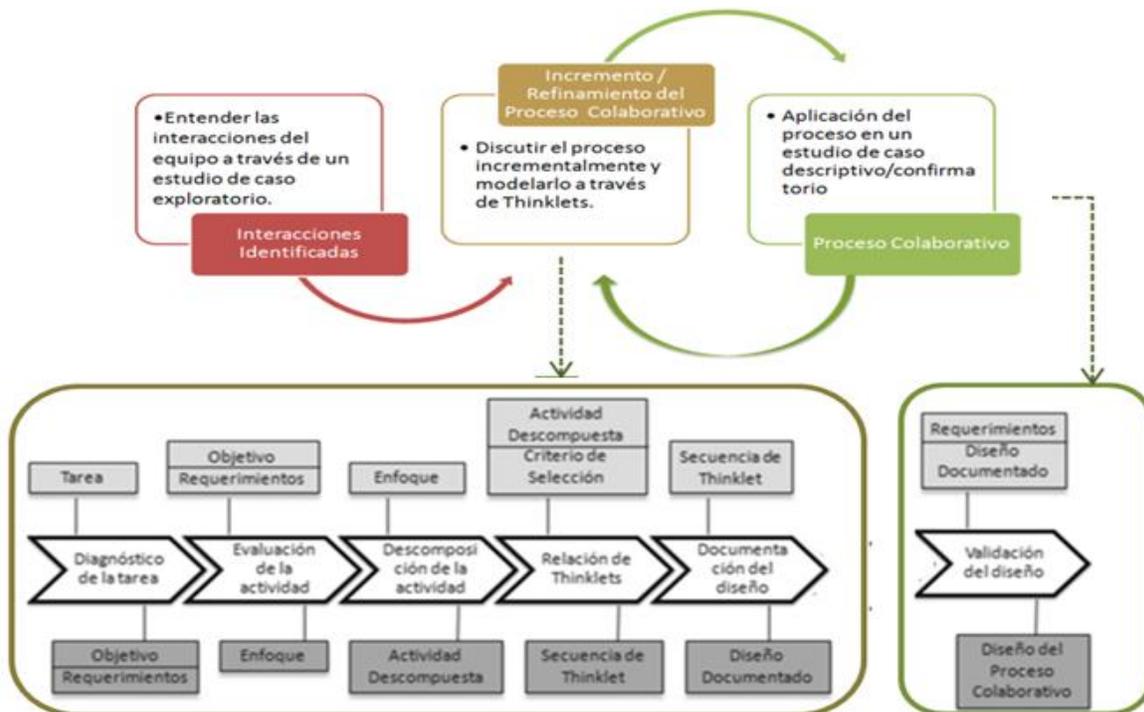


Figura 19. Integración del método CSACE y la Metodología para el Desarrollo de procesos Colaborativos

3.3.1.1 Aplicación fase 1-3 de la Metodología para el desarrollo de Procesos Colaborativos

El primero objetivo de esta investigación está relacionado con la identificación de las actividades del modelo ChildProgramming que requieren colaboración, para ello se

desarrollaran las fases 1, 2 y 3 de la Metodología para el Desarrollo de Procesos colaborativos, las cuales, conllevan a descomponer e identificar las actividades que requieren realizarse de forma colaborativa.



Desarrollo fase 1: diagnóstico de la tarea. En esta fase se describe de forma detallada el modelo ChildProgramming- G como objeto de estudio teniendo en cuenta que es la versión más actual del modelo Childprogramming, a continuación, se presenta la descripción, los entregables, los recursos, los participantes, las ventajas y desventajas de Childprogramming –G.

Diagnóstico de la tarea - Objeto de estudio: modelo ChildProgramming – G
<p>Descripción: ChildProgramming es un modelo para el desarrollo de software llevado a cabo por niños, se compone de tres dimensiones [14]: la dimensión cognitiva, considerada como el esfuerzo que realizará un niño para comprender, analizar y apropiar situaciones presentes en las tareas definidas en el modelo. La dimensión ágil, basada en la promulgación de los valores del manifiesto de las metodologías ágiles de desarrollo de Software; y la dimensión colaborativa, que intenta incrementar la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos mediante el desarrollo de software [14].</p> <p>El modelo define tres fases: pre-juego, juego y post-juego, y propone los siguientes roles: profesor, guía del equipo, equipo de trabajo e investigador. El proceso ChildProgramming guía a los equipo conformados por niños a alcanzar sus objetivos de desarrollo y les brinda un conjunto de prácticas ágiles, cognitivas y colaborativas para el desarrollo de las actividades de construcción de software [14].</p> <p>El Modelo ChildProgramming-G es una extensión del inicial modelo ChildProgramming a través de la incorporación de la dimensión lúdica, que considera la gamificación como base fundamental incluyendo mecánicas y dinámicas de juego, que intervienen directamente sobre la motivación de los niños con el fin de incrementar su desempeño. ChildProgramming-G es el resultado de una serie de estudios de caso enmarcados en actividades escolares y, en los que se realizó un proceso de formulación, recolección y análisis para encontrar las mecánicas y dinámicas que más impacto tienen en los niños en el momento de realizar el aprendizaje de la construcción de software [childprogramming-g].</p>

El proceso ChildProgramming-G está orientado a brindar un marco de trabajo para el aprendizaje de la programación en niños, está dividido en tres fases que están compuestas por varias actividades que cubren todo el proceso desde el inicio de la actividad, hasta que se entrega un producto software, se analiza, se revisa y se obtienen resultados. El propósito de este proceso es convertir el ambiente de la enseñanza del desarrollo de software, en un ambiente de juego para facilitar el aprendizaje. Con la ayuda de actividades que se extraen a partir de la utilización de la gamificación, se puede crear un ambiente de juego sin serlo específicamente [72], y aprovechando las mecánicas de juego de los videojuegos se puede alcanzar un nivel de motivación tal que los niños mejoren su desempeño en el desarrollo de las actividades [44].

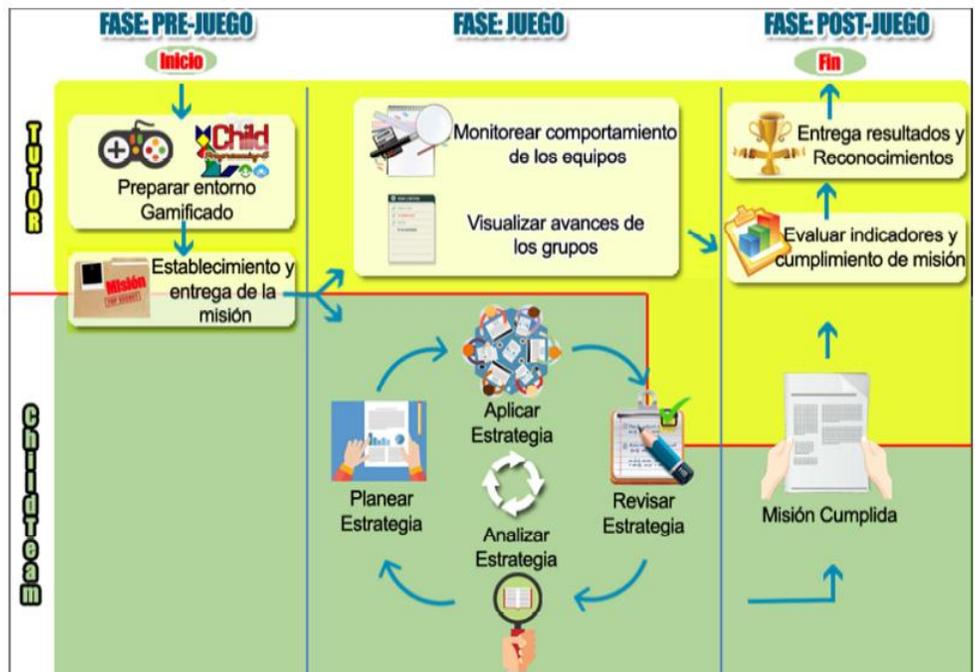


Figura 20. Ciclo de vida del proceso ChildProgramming-G

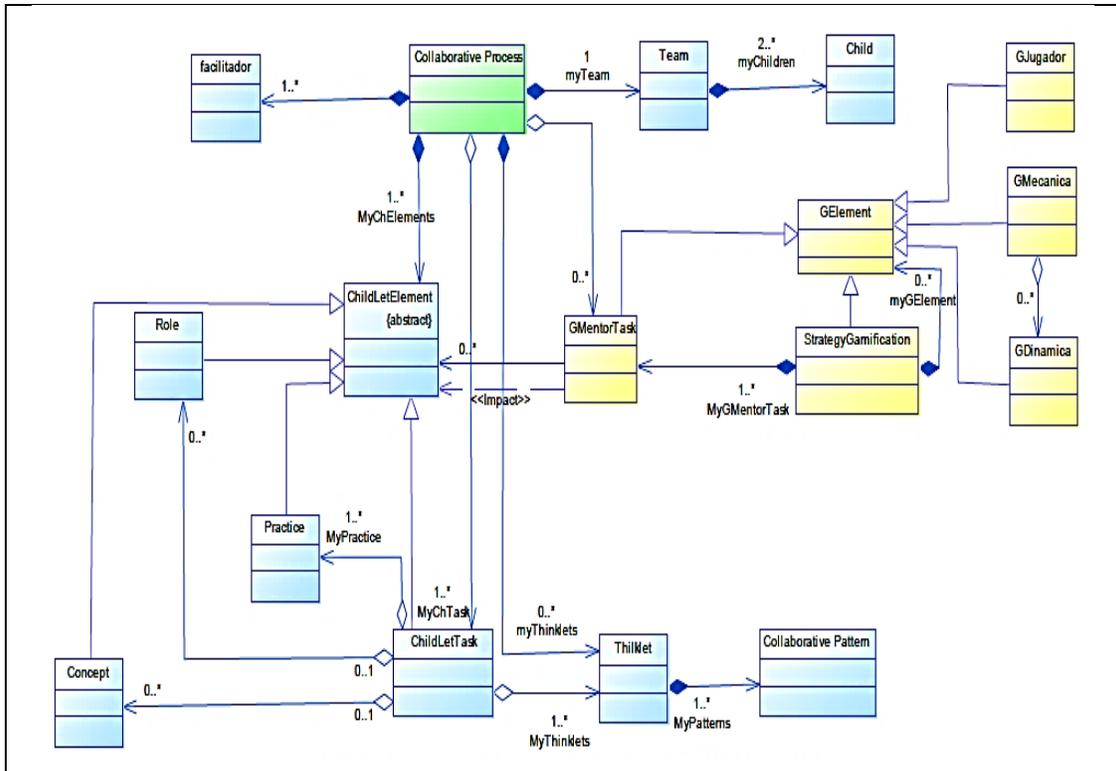


Figura 21. Modelo Conceptual ChildProgramming - G.

Entregables	<ul style="list-style-type: none"> - Versión final del software realizado por los niños (Misión cumplida). - Tablero de tareas, tablero de posiciones y puntuaciones según las mecánicas y dinámicas de juego elegidas.
Recursos	Equipos de cómputo, mesas, instructivos de requerimientos, papel, marcadores, post-it, cartelera, software para desarrollo, herramienta gamitool.
Participantes	Equipo de trabajo, guía del equipo, profesor y observador investigador.
Ventajas	<ul style="list-style-type: none"> - El proceso plantea estrategias de trabajo colaborativo. - Tiene en cuenta los enfoques ágiles para el desarrollo de software, así como prácticas cognitivas, basando su ciclo de vida del proceso Scrum [13]. - Ofrece un conjunto de dinámicas y mecánicas de juego. - Es una herramienta para desarrollar competencias sociales y de pensamiento computacional. - Ofrece mecánicas y dinámicas de juego.
Desventajas	<ul style="list-style-type: none"> - El modelo expone prácticas ágiles, lúdicas y colaborativas, sin embargo, no es claro donde se deben aplicar dichas prácticas.

	<ul style="list-style-type: none"> - Aunque se presenten prácticas colaborativas es necesario concientizar al estudiante de la importancia del trabajo colaborativo. - La presencia del observador podría afectar el desempeño de los niños. - Para que un docente aplique el modelo ChildProgramming-G por primera vez, puede ser necesaria la orientación de un experto, pues la descripción de las fases es poca si un docente desea aplicar el modelo por sí solo. - Falta definir recomendaciones para la gestión de la colaboración en los equipos de trabajo.
Objetivos	<ul style="list-style-type: none"> - Brindar un marco general de trabajo para el aprendizaje de la programación en niños bajo la lúdica, la colaboración y las prácticas ágiles de desarrollo de software.
Requerimientos	<p>Para el desarrollo de este proceso es necesario:</p> <ul style="list-style-type: none"> - Equipos de estudiantes - Docente guía - Investigador - Equipos de cómputo - Herramienta de desarrollo de software. - Hojas de block, pliegos de papel, lapiceros, marcadores etc.

Tabla 12. Descripción del proceso objeto de estudio.

Esta descripción detallada del modelo ChildProgramming- G como objeto de estudio permite una mejor contextualización y análisis del objeto de estudio para obtener las actividades que requieren trabajo colaborativo.



Desarrollo fase 2. Evaluación de la actividad: para el proceso objeto de estudio, son identificadas las actividades generales que lo componen y la secuencia entre ellas. Para el caso del modelo ChildProgramming las actividades generales serían las fases: prejuego, juego y postjuego. En la siguiente tabla se describe cada una de las fases propuestas en ChildProgramming y ChildProgramming-G.

Identificación de las actividades generales del modelo ChildProgramming			
ACTIVIDAD	fases ChildProgramming	fases ChildProgramming – g	diagramas
PREJUEGO	El propósito de esta fase es establecer el objetivo de la misión, denominada así a la entrega de los requisitos que involucra la actividad a realizar.	El propósito de esta fase es preparar el entorno gamificado, establecer y entregar el objetivo de la misión . Preparar el entorno gamificado es una actividad que da el mayor soporte a ChildProgramming-G, y de la cual van a depender las otras actividades agregadas al modelo. Preparar el entorno gamificado. Para preparar el entorno gamificado hay que realizar varias tareas que son: "Establecimiento de mecánicas de juego" que se van a usar en el desarrollo de las actividades de trabajo en los grupos de niños, Configuración de GamiTool herramienta de apoyo opcional) y Definir las reglas de juego.	<p>The diagram illustrates the 'FASE PRE-JUEGO' (Pre-game Phase). It starts with 'Inicio' (Start) in a red oval, leading to a yellow box labeled 'Preparar entorno Gamificado' (Prepare gamified environment). This box contains icons for a game controller and the 'Child' logo. Below this is another yellow box labeled 'Establecimiento y entrega de la misión' (Establishment and delivery of the mission), which includes a 'Misión' (Mission) icon. The entire process is framed by a vertical bar on the left with the word 'OBJETIVO' (Objective) written vertically. A red line points from the 'Misión' box to the 'Establecimiento y entrega de la misión' box.</p>

		<p>Con estas tareas y con la entrega de la misión se provee un punto de partida para gamificar el aula de clase y las tareas que van a realizar los grupos de niños.</p>	
<p>JUEGO</p>	<p>En esta fase se realizan una serie de iteraciones denominadas por ChildProgramming Rondas. El propósito de esta fase es cumplir con la misión y dejar listo el entregable que garantiza el cumplimiento de los objetivos propuestos para la actividad. Una ronda en Child Programming incluye cuatro Estaciones a seguir, las cuales son:</p> <p>Planear la Estrategia Aplicar la Estrategia Revisar la Estrategia Analizar la Estrategia.</p>	<p>En esta fase las iteraciones (Rondas) del modelo anterior (ChildProgramming) continúan igual con el mismo propósito u objetivo: cumplir con la misión y dejar listo el entregable que garantiza el cumplimiento de los objetivos propuestos para la actividad.</p> <p>La ronda se sigue describiendo igual que en ChildProgramming y continúa incluyendo cuatro estaciones a seguir, las cuales son: Planear la estrategia, Aplicar la estrategia, Revisar la estrategia y Analizar la estrategia. Las estaciones Revisar la Estrategia y Analizar la Estrategia se actualizan y las otras dos estaciones se dejan como están para el nuevo modelo. En esta fase de ChildProgramming-G se incluyen otras actividades, las cuales son “Monitorear el comportamiento de los equipos” y “Visualizar actividades”</p>	<p>FASE: JUEGO</p>  <p>El diagrama ilustra el ciclo de la Fase Juego. En la parte superior, un recuadro amarillo contiene las actividades 'Monitorear comportamiento de los equipos' y 'Visualizar avances de los grupos'. Abajo, un ciclo circular muestra cuatro estaciones: 'Planear Estrategia', 'Aplicar Estrategia', 'Revisar Estrategia' y 'Analizar Estrategia', conectadas por flechas azules que indican un flujo continuo. A la izquierda del ciclo, una barra vertical con iconos representa un progreso o puntuación.</p>

Tabla 13. Descripción de actividades generales ChildProgrammig [14].

<p>POSTJUEGO</p>	<p>En esta fase se hace entrega de la Misión Cumplida. En Child Programming, la Misión Cumplida corresponde a la solución completamente implementada acompañada de todo el material asociado al desarrollo de la actividad.</p>	<p>ChildProgramming-G conserva el objetivo del anterior modelo en esta fase, el cual es entregar la Misión Cumplida. La Misión Cumplida corresponde a la solución completamente implementada acompañada de todo el material asociado al desarrollo de la actividad. El Equipo de Trabajo y el Guía del Equipo entregan la Misión Cumplida al Profesor. El profesor al cierre de esta Misión, debe evaluar que el trabajo del equipo refleje un proceso de aprendizaje y que haya permitido, de forma efectiva, cumplir con los objetivos de aprendizaje propuestos, pero aquí es donde ChildProgramming-G hace su aporte e incluye otras actividades que son Evaluar indicadores y cumplimiento de misión, Entrega de resultados y reconocimientos.</p>	
-------------------------	---	---	--



Desarrollo fase 3: Descomposición de la actividad. En esta fase son especificadas las subactividades que componen cada una de las actividades generales identificadas en la fase anterior (Fase 2). De este modo las actividades corresponden a cada una de las fases del modelo, las cuales son: prejuego, juego y postjuego.

Actividad 1	Subactividad	
1. Prejuego	1.1	Preparar el entorno gamificado
	1.1.1	Establecimiento de mecánicas y dinámicas de juego.
	1.1.2	Configuración de GamiTool,
	1.1.3	Definir las reglas de Juego
	1.2	Establecer la misión.
	1.3	Entregar la misión.
Actividad 2	Subactividad	
2. Juego	2.1	Planear la Estrategia.
	2.2	Aplicar la Estrategia.
	2.3	Revisar la Estrategia.
	2.4	Analizar la Estrategia.
	2.5	Monitorear el comportamiento de los equipos.
	2.6	Visualizar actividades.
Actividad 3	Subactividad	
3. Postjuego	3.1	Entregar la misión cumplida.
	3.2	Evaluar indicadores y cumplimiento de la misión.
	3.3	Entrega de resultados y reconocimientos.

Tabla 14. Descomposición de actividades de ChildProgrammig [14].

Luego de obtener las subactividades, son definidas cuales de ellas se realizarían de forma colaborativa, y a cada subactividad definida como colaborativa se le asocia uno o más patrones de colaboración. Para definir las actividades colaborativas se seleccionó un conjunto de criterios como se muestra a continuación.

3.3.1.2 Selección de criterios para identificar actividades realizadas por niños que requieren trabajo colaborativo.

Para identificar cuáles actividades del modelo ChildProgramming-G requieren trabajo colaborativo para su ejecución se tuvo en cuenta los criterios 1, 2, 3, 4 presentados en la siguiente tabla y propuestos por [67], sin embargo, al considerar la interdependencia

positiva y la responsabilidad individual como aspectos fundamentales de los procesos colaborativos, se proponen los criterios 5 y 6. Los siguientes criterios fueron evaluados por nueve expertos en Ingeniería de la Colaboración, seis expertos nacionales y 3 internacionales de las siguientes universidades: Universidad de Lleida, Universidad Nacional de La Plata, Universidad de Granada, Universidad Autónoma de Occidente, Fundación Universitaria de Popayán y la Universidad del Cauca. Se utilizó como instrumento de investigación la encuesta presentada en el Anexo D. En la tabla 15 se presentan los resultados:

N°	Criterio para evaluar las actividades que requieren trabajo colaborativo (Aplicación en proceso de desarrollo de software por niños)	Aplicación en proceso de desarrollo de software por niños								
		Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7	Exp8	Exp9
C1	1. La ejecución de la actividad y el plan de trabajo requieren incluir varias personas, las cuales pueden tener diferentes roles.	Si	Si	Si	Si	Si	Si	Si	Si	Si
C2	2. Se requiere contar con personas que tienen un alto grado de experticia en un área específica de conocimiento, para la ejecución de las actividades.	No	No	No	No	Si	No	Si	No	No
C3	3. Es necesario compartir conocimiento, recursos e información con otras personas.	Si	Si	Si	Si	Si	Si	Si	Si	Si
C4	4. Es necesario tener en cuenta los diferentes aportes, opiniones y puntos de vista de otros integrantes del grupo	Si	Si	Si	Si	Si	Si	Si	Si	Si

	que está ejecutando la actividad.									
C5	5. La actividad permite a los participantes asumir responsabilidades individuales	Si	No							
C6	6. Es necesario definir una estrategia para la ejecución de la actividad.	No	Si	No	Si	Si	Si	Si	Si	No

Tabla 15. Análisis de criterios para definir si una actividad requiere trabajo colaborativo para su ejecución

De acuerdo a la evaluación realizada, el criterio “se requiere contar con personas que tienen un alto grado de experticia en un área específica de conocimiento, para la ejecución de las actividades”, no sería aplicable en los equipos colaborativos de los niños, por lo tanto, los principales criterios para efectuar una valoración preliminar del carácter colaborativo de las actividades que conforman el modelo ChildProgramming - C son:

N° criterio	Criterio de evaluación	Aplicación en proceso de desarrollo de software por niños
C1	La ejecución de la actividad y el plan de trabajo requieren incluir varias personas, las cuales pueden tener diferentes roles.	Si
C2	Es necesario compartir conocimiento, recursos e información con otras personas.	Si
C3	Es necesario tener en cuenta los diferentes aportes, opiniones y puntos de vista de otros integrantes del grupo que está ejecutando la actividad.	Si
C4	La actividad permite a los participantes asumir responsabilidades individuales.	Si
C5	Es necesario definir una estrategia para la ejecución de la actividad.	Si

Tabla 16. Criterios seleccionados para definir si una actividad llevada a cabo por niños requiere trabajo colaborativo para su ejecución

3.4 Definición de actividades colaborativas de ChildProgramming

Para cada una de las subactividades del modelo ChildProgramming-G se aplican los criterios para identificar si una actividad llevada a cabo por niños requiere trabajo colaborativo para su ejecución. Cada actividad es analizada a partir de la definición conceptual que ofrece el modelo.

Actividad		Subactividad	Actor	C1	C2	C3	C4	C5	Colaborativa (Si- no)
1- preJuego	1.1	Preparar el entorno gamificado.	Docente						No
	1.1.1	Establecimiento de mecánicas y dinámicas de juego.	Docente Investigador Equipo						Si
	1.1.2	Configuración de GamiTool.	Docente						No
	1.1.3	Definir las reglas de Juego	Docente- equipo	x	x	x	x	x	Si
	1.2	Establecer la misión.	Docente						No
	1.3	Entregar la misión	Docente - equipo						No
	1.3.1	Entrega de la descripción detallada de la actividad	Docente						No
	1.3.2	Identificación de tareas	Equipo	x	x	x	x	x	Si
	1.3.3	Priorización de tareas	Equipo	x	x	x	x	x	Si
2. Juego	2.1	Planear la estrategia							

	2.1.1	Definición de estrategia	Equipo	x	x	x	x	x	Si
	2.1.2	Definición de tareas	Equipo	x	x	x	x	x	Si
	2.2	Aplicar la Estrategia	Equipo	x	x	x	x	x	Si
	2.3	Revisar la Estrategia	Equipo	x	x	x	x	x	Si
	2.4	Analizar la Estrategia	Equipo	x	x	x	x	x	Si
	2.5	Monitorear el comportamiento de los equipos	Docente						No
	2.6	Visualizar actividades	Docente						No
3. PostJuego	3.1	Entregar la Misión Cumplida	Líder del equipo						No
	3.2	Evaluar indicadores y cumplimiento de la misión	Docente						No
	3.3	Entrega de resultados y reconocimientos	Docente						No

Tabla 17. Subactividades de ChildProgramming que requiere trabajo colaborativo para su ejecución

El modelo ChildProgramming propone tres fases (Actividades): prejuego, juego y postjuego. Al realizar un análisis de las subactividades que requieren trabajo colaborativo, teniendo en cuenta los criterios de la tabla 16, se puede apreciar que las actividades 1.1.3, 1.3.2, 1.3.3, 2.1.1, 2.1.2, 2.2, 2.3, 2.4 en la ejecución de la actividad y el plan de trabajo requieren incluir varias personas, las cuales pueden tener diferentes roles; así mismo, es necesario compartir conocimiento, recursos e información con otras personas, y se debe tener en

cuenta los diferentes aportes, opiniones y puntos de vista de otros integrantes del grupo que está ejecutando la actividad.

En este punto de se han identificado las actividades colaborativas del modelo, las cuales se convierte en el foco principal del trabajo.

3.4.1 Asociación de patrones de colaboración a las subactividades

Los *patrones de colaboración* se definen en términos del “movimiento del grupo desde su estado inicial hasta su estado final” [36]. Cada patrón asocia subpatrones que se pueden relacionar con las actividades. En la siguiente tabla se exponen los patrones de colaboración que pueden ser asociados a cada uno de las subactividades colaborativas identificadas.

Patrones de colaboración	Descripción
Patrón Generación	Es un patrón a partir del cual el grupo crea contenido. Consiste en pasar de tener pocos a muchos conceptos que son compartidos por el grupo.
Patrón Reducción	El objetivo de este patrón es mantener sólo la información que cumple con un determinado criterio o criterios. Consiste en pasar de tener muchos conceptos a unos pocos que el grupo considere que requieren mayor atención.
Patrón Clarificación	El objetivo de este patrón es lograr el entendimiento común de conceptos manejados por el grupo. Consiste en pasar de tener un menor a un mayor conocimiento compartido de los conceptos, las palabras y frases usadas para expresarlos.
Patrón Organización	Consiste en pasar de tener un menor a un mayor conocimiento de las relaciones entre los conceptos que el grupo esté considerando.
Patrón Evaluación	Consiste en pasar de un menor a un mayor conocimiento del valor relativo de los conceptos bajo consideración. Este patrón tiene como efectos apoyar a la toma de decisiones y a la comunicación del grupo.

Patrón construcción de consenso	Moverse de tener pocos a muchos miembros del grupo quienes estarán dispuestos a comprometerse para un objetivo.
--	---

Tabla 18. Patrones definidos en la Ingeniería de la Colaboración [30].

A continuación, se asocian los patrones de colaboración a cada una de las subactividades y posteriormente se realiza el análisis de cada de una ellas.

Actividad	id	Subactividad	Patrones asociados
PreJuego	1.1.3	Definir las reglas de Juego	Patrón generación Patrón construcción de consenso Patrón liderar
	1.3.2	Identificación y priorización de tareas	Patrón generación Patrón organización Patrón evaluación
Juego	2.1.1	Definición de estrategia	Patrón generación Patrón de organización
	2.1.2	Definición de tareas	Patrón generación Patrón de Construcción de consenso
	2.2	Aplicar la estrategia	Patrón de Clarificación Patrón generación Patrón reducción Patrón Organización Patrón propagación de conocimiento.
	2.3	Revisar la estrategia	Patrón Evaluación Patrón de Construcción de Consenso
	2.4	Analizar la estrategia	Patrón evaluación

Tabla 19. Asociación de patrones a las subactividades [30].

A continuación, se presenta cada una de las subactividades y su descripción detallada.

Análisis subactividad 1.1.3: Definir las reglas de juego	
Descripción:	En esta estación los equipos de trabajo junto con el docente definen y asumen las reglas del juego.
Actividad relacionada	Establecimiento de mecánicas y dinámicas de juego.
Entradas	<ul style="list-style-type: none"> - Misión previamente diseñada por el docente. - Equipos de trabajo organizados - Guía del equipo seleccionado. - Material para desarrollo de la actividad. - Mecánicas de juego seleccionadas. - Dinámicas de juego seleccionadas.
Resultados	Reglas de juego definidas.
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
patrones	Justificación
Patrón generación	Este patrón se presenta cuando los participantes de los equipos junto con el docente creen las reglas del juego.
Construcción de consenso	En esta actividad se presenta el patrón de consenso porque es necesario que los integrantes del equipo estén de acuerdo y comprometidos con las reglas de juego definidas para lograr los objetivos.

Tabla 20. Descripción de subactividad 1.1 3 Definir las reglas de juego.

Análisis subactividad 1.3.2: Identificación de tareas	
Descripción:	En esta estación los equipos de trabajo identifican las tareas necesarias para ejecutar para la primera ronda.
Actividad relacionada	1.3.1- Entrega de la descripción detallada de la actividad
Entradas	<ul style="list-style-type: none"> - Misión previamente diseñada por el docente. - Equipos de trabajo organizados - Guía del equipo seleccionado - Material para desarrollo de la actividad.
Resultados esperados	<ul style="list-style-type: none"> - Tareas identificadas.

Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
patrones	Justificación
Generación	Mediante este patrón los niños de los equipos identifican las tareas claves que se deben realizar para alcanzar el desarrollo de la misión.

Tabla 21. Descripción de subactividad 1.3.2: Identificación de tareas

Análisis subactividad 1.3.3: Priorización de tareas	
Descripción:	En esta estación los equipos de trabajo priorizan las tareas teniendo en cuenta.
Actividad relacionada	1.3.1- Entrega de la descripción detallada de la actividad
Entradas	<ul style="list-style-type: none"> - Misión previamente diseñada por el docente. - Equipos de trabajo organizados - Guía del equipo seleccionado - Material para desarrollo de la actividad.
Resultados esperados	<ul style="list-style-type: none"> - Tareas identificadas. - Listado de tareas priorizadas.
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
patrones	Justificación
Organización	En esta estación los equipos de trabajo identifican las tareas y deben organizarlas, priorizándolas teniendo en cuenta su criterio particular para la primera ronda.
Evaluación	Se utiliza este patrón porque los integrantes de los equipos deben encontrar un orden lógico que les permita priorizar las tareas.

Tabla 22. Descripción de subactividad 1.3.3: Identificación y priorización de tareas

Análisis subactividad 2.1.1 Definición de la Estrategia	
Descripción:	El Profesor dará al equipo de trabajo un tiempo para que en consenso con sus integrantes organicen una estrategia o una forma para abordar las tareas y su respectivo desarrollo.
Actividad relacionada	subactividad 1.3.2 Identificación y priorización de tareas.
Entradas	Listado de tareas priorizadas.

Resultados esperados	- Estrategia de trabajo.
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
patrones	Justificación
Generación	Este patrón se presenta cuando los participantes de los equipos junto con el docente creen las reglas del juego.
Evaluación	Los integrantes del equipo evalúan diferentes rutas para solucionar las misiones asignadas por el docente.
Construcción y consenso	Este patrón permitirá que los miembros del grupo se comprometan con el mismo objetivo y definan cual será la estrategia para el desarrollo de la misión.

Tabla 23. Descripción de subactividad 2.1.1 Definición de la estrategia.

Análisis subactividad 2.1.2 Definición de tareas	
Descripción:	El Guía del Equipo solicita a sus integrantes la realización de un diseño inicial o bosquejo para que les ayude a definir las tareas a través de las cuales afrontarán la misión.
Actividad relacionada	subactividad 1.1 Entrega de la misión
Entradas	<ul style="list-style-type: none"> - Listado de tareas priorizadas. - Estrategia predefinida.
Resultados esperados	<ul style="list-style-type: none"> - Bosquejo de tareas. - Tablero "Taskboard".
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador
patrones	Justificación
Construcción y consenso	Este patrón permitirá que los miembros del grupo se comprometan con el mismo objetivo y definan la estrategia.
Organización	Este patrón permitirá la organización de las tareas para afrontar el problema planteado.

Tabla 24. Descripción de subactividad 2.1.1 Definición de tareas.

Análisis subactividad 2.2 Aplicar la Estrategia	
Descripción:	<p>Después de establecer la planeación el equipo de trabajo entra en la ejecución de sus tareas orientado por su Guía de Equipo, quien trabaja a la par con sus compañeros. Es en esta estación donde acumulan la mayor cantidad de aportes para su desarrollo y donde mostraran el mayor índice de actividad evidenciando compromiso y participación activa de cada integrante en el Equipo de Trabajo.</p> <p>En esta estación el Equipo de Trabajo se apropia de los conceptos, prácticas ágiles y prácticas colaborativas impartidas por el Profesor, las cuales les ayudaran en el proceso de desarrollo para ejecutar sus tareas de forma más sencilla y rápida, de esta forma logran mejores resultados al momento de entregar la misión.</p>
Actividad relacionada	subactividad 2.1 planear la estrategia.
Entradas	Estrategia de trabajo, bosquejo de tareas, prácticas ágiles y practicas colaborativas.
Resultados esperados	Aplicación de la estrategia evidenciada en el avance del tablero de tareas.
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador
patrones	Justificación
Clarificación	Este patrón permite que los integrantes del equipo logren el entendimiento común de conceptos, practicas ágiles, y colaborativas, manejados por el grupo.
Generación	Este patrón permite que los miembros del equipo realicen sus aportes para solucionar el problema, mientras se sigue la estrategia planteada.
Organización	Este patrón permite que los miembros de equipo vayan ordenando sus actividades encaminadas al logro del desarrollo de la misión.
Propagación de conocimiento	Mediante este patrón se logra que el conocimiento logrado por un niño se transmita a los demás al realizar tareas juntos.

Tabla 25. subactividad 2.2 Aplicar la Estrategia

Análisis subactividad 2.3 Revisar la estrategia	
Descripción:	En esta estación el equipo de trabajo junto con el Profesor verifica las tareas realizadas y el cumplimiento de las mismas, de tal forma que se logre evidenciar en el resultado. Esta estación permite a los integrantes del equipo evaluar el avance de su misión, conocer su ritmo de trabajo, los cuales serán útiles para la planeación de la estrategia en la siguiente Ronda.
Actividad relacionada	Subactividad 2.2 planear la estrategia.
Entradas	<ul style="list-style-type: none"> - Estrategia de trabajo. - Bosquejo de tareas.
Resultados esperados	<ul style="list-style-type: none"> - Evaluación de la estrategia - Estrategia de la siguiente ronda.
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
patrones	Justificación
Evaluación	Mediante este patrón los integrantes del equipo valoran las tareas realizadas, con el objetivo de tomar decisiones respecto a la estrategia.
Construcción y consenso	Este patrón permite que los integrantes analicen la estrategia y planeen la siguiente ronda.

Tabla 26. Descripción de subactividad 2.3 Revisar la estrategia

Análisis subactividad 2.4 Analizar la estrategia	
Descripción:	<p>En esta estación el Equipo de Trabajo y el Guía del Equipo, evalúa su trabajo como equipo, entorno al desempeño de los integrantes, sus aportes y colaboraciones, así como el compromiso que refleje cada uno, determina si la estrategia empleada funciona o no.</p> <p>Es posible que el Equipo de Trabajo deba replantear la forma como están afrontando la tarea para darle una mejor solución y obtener mejores resultados al final de la Misión, se debe analizar si ha sido acertada o porque no ha dado resultado, para así reorganizan la forma de trabajar para la siguiente Ronda. Esta estación permite</p>

	al Equipo de Trabajo y al Guía del Equipo, analizar posibles situaciones futuras y tomar medidas oportunas para mitigar errores o complicaciones potenciales.
Actividad relacionada	subactividad 2.2 planear la estrategia.
Entradas	<ul style="list-style-type: none"> - Estrategia de trabajo. - Bosquejo de tareas.
Resultados esperados	<ul style="list-style-type: none"> - Evaluación de la estrategia. - Estrategia de la siguiente ronda
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador.
Patrones	Justificación
Evaluación	Este patrón permite valorar el trabajo del equipo y el desempeño de los integrantes.
organización	Mediante este patrón los integrantes ordenar las posibles medidas a tomar en situaciones futuras.

Tabla 27. Descripción de Subactividad 2.4 Analizar la estrategia

3.5 Patrones de colaboración propuestos

Como se puede apreciar en la tabla 18, la mayoría de los patrones de colaboración están relacionados con la gestión del conocimiento. Dentro del grupo solo el patrón construcción de consenso está orientado a la organización y comportamiento del equipo. De los estudios de caso realizados, del análisis de asociación de patrones realizados en las sesiones 4.7 y 4.8, y del trabajo con los docentes de aula que han participado en esta investigación, se proponen los siguientes patrones de colaboración presentes en la interacción de los niños:



Figura 22. Patrón liderar



Figura 23. Patrón propagación de conocimiento

Patrón liderar. Este patrón consiste en que un niño del equipo inicia a motivar a sus compañeros y a dirigirlos hacia el logro de los objetivos.

Patrón propagación de conocimiento. El desarrollo de actividades tales como: la revisión entre pares, la programación en parejas, el uso de técnicas como jigsaw, y la colaboración generada entre los equipos, permitió identificar este patrón que permite construir conocimiento y transmitirlo para lograr el alcance de los objetivos del equipo, o para que otros equipos también logren las metas.

Hasta este momento se ha logrado identificar las actividades que requieren trabajo colaborativo, y se han asociado los patrones de colaboración que involucra cada uno de ellas, sin embargo, en los estudios de casos realizado con los niños se pudo apreciar que en la práctica hay algunas actividades que involucran trabajo colaborativo y no han sido especificadas en la descripción del actual modelo Childprogramming-G.

4. Capítulo 4. Modelo ChildProgramming – C

4.1 Mejoras identificadas del modelo ChildProgramming

En los estudios de caso realizados en esta investigación se pudo identificar algunas mejoras al proceso del modelo ChildProgramming. A continuación, se describe cada una de las mejoras y en la sección 4.2 se presenta el modelo final:



Figura 24. Mejoras identificadas al modelo Childprogramming

4.1.1 Sensibilizar a los estudiantes sobre la importancia del trabajo colaborativo.

Para que una actividad sea colaborativa no solo basta con proveer una solución que le permita a un grupo de personas trabajar conjuntamente alrededor de dicha actividad, sino que es necesario incentivar la necesidad de colaborar y determinar las características principales para las cuales la conformación de grupos de trabajo es más productiva, de tal manera que los niveles de colaboración se incrementen [72].

En los estudios de caso realizados se encontró que “los equipos no logran desarrollar todas las características del trabajo colaborativo” [73], así se apreció que la mayoría de los niños se muestran motivados al ejecutar la misión de desarrollo de software propuesta, sin embargo, a pesar de explicar previamente las prácticas colaborativas descritas en el modelo Chilprogramming, algunos grupos no muestran interés por las actividades relacionadas con la selección del nombre del equipo, la elección del líder, la asignación de roles, responsabilidades y la definición de estrategias para resolver el problema entre otros aspectos [74].

Los estudios de caso mostraron debilidades en los procesos colaborativos, por esta razón se propuso la necesidad de estructurar estrategias para sensibilizar a los equipos de niños antes de iniciar un proceso de desarrollo de software. Con la ayuda de un equipo de docentes que participaron a lo largo de la investigación, y mediante grupos de discusión se plantearon estrategias lúdicas para sensibilización a los niños, sobre la importancia de la colaboración en los equipos (ver figura 25). De esta manera se propuso el proceso de sensibilización a los estudiantes antes de iniciar actividades colaborativas. En el anexo E se presenta la aplicación de un conjunto de estrategias para realizar el proceso de sensibilización en los niños.



Figura 25. Estrategias para sensibilizar a los niños sobre la importancia de la colaboración

La figura 26 presenta el proceso de sensibilización propuesto donde se considera importante analizar la población, definir las estrategias de sensibilización, preparar y ejecutar las actividades y posteriormente reflexionar con los equipos de trabajo.

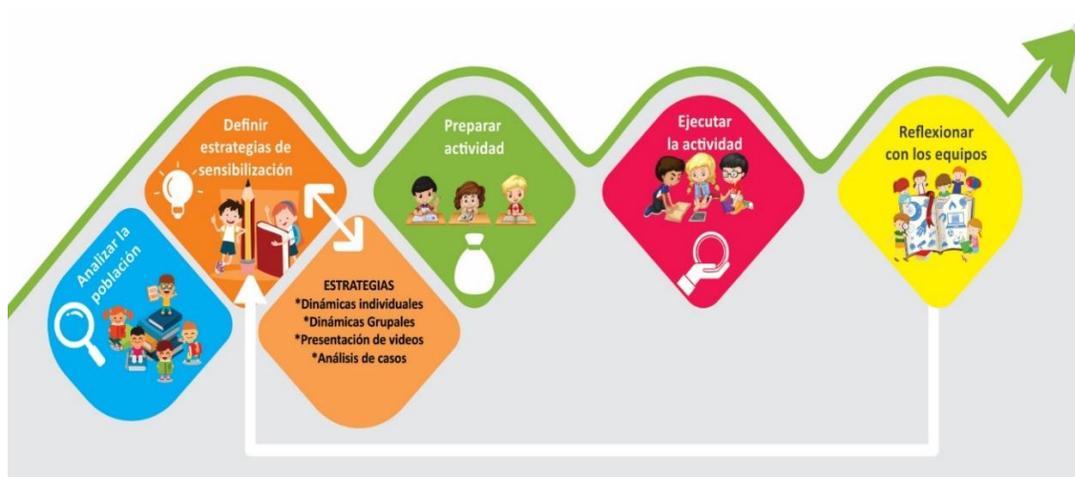


Figura 26. Proceso de sensibilización propuesto

4.1.2 Mejor organización de los equipos de trabajo.

Para mejorar la colaboración es importante la organización de los equipos, de este modo es fundamental conocer la población, definir las precondiciones para los alumnos que permitan especificar la organización de los equipos de trabajo, establecer el tamaño, realizar la designación de roles, elegir el líder, proponer las reglas, los comportamientos deseados dentro de las actividades de desarrollo de software, y elegir el nombre y logotipos de los equipos [69] [75]. Estos aspectos deben ser tenidos en cuenta en el modelo ChildProgramming porque los equipos son el centro del proceso colaborativo y en el actual modelo no se define qué aspectos deben tenerse en cuenta para organizarlos.

4.1.3 Mayor importancia del tablero de tareas

El tablero de tareas Taskboard es uno de los elementos más importantes en la metodología SCRUM [76], y ha sido adaptado al modelo Childprogramming, pues se considera el punto de unión entre todos los integrantes del grupo de trabajo y el Product Owner, además, de ser el eje central de información sobre el estado del proyecto. Este tablero es la estrategia principal para motivar a los niños en la responsabilidad individual, la igual participación y la interdependencia positiva, por lo tanto, se debe orientar mejor al docente en el diseño,

construcción y seguimiento del tablero de tareas. Por lo anterior, el taskboard es el elemento que permite el control y seguimiento del equipo.

4.1.4 Mayor usabilidad del modelo para los usuarios (Docentes).

Mientras que un proceso es un vehículo para hacer un producto, la descripción del proceso es una especificación de cómo se debe llevar a cabo, esta descripción debe ser tan clara que todos puedan instanciarla [77]. La demanda de descripciones de procesos de software sistemático y utilizable es alta [78], y la producción de una descripción de un proceso software usable es esencial, pues los problemas de usabilidad pueden traducirse en dificultades durante la interpretación o promulgación del proceso [77].

En la práctica, las descripciones de los procesos sufren problemas de usabilidad, incluyendo complejidad e inflexibilidad de su estructura, así como información inconclusa e instrucciones inconsistentes [78] [79] [80] de esta manera, al mejorar la calidad de un proceso, se espera también mejorar la calidad de un producto.

Aunque revisando la literatura no se encontraron referentes específicos para evaluar la facilidad de aprendizaje de un modelo de procesos para la enseñanza de la programación de software, en el desarrollo de este trabajo se revisó como principal fuente la norma ISO/IEC 25010, que define la usabilidad como la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones [81], así mismo, se toma la facilidad de aprendizaje como subcaracterística de la usabilidad en la calidad del producto software, más no, de procesos de desarrollo de software.

Dentro de las subcaracterísticas de la usabilidad, la capacidad de aprendizaje ofrece al usuario la facilidad de aprender la aplicación de un producto [82], en este caso se busca adaptar el modelo Childprogramming como un producto, y evaluar su facilidad de aprendizaje proponiendo un conjunto de atributos y métricas, relacionadas con la familiaridad, el lenguaje común, y la consistencia de Childprogramming.

El proceso de software es importante para el éxito de los proyectos [83] y debe ser utilizable porque: ayuda a los intérpretes a entender, comunicar y ejecutar el proceso, facilita el seguimiento y control del proceso, permite la evolución y reutilización, apoya la gestión, evaluación y mejora del proceso, proporcionando una base para la construcción de una cultura de Ingeniería de Software [84][85][80][77].

En el anexo F se presenta la evaluación de la capacidad de aprendizaje que ofrece el modelo Childprogramming como una de las subcaracterísticas de la usabilidad propuestas en la norma ISO/IEC 25010. Para dicha evaluación se definieron métricas relacionadas con la familiaridad, el lenguaje común, la consistencia y la intuición, que a pesar de ser expuestas para la evaluación de productos software, también pueden ser adaptadas para evaluar un modelo de procesos, con el objetivo de identificar posibles mejoras en la estructuración y descripción del modelo ChildProgramming. A continuación, se presentan el análisis de 23 técnicas colaborativas de aprendizaje que podrían utilizarse en el modelo Childprogramming.

4.1.4.1 Técnicas colaborativas de aprendizaje para apoyar el proceso ChildProgrammig –C.

Una técnica colaborativa de aprendizaje es una estrategia de aprendizaje para ser trabajada en equipos. En el anexo G se presenta un conjunto de 23 técnicas colaborativas de aprendizaje [86], que han sido evaluadas para encontrar las más acordes para apoyar el proceso colaborativo de Childprogramming. Las técnicas colaborativas de aprendizaje pueden clasificarse en cuatro categorías, como se muestra la figura 27.



Figura 27. Técnicas de colaboración para apoyar el proceso Childprogrammig –C basada de [86] .

Cada una de las 23 técnicas de colaboración de aprendizaje fue evaluada por un equipo de cuatro docentes, quienes han participado en el desarrollo del modelo Childprogramming a lo largo de la investigación, con el objetivo de analizar si dichas técnicas podrían incorporarse al proceso de desarrollo de software. Las técnicas fueron valoradas de la siguiente manera:

- color verde: si es muy aplicable al modelo Childprogramming
- Color naranja: si su aplicación es media al modelo Chilprogramming
- Color rojo: si su aplicación es baja al modelo Chilprogramming.

En las tablas 28-31 se presenta el análisis realizado por el equipo evaluador de docentes.

TECNICAS COLABORATIVAS DE DISCUSIÓN				
Técnica	Propósito en los estudiantes:			
1: Think-Pair-Share (Pensar para compartir)	Piensan individualmente durante unos minutos, y luego discuta y compare sus respuestas con un compañero antes de hablar con toda la clase.	x		
2: Round Robin	Generan ideas y hablan en orden, pasando de un estudiante al siguiente.	x		
3: Buzz Groups	Discuten informalmente preguntas relacionadas con el curso en pequeños grupos de compañeros		x	
4: Talking Chips	Participan en una discusión grupal y entrega una ficha cada vez que hablen.		x	
5: Three-Step Interview	Logran Entrevistarse con un par e informar lo que aprenden.		x	
6: Debates críticos	Asumen y discuten un problema con un compañero, que está en oposición a sus puntos de vista personales.		x	

Tabla 28. Técnicas colaborativas de discusión

TÉCNICAS COLABORATIVAS DE ENSEÑANZA RECÍPROCA ENTRE COMPAÑEROS				
Técnica	Propósito			
7: Tomar notas	Agrupan la información de sus notas individuales para crear una versión mejorada con la de su compañero.			x
8: Celda de aprendizaje	Aprenden a cuestionarse unos a otros utilizando preguntas que hayan desarrollado individualmente sobre una tarea de lectura u otra actividad de aprendizaje.			x

9: Pecera Fishbowl	Se organizan grupos concéntricos los grupos exteriores escuchan y observan las discusiones de los grupos concéntricos más pequeño.			x
10: Juego de rol	Asumir una identidad diferente y representar un escenario.		x	
11: Jigsaw (Rompecabezas)	Desarrollar el conocimiento de un tema determinado y luego se lo enseñó a otros.	x		
12: Equipos de prueba.	Prepararse para una prueba en grupos de trabajo, tomar la prueba individualmente y luego retomar la prueba en sus grupos		x	

Tabla 29. Técnicas de enseñanza recíproca entre compañeros

TÉCNICAS COLABORATIVAS PARA RESOLVER PROBLEMAS				
Técnica	Propósito			
13: Solución de problemas de pensar en voz alta	Resolver problemas en voz alta para probar su razonamiento	x		
14: Enviar un problema	Tratar de resolver un problema como un grupo, y luego pasar el problema y la solución a un grupo cercano quienes harán lo mismo; el grupo final evaluará la solución.		x	
15:Caso de estudio	Revisar un estudio escrito de un escenario del mundo real y desarrollar una solución al problema presentado en el caso.	x		
16:Solución estructurada de Problemas	Seguir un formato estructurado para resolver problemas.		x	
17:Equipos analíticos	Asumir roles y tareas específicas para analizar de forma crítica actividades como: escuchar una conferencia o ver un video.	x		
18: Grupos de Investigación	Planear, conducir, e informar sobre proyectos de investigación en profundidad		x	

Tabla 30. Técnicas colaborativas para resolver problemas

TÉCNICAS COLABORATIVAS DE ORGANIZACIÓN GRÁFICA					
Técnica	Es una técnica en la cual los estudiantes:				
19: Afinidad grupal	Generar ideas, identificar temas en común, y luego ordenar y organizar las ideas en consecuencia.		x		
20: cuadrícula de grupo	Se les da información y se les pide que los coloquen en las celdas en blanco de una cuadrícula según las rubricas de categoría.		x		
21: Matriz de equipo	Discriminar entre conceptos similares al notar y marcar en un gráfico la presencia o ausencia de características importantes y definitorias.				X
22: Cadenas de Secuencia	Analizar y representar gráficamente una serie de eventos, acciones, roles o decisiones.		x		
23: Webs de Palabras	Generar una lista de ideas y luego organizarlas en una relación gráfica, utilizando líneas o flechas para representar la conexión.	x			

Tabla 31. Técnicas colaborativas de organización gráfica

Como resultado del anterior análisis y la identificación de la relación de los thinklets y las técnicas colaborativas, en este estudio se propone la posibilidad de integrar las técnicas colaborativas de aprendizaje evaluadas en las tablas 28 al 31 a la Metodología para el Desarrollo de Procesos Colaborativos propuesta por Kolfschoten and G.-J. Vreede.

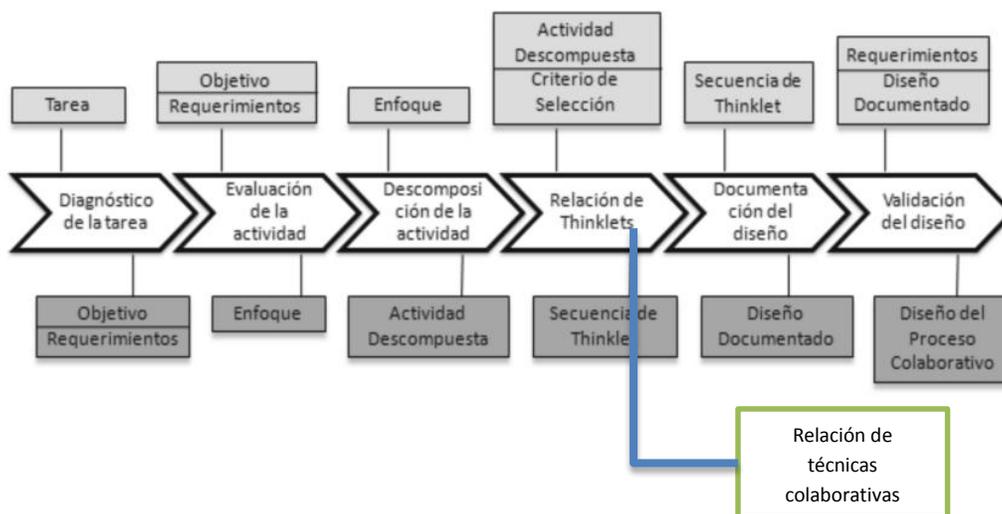


Figura 32. Propuesta de Incorporación de técnicas colaborativas a la Metodología para el Desarrollo de proceso Colaborativos

Como se aprecia en la figura 32, se propone la asociación de las técnicas colaborativas de discusión, organización gráfica, enseñanza recíproca entre compañeros, y resolución de problemas, como herramientas para que los procesos colaborativos puedan describirse más ampliamente.

4.2 Generación de thinklets (Childthinklets)

Al analizar las actividades del modelo Childprogrammig y las técnicas colaborativas se han detectado algunos thinklets para niños que pueden ser útiles en la Ingeniería de la Colaboración, estos son: *Analysis in Pairs*, *Divide et impera*. A continuación, se presenta la descripción de los thinklets propuestos.

Nombre del thinklet	Analysis in Pairs (Análisis en parejas)
Descripción general:	Este thinklet se basa en la técnica colaborativa de discusión Think-Pair-Share (Pensar para compartir), donde los participantes analizan individualmente los problemas durante unos minutos, y luego discuten, comparten y comparan sus respuestas con un compañero antes de hablar con el resto del equipo.
Escoger este thinklet:	<ul style="list-style-type: none"> • Cuando sea necesario realizar una discusión sobre un tema particular en parejas, para posteriormente socializarlo con el grupo. • Cuando se necesita que los miembros del grupo generen discusión con base en información generada por otro miembro del grupo. • Cuando se quiera la participación de todos los integrantes del grupo, pues el análisis en parejas conlleva la igual participación de los integrantes.
Entradas	Un conjunto de temas que serán discutidos por el grupo.
Salidas	Conjunto de ideas, opiniones, soluciones de los temas discutidos.
Ejemplo de uso:	El desarrollo de software conlleva a que los niños planteen soluciones mentales que son compartidos con sus compañeros para plasmarlos en algoritmos. En el caso de Childprogramming-C este thinklet es recurrente durante la actividad “desarrollo de la misión”, donde al desarrollar la aplicación software, es necesario la programación por parejas como estrategia de las metodologías ágiles de desarrollo.

Patrones asociados	Patrón generación y propagación de conocimiento.
--------------------	--

Tabla 33. Childthinklet Analysis in Pairs

Nombre del thinklet	Divide et impera (Divide y vencerás)
Descripción general:	Este thinklet se basa en la técnica colaborativa de resolución de problemas: <i>structured Problem Solving</i> , donde los niños al dividir los procesos de un problema en pasos manejables, no se sienten abrumados y aprenden a identificar, analizar, y resolver problemas de una manera organizada. En este thinklet es fundamental el rol del líder del equipo, porque es uno de los principales responsables de la división y asignación de tareas.
Escoger este thinklet:	<ul style="list-style-type: none"> • Cuando sea necesario resolver problemas que pueden dividirse. • Cuando se desee asignar responsabilidades a cada uno de los integrantes del grupo. • Cuando se quiera la participación de todos los integrantes del grupo.
Entradas	Un conjunto de temas que serán discutidos por el grupo.
Salidas	Conjunto de ideas, y soluciones de los temas discutidos.
Ejemplo de uso:	En el caso de Childprogramming-C, este thinklet es recurrente durante en las actividades “planear estrategia” y “aplicar estrategia”, donde al planear la estrategia los niños descubren la importancia de dividir estructuralmente los problemas para llegar a la solución. De igual forma este patrón es evidente en el tablero de asignación de tareas, donde la solución del problema se divide y se asignan responsables.
Patrones asociados	Organización.

Tabla 34. Childthinklet Divide et impera

Un Proceso Colaborativo (Collaborative Process) es un conjunto ordenado de Tareas (ChildLet Tasks) con los Conceptos (Concepts), Roles (Role), Prácticas Ágiles (Agile Practice) y Thinklets relacionados, los cuales a su vez son especificaciones de un elemento más abstracto conocido como ChildLetElement. El proceso colaborativo está compuesto por Patrones Colaborativos (Collaborative Patterns) encapsulados a través de Thinklets, que a su vez describen la dinámica de una Tarea (Childlet Task). El proceso colaborativo, es aplicado por equipos (Team) conformados por niños (Child) quienes aplican las tareas de acuerdo a las especificaciones del proceso [44].

Las tareas (GmentorTask) son agregadas o creadas por el tutor o mentor y son desarrolladas por los grupos de niños. Estas tareas van acompañadas de una Estrategia de Gamificación (StrategyGamification) la cual contiene los elementos que hacen parte de la gamificación (GElements) y que forman parte del principal agregado de la versión gamificada de Childprogramming. Los elementos de la estrategia de gamificación son las mecánicas de juego (GMecanica), las dinámicas de juego (GDinamica) y los tipos de jugadores (GJugadores) que nos dan un diagnóstico inicial de cómo podría ser la estrategia de gamificación [44].

Es importante tener una visión de lo que puede ser posible con la ayuda de la tecnología y de la clase de investigación que necesita ser conducida, ya que para tener una actividad colaborativa que genere aprendizaje en sus participantes es necesario ir más allá de aceptar y disponer un conjunto de prácticas de clase, laboratorios y las respectivas herramientas. Se debe tener en cuenta un enfoque más profundo para garantizar una colaboración entre los equipos de trabajo, y así mismo un aprendizaje común e igualitario por medio del análisis de todos los factores externos que se deben involucrar en dicho proceso[87].

En la figura 30 se presenta la propuesta del modelo Childprogramming-C donde se puede apreciar que el componente lúdico, ágil y colaborativo sigue siendo la base fundamental del modelo:

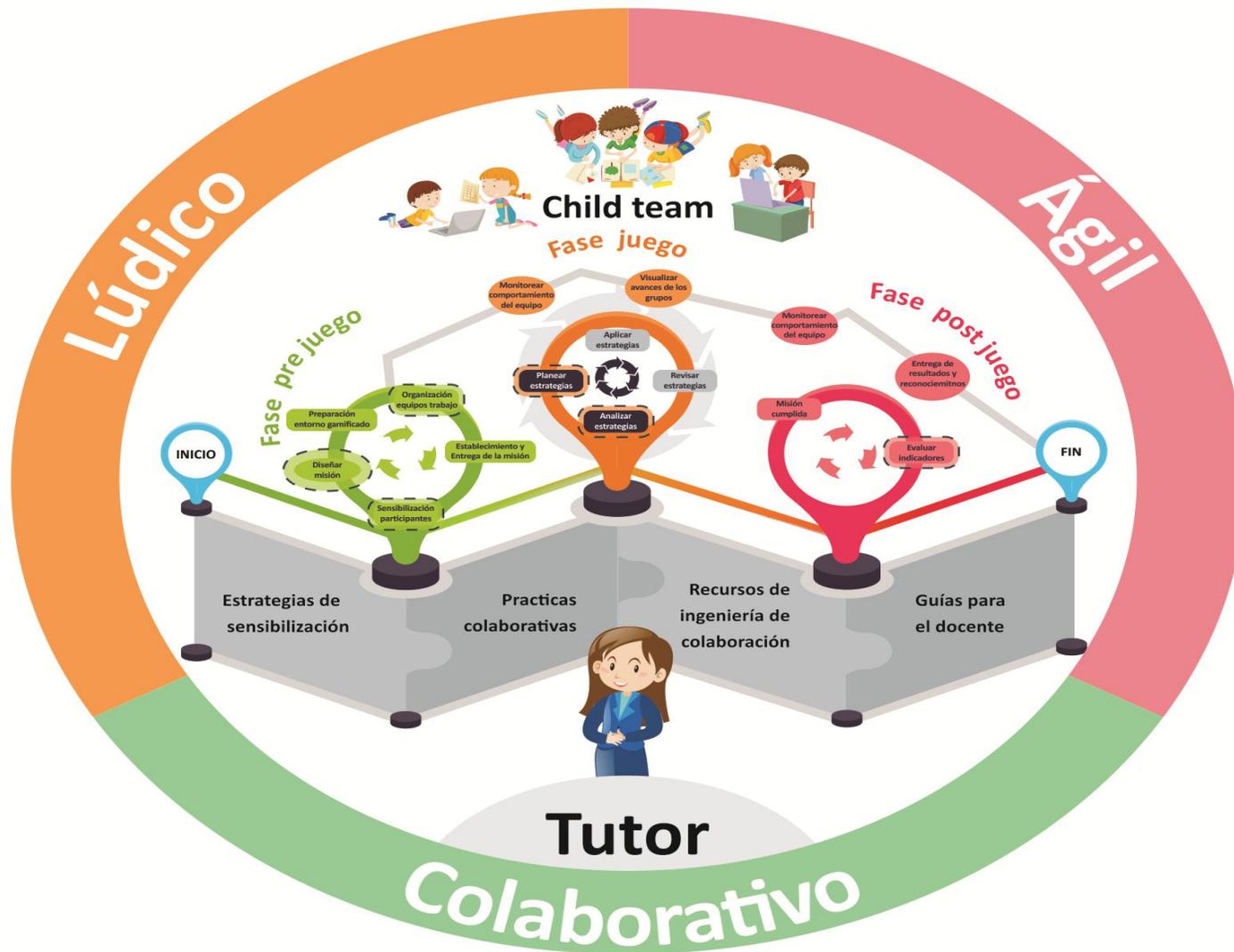


Figura 29 Modelo ChildProgramming –C. fuente propia

Como se puede apreciar en la figura 30, el modelo ChildProgramming-C cuenta con el componente ágil, lúdico y colaborativo inicial, mantiene las fases de prejuego, juego y postjuego, mejorando el componente colaborativo en algunas de las actividades, además se incorpora nuevas actividades tales como: “sensibilización de los participantes” y “organización de los equipos”.

Considerando que el aprendizaje colaborativo efectivo depende de factores sociales sutiles y estructuración pedagógica, no solo simple tareas y tecnologías [88]. El modelo también sugiere un conjunto de estrategias de sensibilización, prácticas colaborativas y recursos de colaboración, que ayudan a promover la colaboración dentro de un proceso de desarrollo de software para modelar, descomponer problemas y encontrar soluciones de forma algorítmica. A continuación, se describen cada uno de los componentes del modelo:

4.3.2 Actores del Modelo ChildProgramming

Los actores en Childprogramming son los encargados de llevar a cabo el proceso de desarrollo de software, el modelo propone tres actores: el profesor quien es el encargado de proponer la misión y guiar el proceso, los estudiantes quienes se encargan de realizar las tareas de desarrollo de software, y el investigador quien se encarga de observar las interacciones de los equipos para mejorar el proceso.

Childprogramming se compone de los siguientes actores:

Actores del Modelo ChildProgramming	
	<p>Profesor o tutor: encargado de transmitir el conocimiento teórico y definir las actividades prácticas que realizarán los niños en sus equipos de trabajo durante el transcurso del proyecto. En ChildProgramming-C el docente juega un papel muy importante como dinamizador de las estrategias para sensibilizar a los estudiantes de la importancia de colaborar al ser parte de un equipo de desarrollo de software; del mismo modo el profesor es quien realiza el control del proceso colaborativo y propone actividades que permiten el desarrollo del pensamiento computacional y la solución de problemas.</p>

	<p>Estudiantes o alumnos (Niños): participantes fundamentales, fuente principal de información y ejecutores de las actividades propuestas por el profesor. En ChildProgramming-C se busca que el estudiante no solo busque ganar los retos de programación, sino también desarrollar el pensamiento computacional al compartir con sus compañeros.</p>
	<p>Observadores (Investigador): personas encargadas de apoyar el proceso en cualquier momento considerando las disposiciones del profesor, y observando de manera detallada del desarrollo de los niños en sus equipos para cada actividad propuesta.</p>

Tabla 35. Actores del Modelo ChildProgramming –C

4.3.3 Descripción de las fases y actividades del modelo

A continuación se presentan cada una las fases del modelo Childprogramming – C:

FASES	ACTIVIDADES CHILDPROGRAMMING – C
<p>PREJUEGO</p>	<p>El propósito de esta fase además de preparar el entorno gamificado es establecer y entregar el objetivo de la misión, es también sensibilizar a los niños sobre su participación en los equipos de trabajo. Entre las actividades de esta fase se tiene:</p> <p>Preparar el entorno gamificado. Preparar el entorno gamificado es una actividad propuesta por ChildProgramming-G, para esto hay que realizar varias tareas las cuales son: “Establecimiento de las mecánicas y las dinámicas de juego” que se van a usar en el desarrollo de las actividades de trabajo en los grupos de niños, configuración de GamiTool (herramienta de apoyo opcional) y Definir las reglas de juego. Con estas tareas y con la entrega de la misión se provee un punto de partida para gamificar el aula de clase y las tareas que van a realizar los grupos de niños.</p> <p>Sensibilizar a los participantes: desde el modelo Childprogramming-C se propone la actividad sensibilizar a los participantes, la cual consiste en realizar actividades estructuradas que sensibilicen a los niños sobre la importancia del trabajo colaborativo y la responsabilidad al ser parte de un equipo, para esto el modelo ofrece un conjunto estrategias que permiten motivar a los niños al trabajo colaborativo en el anexo E.</p>

	<p>Diseñar la misión: el diseño de la misión debe tener en cuenta las características de los participantes y definir los criterios de evaluación de la calidad de los productos.</p> <p>Organizar los equipos de trabajo: Jhonson plantea el uso de grupos pequeños, de al menos dos personas para poder aplicar el aprendizaje colaborativo [89], algunos autores indican la efectividad del aprendizaje colaborativo se realiza en grupos pequeños, que contienen entre dos a cinco miembros; sin embargo, cada una de las técnicas tiene su propio valor mínimo y máximo [88]. En última instancia es potestad del docente definir este valor según sea el número de estudiantes, la mecánica de trabajo y el tiempo que dispone para realizar la actividad, formando preferiblemente grupos seleccionados por él y que sean heterogéneos permitiendo que los alumnos tengan acceso a diversas perspectivas y métodos de resolución de problemas, y produciendo un mayor desequilibrio cognitivo, necesario para estimular el aprendizaje y desarrollar la responsabilidad individual [90].</p> <p>En la organización de los equipos de trabajo es importante tener en cuenta aspectos tales como la edad, sexo, conocimientos previos, y aspectos del contexto externo. En esta actividad es fundamental orientar a los niños en la elección del líder del equipo, la asignación de nombre e identificadores al equipo y la definición de reglas.</p>
<p>JUEGO</p>	<p>En esta fase las iteraciones (Rondas) del modelo ChildProgramming continúan igual con el mismo propósito u objetivo: cumplir con la misión y dejar listo el entregable que garantiza el cumplimiento de los objetivos propuestos para la actividad.</p> <p>La ronda se sigue describiendo igual que en ChildProgramming y continúa incluyendo cuatro estaciones a seguir, las cuales son: planear la estrategia, aplicar la estrategia, revisar la estrategia y analizar la estrategia. Las estaciones planear la estrategia y analizar la estrategia se actualizan y las otras dos estaciones se dejan como estaban planteadas en el modelo inicial.</p> <p>En la estación planear la estrategia, la definición de la estrategia permitirá al Profesor dar al equipo de trabajo un tiempo para que en consenso con sus integrantes organicen una estrategia o una forma para abordar las tareas y su</p>

Tabla 36. Descripción del modelo Childprogramming-C

	<p>respectivo desarrollo, teniendo en cuenta la importancia del trabajo en equipo y la responsabilidad de cada uno de los niños.</p> <p>En la definición de las tareas el guía del equipo solicita a sus integrantes la realización de un diseño inicial o bosquejo para que les ayude a definir las tareas a través de las cuales afrontarán la misión. En esta actividad es fundamental dar más importancia al tablero taskboard pues se considera una herramienta clave para dirigir y coordinar al equipo de trabajo, además de realizar la asignación de tareas y responsabilidades asignadas en dicho tablero.</p> <p>En esta fase el diseño del tablero taskboard por parte de los integrantes del equipo, se convierte en una fundamental actividad para lograr una verdadera interdependencia positiva, igual participación y responsabilidad individual de los participantes.</p> <p>Después de establecer la planeación, el equipo de trabajo entra en la actividad aplicar la estrategia la cual incluye la ejecución de las tareas, orientado por su guía de equipo, quien trabaja a la par con sus compañeros. Es en esta estación donde acumulan la mayor cantidad de aportes para su desarrollo y donde mostrarán el mayor índice de actividad evidenciando compromiso y participación activa de cada integrante en el equipo de trabajo. En esta estación el equipo de trabajo se apropia de los conceptos, prácticas ágiles y prácticas colaborativas impartidas por el profesor, las cuales les ayudarán en el proceso de desarrollo para ejecutar sus tareas de forma más sencilla y rápida, de esta forma lograrán mejores resultados al momento de entregar la misión.</p> <p>En la actividad revisar la estrategia también se propone revisar la calidad del producto software verificando los criterios y evaluación predefinidos.</p> <p>Las actividades monitorear el comportamiento de los equipos y visualizar actividades no se modifican.</p>
<p>POSTJUEGO</p>	<p>Esta fase conserva el objetivo del modelo inicial, el cual es entregar el producto software realizado. La misión cumplida corresponde a la solución completamente implementada acompañada de todo el material asociado al desarrollo de la actividad. El equipo de trabajo y el guía del equipo entregan la misión cumplida al profesor. El profesor al cierre de esta misión, debe evaluar que el trabajo del equipo refleje un proceso de aprendizaje y que haya permitido cumplir con los</p>

	objetivos de aprendizaje propuestos de forma efectiva. En esta fase Childprogramming-C hace su aporte e incluye otra actividad que es la evaluación del producto software colaborativamente y no únicamente por parte del profesor.
--	---

4.3.4 Prácticas propuestas para childprogramming-C

El modelo Childprogramming inicial propone un conjunto de prácticas colaborativas dirigidas al desempeño de los hacia los niños. Childprogramming-C propone un conjunto de prácticas que conllevan al mejoramiento de la colaboración, pero estas deben ser incentivadas por los docentes. Las propuestas son las siguientes:

Nombre de la práctica:	Sensibilizar a los estudiantes sobre la importancia de la colaboración
	DESCRIPCIÓN: Los docentes pueden utilizar diferentes estrategias para que los niños comprendan la importancia de aprender a colaborar, antes de iniciar el proceso de desarrollo de software.
Responsable:	Docente
Nombre de la práctica:	Enfatizar en la importancia del taskboard
	DESCRIPCIÓN: El tablero taskboard se convierte en un instrumento que controla la asignación de tareas y la responsabilidad de los niños, por esta razón, es una herramienta que debe tomar mucha importancia durante todo el proceso de resolución de problemas de forma algorítmica.
Responsable:	Docente
Nombre de la práctica:	Promover la importancia de "colaborar para aprender"

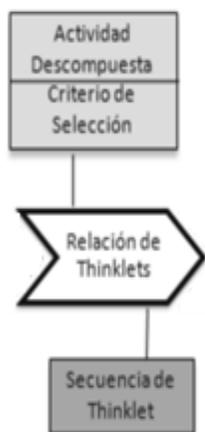
	<p>DESCRIPCIÓN:</p> <p>El docente es el principal encargado de promover en los estudiantes el deseo de colaborar para aprender nuevos conceptos y habilidades en el desarrollo de habilidades computacionales.</p>
<p>Nombre de la práctica:</p>	<p>Promover la importancia del desarrollo del software para el desarrollo del pensamiento computacional</p>
	<p>DESCRIPCIÓN:</p> <p>El docente también cumple la misión de promover en los equipos de trabajo la importancia de la programación de software en el desarrollo del pensamiento computacional.</p>
<p>Responsable:</p>	<p>Docente</p>
<p>Nombre de la práctica:</p>	<p>Proponer retos más libres</p>
	<p>DESCRIPCIÓN:</p> <p>Los retos o misiones que propone el docente para ser desarrollado por los estudiantes no deben limitar los intereses de los mismos. De las experiencias realizadas se observó que los equipos trabajan más colaborativamente cuándo se proponen actividades de programación poco restringidas.</p>
<p>Responsable:</p>	<p>Docente</p>
<p>Nombre de la práctica:</p>	<p>Gamificar la colaboración en el curso</p>
	<p>DESCRIPCIÓN:</p> <p>La colaboración puede pasar los límites de sus equipos de trabajo, cuando mediante la gamificación se motivan los equipos más avanzados a ayudar a los equipos que presentan dificultades en el desarrollo del software.</p>
<p>Responsable:</p>	<p>Docente</p>

Tabla 37. Descripción del modelo Childprogramming-C

4.3.5 Modelo ChildProgramming desde la ingeniería de la colaboración

Una vez definido el modelo Childprogramming-C, en esta sección se presenta el modelamiento desde la Ingeniería de la Colaboración, para ello se aplica la fase 4, fase 5 y fase 6 de la metodología para el desarrollo de procesos colaborativos.

4.3.5.1 Relación de Thinklets



La relación de Thinklets hace parte de la fase 4 de la metodología para el desarrollo de actividades colaborativas. En esta fase son asociados los Thinklets a las subactividades colaborativas que componen el proceso objeto de estudio. Los Thinklets identificados deben adecuarse a los procesos que conforman las subactividades, a los recursos, al grupo y hasta a las propias habilidades de las personas involucradas en la ejecución de los procesos colaborativos [52].

En la tabla 36 se presenta la asociación de los thinklets a cada una de las subactividades colaborativas del proceso de ChildProgramming. De igual manera se relacionan algunas técnicas de aprendizaje colaborativo que pueden ayudar al docente en el desarrollo de las subactividades.

Subactividad: 1.1.3 Definir las reglas de Juego		
Patrones	Thinklets	Justificación de selección del thinklets
Generación	FreeBrainstorm	Este thinklet permite que los niños participantes de los equipos intercambien ideas para generar las reglas de juego del equipo.
Construcción de consenso	<i>PointCounterPoint</i>	Este thinklet ayuda a encontrar un punto común entre las diferentes reglas de juego que propongan los integrantes del grupo.
técnicas de aprendizaje colaborativo sugeridas: <i>Think-Pair-Share</i> , <i>Talking Chips</i> , <i>debates críticos</i> .		
Subactividad: 1.3.2 Identificación de tareas		
Patrones	Thinklets	Justificación de selección del thinklets

Patrón generación	FreeBrainstorm	Este thinklet permite que los niños propongan las tareas que debe realizar el grupo para posteriormente priorizarlas.
técnicas de aprendizaje colaborativo sugeridas: <i>Solución de problemas, think aloud pair y analytic teams.</i>		
Subactividad: 1.3.3 Priorización de tareas		
Patrones	Thinklets	Justificación de selección del thinklets
Organización	<i>RichRelations</i>	Permita que los participantes organicen las tareas teniendo en cuenta las prioridades según categorías y orden definido.
Evaluación	<i>StrawPoll</i>	Los participantes son instruidos para calificar un conjunto de contribuciones, cada ítem en una escala designada usando los criterios designados.
Técnicas de aprendizaje colaborativo sugeridas: <i>Webs de Palabras, debates críticos.</i>		
Subactividad: 2.1.1 Definición de estrategia		
Patrones	Thinklets	Justificación de selección del thinklets
Patrón generación	FreeBrainstorm	Mediante este thinklet los niños proponen las ideas necesarias para construir la estrategia con la cual desarrollaran la misión de programación propuesta por el docente.
Patrón de organización	<i>RichRelations</i> <i>Divide et impera</i>	Permita que los participantes organicen las tareas teniendo en cuenta las prioridades según las categorías y orden definido.
Técnicas de aprendizaje colaborativo sugeridas: <i>debates críticos, think-pair-Share.</i>		
Subactividad: 2.1.1 Definición de tareas		
Patrones	Thinklets	Justificación de selección del thinklets
Construcción y consenso	FreeBrainstorm	Mediante este thinklet los niños proponen las ideas necesarias para construir la estrategia con la cual desarrollaran la misión de programación propuesta por el docente.
Organización	<i>RichRelations</i>	Permita que los participantes organicen las tareas teniendo en cuenta las prioridades según las categorías y orden definido.
Análisis subactividad 2.1.2 Definición de tareas		
Descripción	El Guía del equipo solicita a sus integrantes la realización de un diseño inicial o bosquejo para que les ayude a definir las tareas a través de las cuales afrontarán la misión.	
Actividad relacionada	subactividad 1.1 Entrega de la misión	
Entradas	- Listado de tareas priorizadas.	

	- Estrategia predefinida.	
Resultados esperados	- Bosquejo de tareas. - Tablero "Taskboard".	
Participantes	Equipo de trabajo, guía del equipo, profesor, observador investigador	
patrones	Justificación	
Construcción y consenso	Este patrón permitirá que los miembros del grupo se comprometan con el mismo objetivo y definan las tareas.	
Organización	Este patrón permitirá la organización de las tareas para afrontar el problema planteado.	
Subactividad: 2.2 Aplicar la Estrategia		
Patrones	Thinklets	Justificación de selección del thinklets
Clarificación	<i>Pin the tail on the donkey</i>	Permite que los participantes realicen una mayor deliberación de conceptos importantes y necesarios para el desarrollo de la estrategia.
Generación.	<i>FreeBrainstorm</i>	Mediante este thinklet los niños proponen las ideas necesarias para desarrollar la estrategia y lograr la misión de programación propuesta por el docente.
Reducción	<i>BroomWagon</i>	Este thinklet permite que el equipo seleccione las ideas más acordes para lograr la misión.
Organización	<i>RichRelations</i> <i>Divide et impera</i>	Permita que los participantes desde sus diferentes roles, organicen las tareas para desarrollar la estrategia planteada.
Patrón propagación de conocimiento	<i>share knowledge</i> <i>Analysis in Pairs</i>	Mediante este thinklet los niños comparten soluciones y transmiten sus conocimientos a sus compañeros de equipo para buscar solución a los retos. Este thinklet juega un papel muy importante cuando los niños programan por parejas.
Técnicas de aprendizaje colaborativo sugeridas: <i>buzz groups</i> , <i>analytic teams</i> y <i>think-pair-share</i> .		
Subactividad: 2.3 Revisar la estrategia		
Patrones	Thinklets	Justificación de selección del thinklets
Evaluación	<i>StrawPoll</i>	Los participantes son instruidos para calificar la estrategia utilizada, si realmente se está logrando alcanzar el objetivo.
Construcción de consenso	<i>MoodRing</i>	Rastrear continuamente el nivel de consenso dentro del grupo con respecto al tema actualmente en discusión.
	<i>PointCounterPoint</i>	Ayuda a encontrar un punto común entre diferentes puntos de vista de los integrantes del grupo.

Técnicas de aprendizaje colaborativo sugeridas: <i>analytic teams</i> y <i>think aloud pair</i> .		
Subactividad 2.4 : Analizar la estrategia		
Patrones	Thinklets	Justificación de selección del thinklets
Evaluación	<i>StrawPoll</i>	Los participantes son instruidos para calificar la estrategia utilizada, si realmente se está alcanzando el objetivo o si es necesario cambiar de estrategia.
Organización	<i>RichRelations</i>	Permita que los participantes desde sus diferentes roles, asuman con responsabilidad sus tareas para la siguiente ronda.
Técnicas de aprendizaje colaborativo sugeridas:		<i>analytic teams</i> y <i>think aloud pair</i> , <i>word webs</i> .

Tabla 38. Relación de thinklets subactividad 2.4

FASE 5: a partir de la información obtenida en las fases anteriores, en esta fase deben generarse los siguientes elementos definidos en Ingeniería de la Colaboración [8]: la agenda detallada y modelo de facilitación del proceso.

4.3.5.2 Agenda detallada.

Presenta de manera clara y detallada la información relacionada con cada actividad que hace parte del proceso del diseño colaborativo. Los tiempos estimados fueron tomados de acuerdo a los casos de estudio realizados.

N°	Actividad	Entregable	Patrón y Thinklet	Tiempo estimado	participantes
Actividad: Prejuego					
Subactividad: Preparar el entorno gamificado.					
1.1.3	Definir las reglas de Juego.	Reglas de juego definidas para el desarrollo de la misión, teniendo en cuenta las dinámicas y mecánicas de gamificación.	Patrón generación. <i>Thinklet:</i> <i>FreeBrainstorm</i> Patrón Construcción de consenso <i>Thinklet:</i> <i>PointCounterPoint</i>	15	Equipos de niños
Subactividad: Establecer la misión					

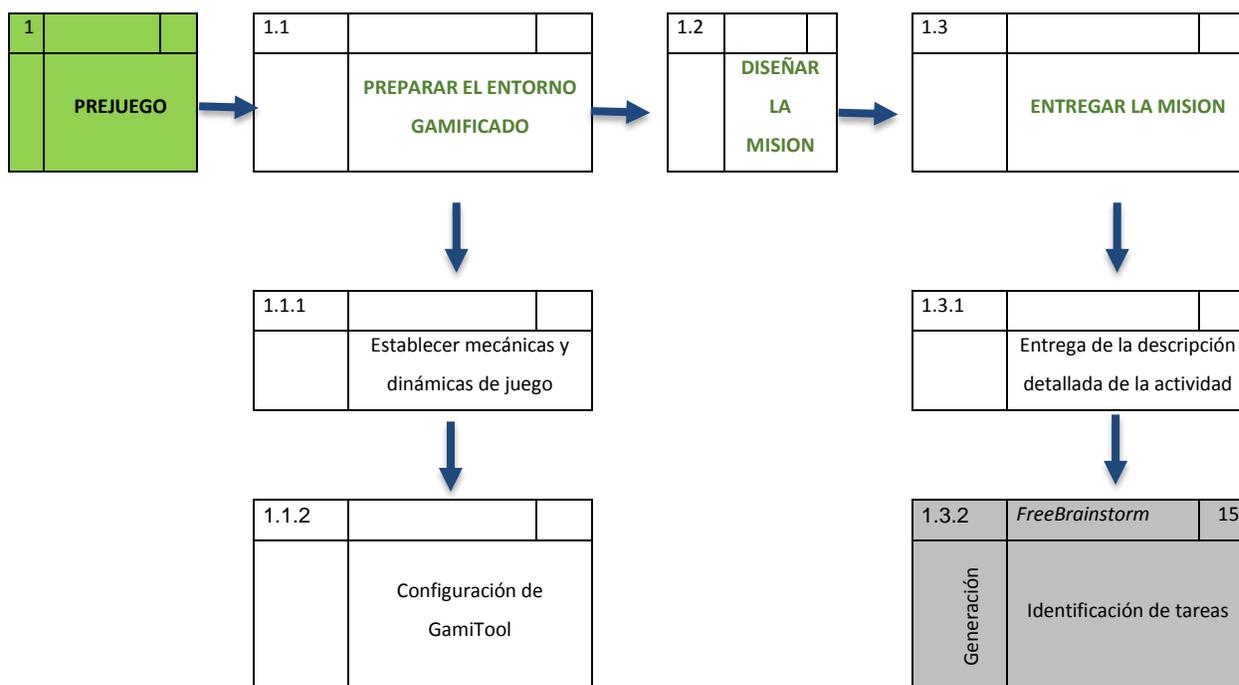
1.3.2	Identificación de tareas.	Tareas identificadas.	a) Patrón generación <i>Thinklet: FreeBrainstorm</i>	15	Equipos de niños – Asesoría docente
1.3.3	Priorización de tareas.	Listado de tareas priorizadas.	a) Patrón organización <i>Thinklet: RichRelations</i>	10	Equipos de niños – Asesoría docente
			b) Patrón evaluación <i>Thinklet: StrawPoll</i>		
Subactividad: Entregar la misión					
1.3.4	Definición de estrategia	Estrategia de trabajo.	Patrón generación. <i>Thinklet FreeBrainstorm</i>	20	Equipos de niños
			Patrón de organización. <i>Thinklet RichRelations</i>		
1.3.	Definición de tareas.	Bosquejo de tareas.	Patrón generación. <i>Thinklet FreeBrainstorm</i>	20	Equipos de niños
		Tablero "Taskboard".	Patrón de Construcción de consenso. <i>Thinklet: PointCounterPoint</i>		
2.2	Aplicar la Estrategia.	Aplicación de la estrategia evidenciada en el avance del tablero de tareas.	Patrón de Clarificación. <i>Thinklet: Pin the tail on the donkey</i>	60 (este tiempo depende de la misión que se desee desarrollar)	Equipos de niños
			Patrón generación. <i>Thinklet: FreeBrainstorm</i>		
			Patrón reducción <i>Thinklet: BroonWagon</i>		
			Patrón Organización <i>Thinklet: RichRelations</i>		
			Patrón propagación de conocimiento. <i>Thinklet: share knowledge</i>		
2.3	Revisar la Estrategia	Evaluación de la estrategia	Patrón Evaluación <i>Thinklet: StrawPoll</i>	10	Equipos de niños

		Estrategia de la siguiente ronda.	Patrón de Construcción de Consenso <i>Thinklet: MoodRing</i> <i>Thinklet: PointCounterPoint</i>		
2.4	Analizar la Estrategia	Evaluación de la estrategia. Estrategia de la siguiente ronda	Patrón evaluación <i>Thinklet: StrawPoll</i>	15	Equipos de niños

Tabla 39. Agenda detallada de actividades colaborativas

4.3.5.3 Modelo de facilitación del proceso MDF.

A continuación, se muestra el modelo de facilitación de procesos de la fase prejuego donde se representa en color gris las actividades y subactividades que requieren patrones y thinklets de colaboración.



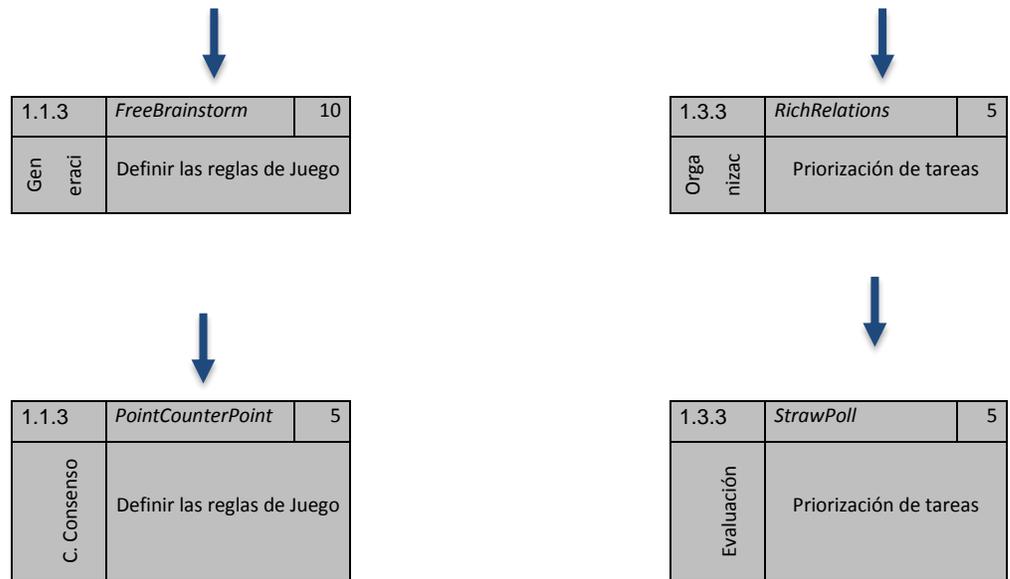
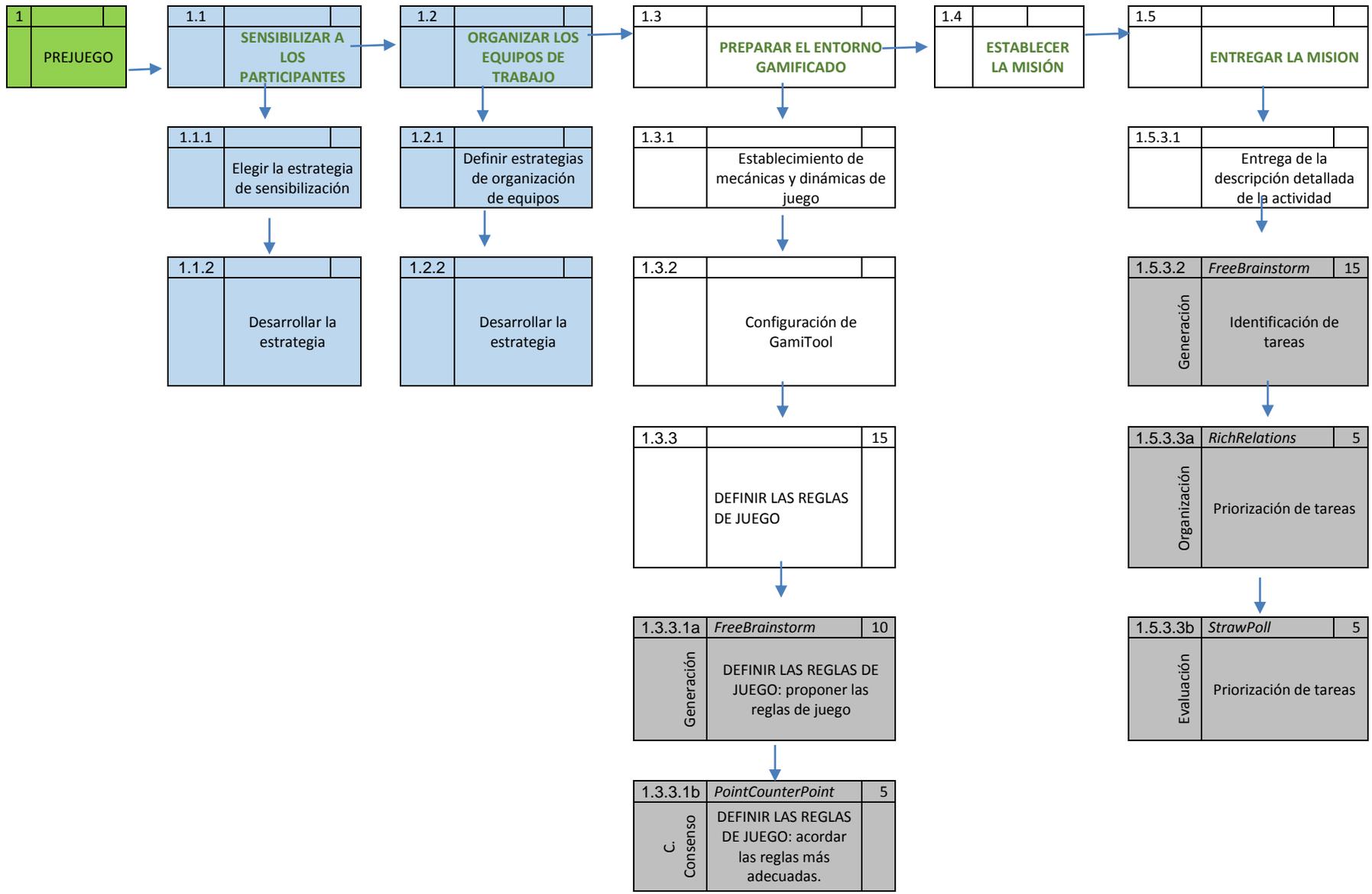


Figura 30. Modelo de facilitación de procesos – fase Prejuego.

En la siguiente figura se muestra el modelo de facilitación de procesos final de la fase prejuego donde se muestran en color gris las actividades que requieren patrones y thinklets de colaboración.



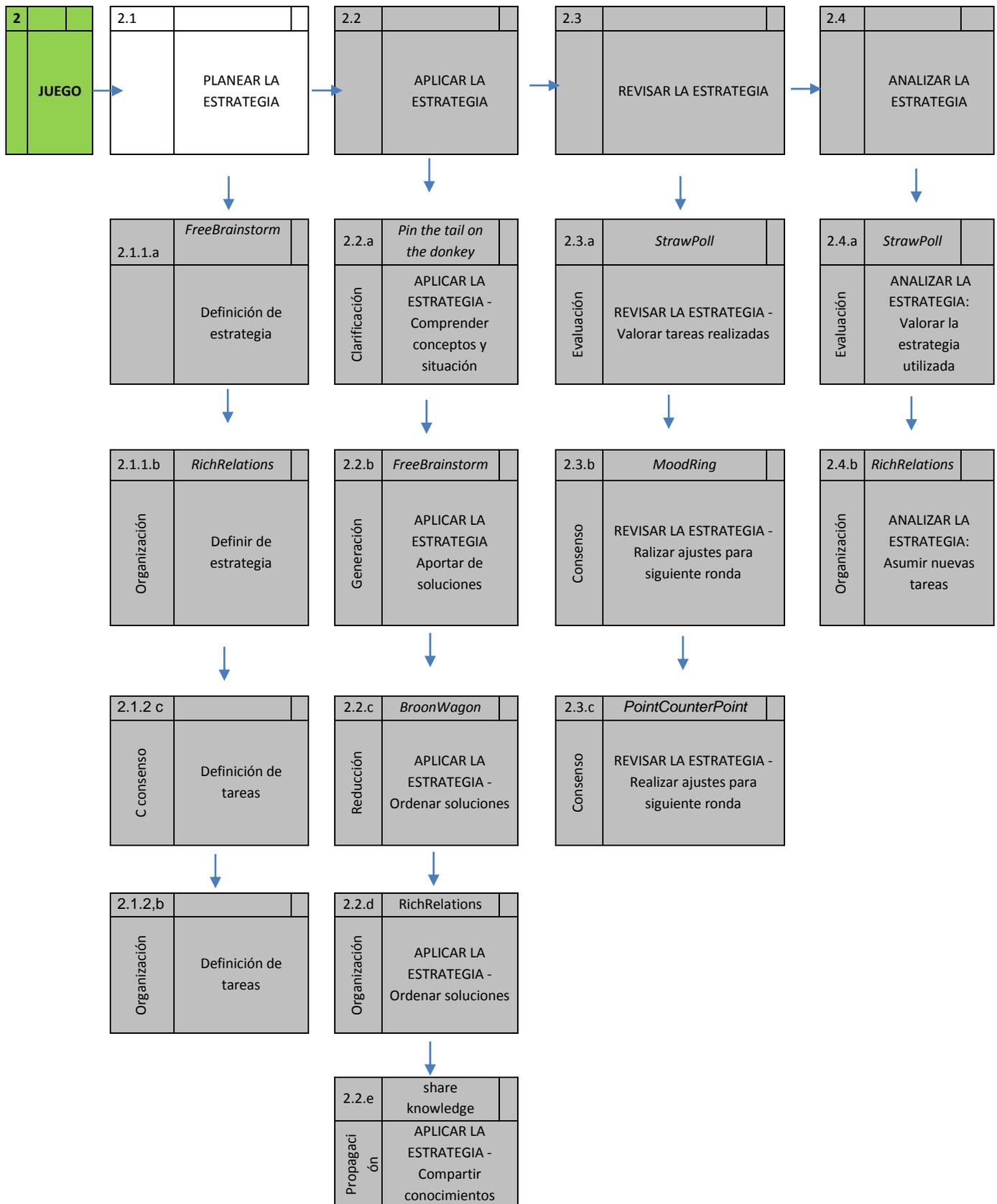


Figura 31. Modelo de facilitación de procesos Actividades colaborativas Childprogramming-C

Fase 6: Validación del diseño

Por último, se debe validar el diseño del método colaborativo objeto de estudio; para tal fin, la metodología ofrece las siguientes formas de validación: prueba piloto, recorrido, simulación, revisión [13]. Para la validación preliminar del modelo Childprogramming-C desde un enfoque colaborativo, se utilizó la prueba piloto y la revisión que se describen a continuación.

- Prueba piloto: consiste en ejecutar el método colaborativo diseñado para evaluar la efectividad del proceso [13]. Esta prueba se realizó mediante los dos estudios de caso presentados en el siguiente capítulo.

Revisión: consiste en generar discusión sobre los diseños realizados entre el equipo de trabajo y los potenciales usuarios que ejecutarán el método colaborativo, para ayudar a identificar partes del diseño ineficientes [13]. El método de revisión se llevó a cabo con el equipo de docentes que participaron a lo largo de la investigación, como uno de los grupos de usuarios potenciales, quienes en varias validaciones ayudaron a identificar mejoras en el diseño del modelo.

5. Capítulo 5. Evaluación del modelo

Comprender y analizar la colaboración de los procesos de aprendizaje requieren un análisis minucioso de la interacción grupal en el contexto de los objetivos de aprendizaje [91], para ello en esta investigación se realizó dos estudios de caso confirmatorios, como lo sugiere la última fase del método CSACE (*Case Study based Analysis in Collaboration Engineering*).

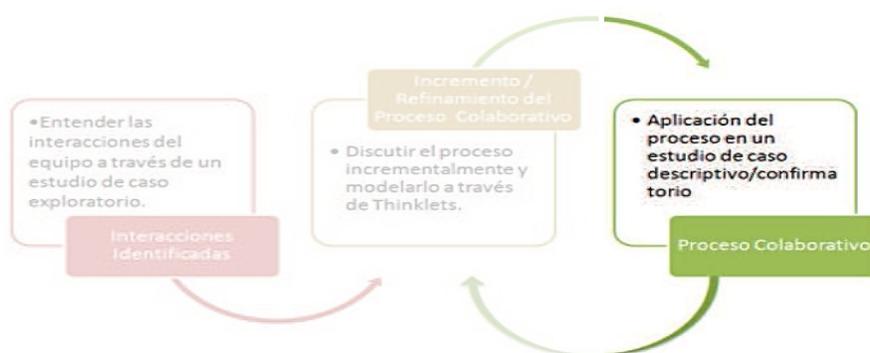


Figura 32. Fase 3 del método CSACE

5.1 Estudios de caso confirmatorios

Para la evaluación del modelo se desarrollaron dos estudios de caso confirmatorios (ver figura 34), con grupos de 12 niños cada uno. Se eligió un número de niños pequeño porque en prácticas realizadas con grupos numerosos fue más difícil el proceso de observación. Se tuvo un grupo de control donde se aplicó Childprogrammig (EC1) y un grupo de validación del modelo colaborativo (EC1') como se muestra en la siguiente figura.

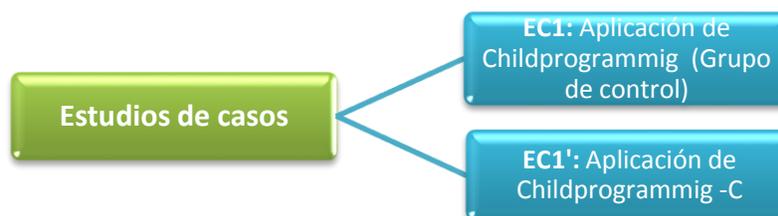


Figura 33. Estudios de caso definidos para la evaluación.

Cada estudio de caso se realizó en las siguientes sesiones:

CE	Modelo aplicado	Sesiones realizadas
	Estudio de caso 1 (EC1): Aplicación de Childprogramming	La primera parte consistió en la construcción de un poliedro en equipos de trabajo, y la caracterización de la población. Esta sesión tuvo una duración de 120 minutos (30 minutos para el proceso de caracterización).
		La segunda parte tuvo una duración de 3 horas con 45 minutos y consistió en la realización de tres retos sencillos de programación en Scratch. (Se desarrolló en dos sesiones de clase).
	Estudio de caso 1' (EC1'): Aplicación de Childprogramming.-C	La primera sesión consistió en la fase de sensibilización para que los niños aprendieran a valorar la importancia del trabajo en equipo, aquí se desarrollaron varias actividades lúdicas. Tiempo: 90 minutos.
		La segunda sesión se desarrolló en 1 hora con 10 minutos donde se reflexionó junto a los niños sobre las dinámicas realizadas y se presentaron videos de trabajo en equipo.
		La tercera sesión consistió en la construcción de un poliedro en equipos de trabajo, esta sesión tuvo una duración de 50 minutos.
		La cuarta sesión tuvo una duración de 3 horas con 10 minutos y consistió en la realización de tres retos sencillos programados en Scratch.

Tabla 40. Estudios de caso realizados

5.1.1 Contexto de la investigación

El desarrollo de los estudios de caso se realizó en las instalaciones de la Institución Educativa Agroempresarial Huasanó del municipio de Caloto - Cauca. La composición étnico-cultural de la población es homogénea, con mayoría de población mestiza e indígena. Los estudiantes son procedentes de estratos bajo, cuyos padres han superado el

nivel de básica primaria. Dentro del factor económico se puede encontrar que la mayoría de las familias tienen un estrato entre 0 y 1, con dedicación a la agricultura.

En cuanto a recursos, la institución está dotada de los elementos necesarios para el desarrollo de actividades académicas, deportivas y de recreación, dentro de lo cual se destacan los recursos tecnológicos como los habilitados en la sala de informática la cual cuenta con 20 computadores portátiles donados a la institución a través del programa Computadores para Educar, por el Ministerio de Tecnologías de la información y las comunicaciones de Colombia. Son constantes los problemas de energía y la mayoría del año no se cuenta con conexión a internet.

5.1.2 Indicadores de evaluación seleccionados

La mayoría de los estudios sobre evaluación de colaboración están enfocados a sistemas virtuales de aprendizaje [92], [15], que disponen de aplicaciones software para medir: la actividad de los usuarios, la gestión de mensajes, la gestión de tiempos y puntajes; de esta forma, muchos de los datos para calcular los indicadores propuestos en los estudios revisados son dados por el rendimiento del sistema. En la ejecución de los presente estudios de caso no se pudo contar con tantos datos cuantitativos, sino más bien con información generada a partir de la observación del comportamiento de los equipos por parte del investigador y el docente, además de las encuestas realizadas a los niños durante la aplicación del modelo.

El nivel de colaboración que puede alcanzar un grupo, está definido por un conjunto de indicadores que tienen como propósito esencial evaluar el proceso de aprendizaje colaborativo [15]. Cuatro de los indicadores están basados en las siguientes actividades propuestas por Johnson & Johnson: uso de estrategias, cooperación intra-grupo, chequeo de los criterios de éxito, y monitoreo. El quinto indicador está basado en el desempeño del grupo, [15], [92], [93]. Del mismo modo en el artículo “Indicadores de Cooperación en el Trabajo Grupal” [70] se seleccionan algunos indicadores de colaboración, que puedan ser usados para evaluar la forma en que distintos grupos de personas enfrentan una tarea en un sistema colaborativo.

A partir de los estudios presentados y utilizados en [15], [69], [70], [92] y [93], para esta investigación se adoptaran los siguientes indicadores con el objetivo de valorar el nivel de colaboración que alcanzan los equipos al utilizar el modelo Childprogramming-C.



Figura 34. Indicadores para valorar el nivel de colaboración.

5.1.2.1 **Estrategia Aplicada.**

Este indicador intenta determinar la habilidad de los miembros del grupo para generar, comunicar y aplicar de forma consistente estrategias para resolver un problema conjuntamente [88].

La discusión de la estrategia para resolver el problema, ayuda a los miembros del grupo a construir una visión compartida o un modelo mental de sus metas y tareas requeridas, este modelo mental puede mejorar la coordinación, porque cada miembro sabe cómo su tarea se ajusta a los objetivos generales del equipo [93]. Es suficiente considerar el éxito o el fracaso para resolver el problema [93].

Sin embargo, para los estudios de caso planeados, se ha considerado no solo el éxito o fracaso para resolver el problema, sino también la percepción de los integrantes del equipo, el docente y el observador respecto a la estrategia planteada y desarrollada por cada uno de los equipos. Por lo anterior se plantea el siguiente indicador:

$$\text{IEA: } V_{\text{EA}} * 30\% + V_{\text{DEA}} * 35\% + V_{\text{OEA}} * 35\%$$

Dónde:

IEA=Indicador para Estrategia Aplicada.

V_{EA} = Valoración del equipo para la estrategia aplicada.

V_{OEA} = Valoración del observador para la estrategia aplicada.

V_{DEA} =Valoración del docente para la estrategia aplicada.

5.1.2.2 Cooperación Intra-Grupo.

Esta actividad corresponde a la aplicación de estrategias de colaboración durante el proceso de trabajo en equipo. Si cada miembro del grupo es capaz de entender cómo su tarea está relacionada con la meta del equipo, entonces cada uno puede anticipar sus acciones, requiriendo el grupo menos esfuerzo de coordinación [93]. Debido a esto, el segundo indicador será CIG, y también estaría dado por la apreciación del equipo, el docente y el observador.

$$\text{ICIG: } V_{\text{CIG}} * 30\% + V_{\text{DCIG}} * 35\% + V_{\text{OCIG}} * 35\%$$

Dónde:

ICIG= Índice de cooperación intra-grupo.

V_{CIG} = Valoración dada por el equipo para la cooperación intra-grupo.

V_{OCIG} = Valoración para por el observador para la cooperación intra-grupo.

V_{DCIG} =Valoración dada por el docente para la cooperación intra-grupo.

5.1.2.3 Revisión de los criterios de éxito.

Este indicador mide el grado de compromiso de los miembros del grupo en el repaso de reglas, procesos y roles durante las actividades. Esto puede incluir revisión de los

resultados de la última tarea, asignación de actividades específicas, y definir tiempos para completar las tareas [93]. El tercer indicador IRCE refleja el interés en el desempeño individual y del equipo, y sería valorado por los integrantes del equipo, el docente y el observador. Cabe resaltar la importancia del diseño y control de tablero de tareas para valorar este criterio.

$$\text{IRCE: } VE_{\text{RCE}} * 30\% + VD_{\text{RCE}} * 35\% + VO_{\text{RCE}} * 35\%$$

Dónde:

IRCE= Índice de revisión de criterios de éxito.

VE_{RCE} = Valoración dada por el equipo para la revisión de los criterios de éxito.

VD_{RCE} =Valoración dada por el docente para la revisión de los criterios de éxito.

VO_{RCE} = Valoración dada por el observador para la revisión de los criterios de éxito.

5.1.2.4 Monitoreo.

El monitoreo se entiende como una actividad regulatoria. El objetivo de esta actividad es asegurar que el grupo mantenga las estrategias elegidas para resolver el problema, manteniéndose enfocado en los objetivos y los criterios de éxito. Si un participante no mantiene el comportamiento esperado, el grupo no alcanzará el objetivo común [93]. En este sentido, el cuarto indicador se define como:

$$\text{IM: } VE_{\text{M}} * 30\% + VD_{\text{M}} * 35\% + VO_{\text{M}} * 35\%$$

Dónde:

IM= Indicador de monitoreo

VE_{M} = Valoración dada por el equipo para el monitoreo.

VD_{M} =Valoración dada por el docente para el monitoreo.

VO_{M} = Valoración dada por el observador para el monitoreo.

5.1.2.5 *Desempeño.*

Este indicador hace referencia a la calidad de la solución propuesta a la situación. Los tres aspectos a tener en cuenta son: Calidad (que tan bueno es el resultado del trabajo colaborativo), Tiempo (total de tiempo transcurrido) y Trabajo (Cantidad total de trabajo realizado). El quinto identificador se define como:

$$ID: VE_M * 30\% + VD_M * 35\% + VO_M * 35\%$$

Dónde:

ID= Indicador de desempeño.

VE_M = Valoración dada por el equipo para el indicador del desempeño.

VD_M = Valoración dada por el docente para el indicador del desempeño.

VO_M = Valoración dada por el observador para el indicador del desempeño.

5.1.3 Instrumentos de evaluación utilizados

Dentro de esta investigación los instrumentos empleados aportan datos que posteriormente son analizados y evaluados para dar validez a la información recogida. Para ésta investigación, los principales instrumentos seleccionados son:

- *Encuesta*: establece una comunicación interpersonal entre el grupo de investigación y los sujetos de estudio con el fin de obtener datos por escrito de las respuestas de los encuestados y conocer los estados de opinión, características o hechos específicos, que estén relacionados con el tema propuesto. Este instrumento fue aplicado a los estudiantes.
- *Listas de chequeo*: Las listas de chequeo ayudan básicamente a verificar que ciertos aspectos importantes dentro del estudio de investigación se estén revisando y cumpliendo. Los instrumentos de listas de chequeo fueron diligenciados por el docente y el observador.

A continuación, se presenta el instrumento de encuesta aplicado a cada uno de los estudiantes y la lista de chequeo realizada por el docente y el observador.

ENCUESTA 1 – VALORACION DE INDICADORES PARA EVALUAR LA COLABORACION DE LOS EQUIPOS DE TRABAJO

Nombres y apellidos: _____

Numero de equipo: _____

Estimado estudiante gracias por participar de esta actividad. Después de haber participado en el desarrollo de los retos, valore de 1 a 5 los siguientes aspectos:

Estrategia Aplicada	Valoraciones	V. Final
¿El equipo identificó una estrategia para desarrollar los retos propuestos?	1____ 2____ 3____ 4____ 5____	
¿Considera que la estrategia aplicada fue la más adecuada?	1____ 2____ 3____ 4____ 5____	
¿La estrategia planteada fue plasmada en el taskboard?	1____ 2____ 3____ 4____ 5____	
Cooperación Intra-Grupo		V. Final
Califique el compromiso de los integrantes del equipo.	1____ 2____ 3____ 4____ 5____	
Califique su compromiso como integrante del equipo.	1____ 2____ 3____ 4____ 5____	
¿Cree que todos tuvieron una igual responsabilidad en las tareas asignadas?	1____ 2____ 3____ 4____ 5____	
Revisión de los criterios de éxito		V. Final
¿Se asignaron tiempos para el logro de las tareas?	1____ 2____ 3____ 4____ 5____	
¿El equipo realizó un seguimiento a las actividades asignadas en el taskboard?	1____ 2____ 3____ 4____ 5____	
Monitoreo		V. Final
En el taskboard se identifica una secuencia de pasos, ordenados y finitos para lograr el objetivo.	1____ 2____ 3____ 4____ 5____	
Desempeño.		V. Final
¿El producto terminado cumple con los requerimientos planteados por el reto?	1____ 2____ 3____ 4____ 5____	
¿El producto es entregado en el tiempo establecido?	1____ 2____ 3____ 4____ 5____	
Califique el compromiso y desempeño de su equipo en cada una de las actividades.	____ 2____ 3____ 4____ 5____	
Califique el desempeño del líder de su equipo.	1____ 2____ 3____ 4____ 5____	

¿Cuáles fueron las mayores dificultades en su equipo para el logro de los objetivos?

¿Qué aspectos cree que debe mejorar para trabajar en equipo?

INSTRUMENTO DE OBSERVACION PARA DOCENTES Y OBSERVADOR

Nombres y apellidos: _____

Área de desempeño: _____

Después de participar en el estudio de caso, por favor valore los siguientes indicadores para cada uno de los equipos. La escala de valoración es de 1 a 5, siendo uno lo más bajo y 5 la valoración más alta.

INDICADOR	Equipos			
	E 1	E 2	E 3	E 4
Estrategia Aplicada				
Cooperación Intra-Grupo				
Revisión de los criterios de éxito				
Monitoreo				
Desempeño del equipo.				
Total				

Observaciones:

Firma del Observador u docente: _____

Muchas Gracias por su colaboración



5.1.4 Caso de estudio 1

A continuación se presentan los estudios de caso realizados.

Diseño de estudio de caso. En la siguiente tabla se presenta el diseño del presente estudio de caso.

Pregunta de investigación
¿Cómo se valora la colaboración en los equipos de niños entre 10 y 12 años de edad al intentar resolver un problema utilizando el modelo ChildProgramming?
Objetivos del estudio de caso
<ul style="list-style-type: none">• Medir los indicadores de colaboración en los equipos de trabajo al utilizar el modelo ChildProgramming.
Lugar
Institución educativa de Caloto – Cauca.
Selección del estudio y unidad de análisis
La unidad de análisis para este estudio de caso son los equipos de trabajo. El estudio de caso considera 3 unidades de análisis con 12 sujetos de investigación. Las unidades de análisis seleccionadas se organizarán de acuerdo a grupos de trabajos previos de los estudiantes en sus diferentes clases.

Tabla 41. Diseño del estudio de caso 1

5.1.4.1 Ejecución del estudio de caso.

La actividad descrita en la tabla 39 se realizó durante dos sesiones, donde inicialmente se hizo una introducción de cómo se iba a trabajar y en qué consistía la actividad. Al conformar los equipos se presentaron las tareas a realizar y posteriormente los niños de forma libre construyeron una torre de forma poliédrica. A continuación, se presenta la descripción de la actividad realizada, la cual fue propuesta para la validación inicial de ChildProgramming.

Descripción de la actividad 1– sesión 1	
Nombre de la Actividad:	Construyendo modelos. La primera sesión consistió en la construcción de un poliedro en equipos de trabajo y la caracterización de la población, esta sesión tuvo una duración de 2 horas.
Descripción:	Es un juego lúdico – colaborativo que hace parte de la misión. El objetivo es construir una estructura poliédrica en condiciones normales, con características similares al ejemplo entregado y contando con un factor de tiempo. El éxito de esta misión depende del grado de colaboración entre los integrantes de cada equipo por ello es necesario que coordinen muy bien el trabajo y manejen un único canal de comunicación de tal forma que se entiendan, comprendan y compartan sus ideas.
Participantes :	<ul style="list-style-type: none"> ✓ Niños de séptimo grado de la institución educativa. ✓ Docente del área de informática de la institución.
Conformación de los Equipos:	<p>Para este estudio de caso se contó con un curso donde se organizaron grupos teniendo en cuenta el número de niños que lo conformaban así:</p> <p>Grado 7°: 3 equipos.</p> <p>Se permitió que los niños organizarán sus equipos, pues ellos han trabajado desde años anteriores en actividades grupales, esto se considera un factor importante para identificar su comportamiento colaborativo e innato.</p>
Ambiente de trabajo:	Se ejecutó la actividad en un ambiente amplio donde se pudieran dispersar los niños y no generar desorden, por ello el espacio adaptado fue el patio principal de la institución, el cual estaba dotado de varias mesas de trabajo, pues la actividad estaba pensada para conformar equipos y ubicarlos en cada mesa. La distribución de las mesas estaba organizada en orden de equipos, lo que garantizaría una mejor observación.
Materiales:	<p>Para el desarrollo de la actividad cada equipo contó con los siguientes materiales para realizar las construcciones.</p> <ul style="list-style-type: none"> - 24 espaguetis - 25 Masmelos por equipo - Una hoja de papel de color - Un lápiz - Un borrador - Un plegable con información de la actividad

<p>Metodología:</p>	
<p>Entregables de la Actividad:</p>	<p>Torre con forma poliédrica construida a base de espaguetis y masmelos.</p>
<p>Unidades Generadas</p>	<p>La unidad generada para esta actividad fue una la torre poliédrica de palos y masmelos construida y entregada de forma tangible. (Una sola unidad)</p>
<p>Defecto</p>	<p>Estructuras o torres con imperfecciones de forma, apariencia y similitud con respecto al modelo presentado.</p>

Tabla 42. Descripción Actividad No. 1 tomada y adaptada de [14]

A continuación, se presentan el plegable utilizado y algunas evidencias fotográficas de la actividad realizada.



Figura 36. Evidencia fotográfica - Actividad 1

Actividad 2: a continuación, se presenta la actividad número dos del estudio de caso.

Descripción de la actividad 2	
Nombre de la Actividad:	Construyendo tres retos en Scratch.
Descripción:	Esta actividad consiste en que los estudiantes desarrollen tres retos en Scratch.
Participantes:	<ul style="list-style-type: none"> ✓ Niños de grado séptimo de la Institución Educativa AH. ✓ Docente del área de informática de la institución.
Conformación de los Equipos:	Para este estudio de caso se contó con un curso donde se organizaron grupos teniendo en cuenta el número de niños que lo conformaban así: Grado 7°: 3 equipos de 4 integrantes.
Ambiente de Trabajo:	Sala de sistemas
Materiales:	<ul style="list-style-type: none"> - Equipos de computo - Cartulina - Marcadores - Mesas de trabajo.

Una vez conformados los equipos se ubicaron en la mesa asignada la cual tenía sobre ella el material suficiente y necesario para desarrollar la actividad. Para el desarrollo de la aplicación se utilizó el modelo Childprogramming inicial. La siguiente figura presenta la guía de trabajo.

Metodología:

*DESARROLLANDO EL PENSAMIENTO COMPUTACIONAL CON
CHILDPROGRAMMING
Universidad del Cauca
J.A. Hurtado- C. Collazos*

Misión 1 (Vale por 10 Puntos)

Objetivo: Construir en Scratch el juego de la explosión de bombas con un disparador por medio del ratón de acuerdo al centro de una mira (Manejarlo el Ratón)

Consideraciones de la misión:

1. Hacer un par de bombas que se muevan en la pantalla girando y rebotando en los bordes.
2. Hacer una mira que se mueva con el ratón.
3. Al presionar el botón derecho del ratón, si una bomba está dentro de la mira, entonces será reventada, y el puntaje del juego incrementado en el valor de acuerdo al valor de la bomba (3, 5, 20 Puntos).
4. Cada bomba reventada renace a los 0.3 Segundos.
5. El juego tiene una duración de 60 segundos.
6. Se debe informar permanentemente sobre el puntaje y el tiempo.

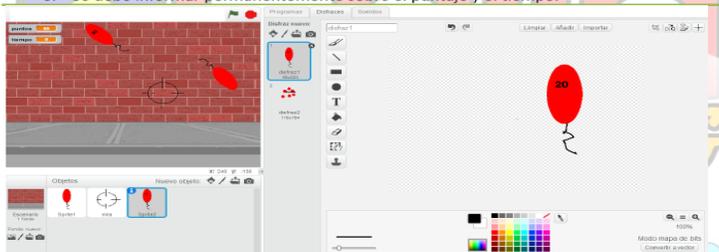


Figura 1. Juego de la Explosión de Globos

Misión 2 (Vale por 10 Puntos)

Objetivo: Construir un programa que permita conducir un auto a través de un camino.

Consideraciones de la misión:

1. Hacer un escenario que brinde un camino rectangular por donde deberá transitar un auto.
2. Al presionar la barra espaciadora el auto desde un punto de partida delimitado arrancará sin control de velocidad en línea recta hacia la dirección que esté apuntando, con una velocidad inicial determinada por el jugador.
3. El conductor del auto sólo podrá hacer los giros a través de las flechas sin control de la velocidad.

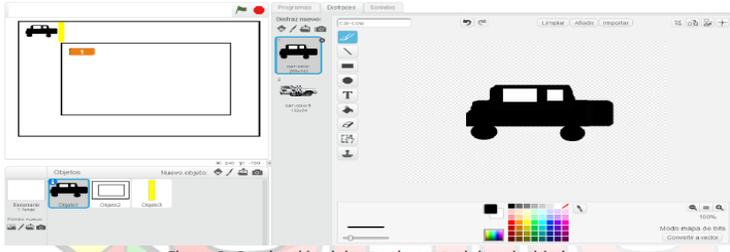
	<p>4. Si el auto toca el borde del camino, entonces el juego se terminará indicando las vueltas que se lograron.</p> <p>5. Una vuelta se cuenta cuando vuelve a pasar por el punto de partida.</p> <p>6. Se debe informar al conductor cuántas vueltas lleva en la medida que avanza.</p> <p>7. El auto automáticamente incrementa en un valor constante equivalente al 35% de su velocidad inicial a cada vuelta.</p>  <p>Figura 2. Conducción del auto sin control de velocidad</p> <p>Misión 3 (20 Puntos) Objetivo: Construir una historieta de 5 escenas. Descripción General:</p> <ol style="list-style-type: none"> 1. Desarrollo una historieta que involucre a menos 3 personajes 2. Debe contener mínimo 3 escenarios. 3. La historieta debe mostrar 5 escenas como mínimo.
<p>Entregables de la Actividad:</p>	<ul style="list-style-type: none"> - Aplicaciones software realizada. - Taskboard.
<p>Unidades Generadas</p>	<p>Aplicaciones de: misión 1-3</p>

Tabla 43. Descripción Actividad No. 2

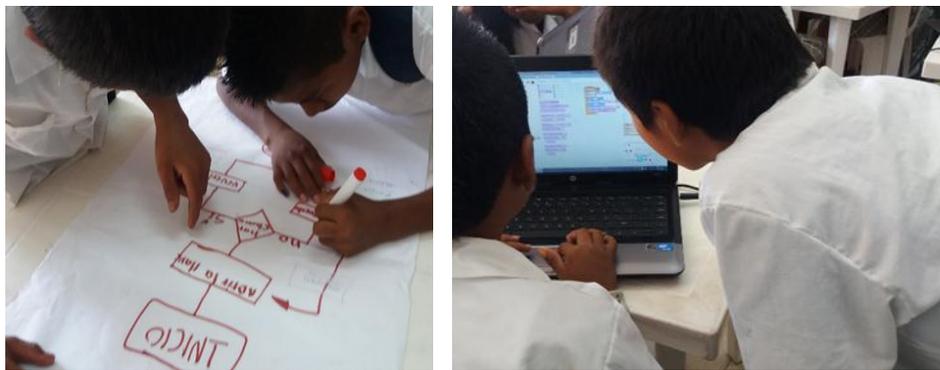


Figura 38. Evidencia fotográfica - Actividad 2

5.1.4.2 Resultados de estudio de caso 1.

La siguiente tabla muestra los resultados de cada indicador valorado por cada uno de los integrantes del equipo y el promedio del equipo.

Equipo	Estudiante	Estrategia Aplicada	Cooperación Intra-Grupo	Revisión de los Criterios de éxito	Monitoreo	Desempeño del equipo
EQ1	Estudiante 1	3	3	2	2	2
	Estudiante 2	2	3	2	2	3
	Estudiante 3	3	2	2	2	3
	Estudiante 4	3	1	1	2	3
Promedio		VE _{EA} =2,75	VE _{CIG} =2,25	VE _{RC} =1,75	VE _M =2	VE _D =2,75
EQ2	Estudiante 5	2	2	2	2	2
	Estudiante 6	2	3	3	3	3
	Estudiante 7	2	3	3	3	3
	Estudiante 8	2	3	3	3	4
Promedio		VE _{EA} =2	VE _{CIG} =2,75	VE _{RC} =2,75	VE _M 2,75	VE _D =3
EQ3	Estudiante 9	4	4	4	5	5
	Estudiante 10	3	4	4	4	4
	Estudiante 11	4	3	4	4	4
	Estudiante 12	4	3	4	4	4
Promedio		VE _{EA} =3,75	VE _{CIG} =3,5	VE _{RC} =4	VE _M 4,25	VE _D =4,25

Tabla 44. Resultados indicadores del estudio de caso 1

Resultados del observador y el docente (promediados):

Equipos	Estrategia Aplicada	Cooperación Intra-Grupo	Revisión de los Criterios de éxito	Monitoreo	Desempeño del equipo	Desempeño del líder
EQ1	3	3	2	2	3	3
EQ2	2	2	2	2	2	2
EQ3	3	3	4	4	4	4

Tabla 45. Resultados de indicadores del estudio de caso 1 – Docente e investigador

Los valores de los indicadores de colaboración en cada uno de los equipos sería:

Equipos	Estrategia Aplicada IEA	Cooperación Intra-Grupo ICIG	Revisión de los Criterios de éxito IRCE	Monitoreo IM	Desempeño del equipo ID
Equipo 1	2,925	2,775	1,925	2	2,925
Equipo 2	2	2,525	2,525	2,525	2,7
Equipo 3	3,525	3,35	4	4,175	4,175
promedio	2,8167	2,8833	2,8167	2,9000	3,2667

Tabla 46. Promedio de valoración de indicadores del estudio de caso 1

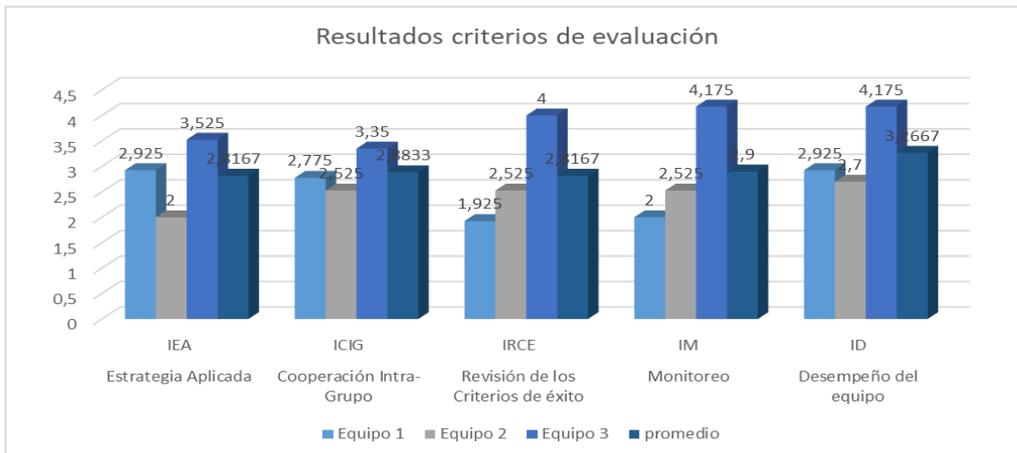


Figura 39. Gráfica Indicadores de colaboración - estudio de caso 1.

Como se puede apreciar los valores de los indicadores Cooperación intra-grupo y la estrategia aplicada son las más bajas. En promedio los valores de los indicadores están entre 2.8 y 3.2, lo cual se podría decir que están en un rango aceptable.

5.1.5 Estudio de caso 2

5.1.5.1 *Diseño de estudio de caso.*

En la siguiente tabla se presenta el diseño del presente estudio de caso.

Pregunta de investigación
¿Cómo se valora la colaboración en los equipos de niños entre 10 y 12 años de edad al intentar resolver un problema utilizando el modelo ChildProgramming-C?
Objetivo del estudio de caso
<ul style="list-style-type: none">• Medir los indicadores de colaboración en los equipos de trabajo al utilizar el modelo ChildProgramming-C.
Lugar
Institución educativa de Caloto – Cauca.
Selección del estudio y unidad de análisis
La unidad de análisis para este estudio de caso son los equipos de trabajo. El estudio de caso considera 3 unidades de análisis con 12 sujetos de investigación. Las unidades de análisis seleccionadas se organizarán de acuerdo a grupos de trabajos previos de los estudiantes en sus diferentes clases. Este estudio permitirá analizar las características y comportamientos de los equipos de trabajo. La técnica de recolección de información seleccionada es la observación y se aplicará como instrumento una lista de chequeo, de igual manera se aplicará una encuesta a los estudiantes.
Criterios para identificar colaboración en equipos de trabajo.
- Se utilizaron los mismos criterios definidos para el estudio de caso anterior.

Tabla 47. Diseño del estudio de caso confirmatorio

5.1.5.2 *Ejecución del estudio de caso.*

El estudio de caso se realizó en cuatro sesiones, las dos primeras se orientaron a la actividad de sensibilizar a los niños sobre la importancia del trabajo colaborativo, la tercera en analizar la colaboración y la última sesión se aplicaron las demás fases del modelo Childprogramming -C para el desarrollo de los tres retos de programación propuestos. A continuación, se describe cada una de las sesiones.

Sesión 1: La primera sesión consistió en la fase de sensibilización para que los niños aprendan a valorar la importancia del trabajo en equipo, aquí se desarrollaron varias actividades lúdicas. Tiempo 90 minutos donde se utilizó varias de las estrategias de sensibilización propuestas.

Actividades de sensibilización al trabajo en equipo



Figura 40: Actividad de sensibilización colaborativa 1 – Escribir juntos



Figura 41. Actividad de sensibilización colaborativa 2 – Competencia tarros



Figura 42. Actividad de sensibilización colaborativa 3 – Competencia con pitillos



Figura 43. Actividad de sensibilización colaborativa 4- competencia con bombas



Figura 44. Actividad de sensibilización colaborativa 5 organización de números



Figura 45. Actividad de sensibilización colaborativa 6 – mi guía

Tabla 48. Diseño del estudio de caso confirmatorio

Sesión 2. La segunda sesión se desarrolló en 1 hora con 10 minutos donde se reflexionó junto a los niños sobre las dinámicas realizadas y se presentaron videos de trabajo en equipo.

Los videos presentados se consultaron en:

- ✓ <https://www.youtube.com/watch?v=e2xNXr4oox8>
- ✓ <https://www.youtube.com/watch?v=LAOICltn3MM>
- ✓ <https://www.youtube.com/watch?v=226iRoOSIWM>
- ✓ <https://www.youtube.com/watch?v=Q0rRUAMevUI>
- ✓ <https://www.youtube.com/watch?v=HilITPKoOsw>
- ✓ <https://www.youtube.com/watch?v=TiNUaHXsjBA>

Sesión 3: para esta actividad se realizó la construcción del poliedro descrita en la figura 36, obteniendo muy buenos resultados en menor tiempo de la actividad realizada en el caso 1, además se observó una mejor estrategia y asignación de roles.

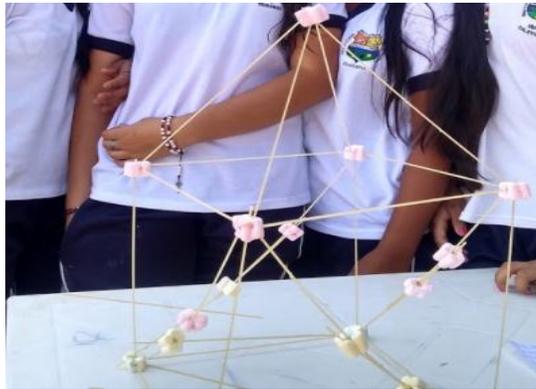


Figura 46. Actividad de sensibilización 1 - Poliedro

Sesión 4: La cuarta sesión tuvo una duración de 3 horas con 10 minutos y consistió en la realización de tres retos sencillos programado en Scratch siguiendo el modelo Childprogramming- C. (Ver figura 48). En esta sesión fue muy importante observar como la programación por parejas fue más productivo que en caso de estudio 1, de igual manera se llegó más rápido al consenso y a la toma de decisiones compartidas.

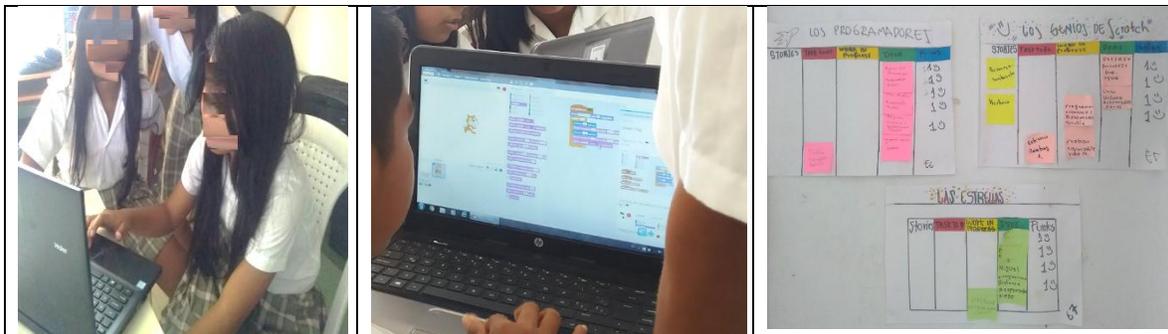


Figura 47. Desarrollo de aplicaciones software

Resultados estudio de caso 2.

En la tabla 48 se presenta los resultados de las encuestas realizadas a los niños en el estudio de caso 2:

Tabla 49. Resultados del estudio de caso 2

Equipo	Estudiante	Estrategia Aplicada	Cooperación Intra-Grupo	Revisión de los Criterios de éxito	Monitoreo	Desempeño del equipo
Las estrellas	Estudiante 1	3	5	5	5	4
	Estudiante 2	4	5	4	4	4
	Estudiante 3	3	3	4	4	4
	Estudiante 4	3	4	4	4	4
Promedio		$VE_{EA} = 3,25$	$VE_{CIG} = 4,25$	$VE_{RC} = 4,25$	$VE_M = 4,25$	$VE_D = 4,25$
Los programadores	Estudiante 5	3	4	4	4	4
	Estudiante 6	4	5	3	4	5
	Estudiante 7	5	5	5	5	5
	Estudiante 8	5	3	4	4	4
Promedio		$VE_{EA} = 4,25$	$VE_{CIG} = 4,25$	$VE_{RC} = 4$	$VE_M = 4,25$	$VE_D = 4,25$
Los genios de Scratch	Estudiante 9	4	4	4	5	5
	Estudiante 10	3	3	1	3	2
	Estudiante 11	4	4	4	4	4
	Estudiante 12	4	3	5	5	5
Promedio		$VE_{EA} = 3,75$	$VE_{CIG} = 3,5$	$VE_{RC} = 3,5$	$VE_M = 4,24$	$VE_D = 4$

5.1.5.3 Resultados del estudio de caso

En la tabla 49 se presentan los resultados de las valoraciones dadas a cada uno de los grupos por parte del observador y el docente (valores promediados):

Equipos	Estrategia Aplicada	Cooperación Intra-Grupo	Revisión de los Criterios de éxito	Monitoreo	Desempeño del equipo
Las estrellas	3	3	5	5	4
Los programadores	4	4	5	5	4
Los genios de Scratch	3	3	5	5	4

Tabla 50. Resultados del estudio de caso 2 – docente e investigador

Los valores de los indicadores de colaboración en cada uno de los equipos sería:

Equipos	Estrategia Aplicada IEA	Cooperación Intra-Grupo ICIG	Revisión de los Criterios de éxito IRCE	Monitoreo IM	Desempeño del equipo ID
Las estrellas	3,075	3,375	4,775	4,775	4
Los programadores	4,075	4,075	4,7	4,775	4,15
Los genios de Scratch	3,225	3,15	4,55	4,775	4
Promedio	3,45	3,53	4,67	4,775	4,05

Tabla 51. Promedio de valoración de indicadores del estudio de caso 2

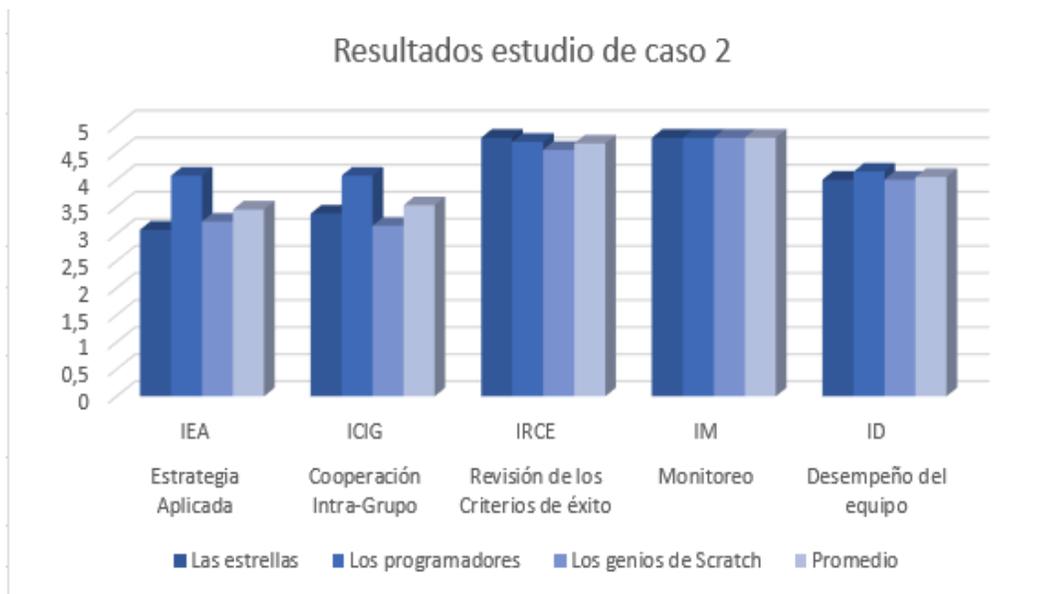


Figura 48. Gráfica Indicadores de colaboración - estudio de caso 1.

Comparación de indicadores en los dos	Estrategia Aplicada IEA		Cooperación Intra-Grupo ICIG		Revisión de los Criterios de éxito IRCE		Monitoreo IM		Desempeño del equipo ID	
	EC1	EC2	EC1	EC2	EC1	EC2	EC1	EC2	EC1	EC2

casos de estudio.	2,925	3,075	2,775	3,375	1,925	4,775	2	4,775	2,925	4
	2	4,075	2,525	4,075	2,525	4,7	2,525	4,775	2,7	4,15
	3,525	3,225	3,35	3,15	4	4,55	4,175	4,775	4,175	4
PROMEDIO	2,8	3,5	2,9	3,5	2,8	4,7	2,9	4,8	3,3	4,1

Tabla 52. Resultados finales de criterios evaluados.

De los resultados de la tabla 51 se puede decir que:

- Los indicadores Aplicación de estrategia y Cooperación intragrupo varían muy poco (0,7 y 0,6 puntos).
- Los indicadores Revisión de los criterios de éxito y monitoreo, tienen una mayor variación (1,9 puntos).
- Los Taskboard fue una importante herramienta para el desempeño y evaluación de los indicadores aplicación de estrategia y Cooperación intragrupo.



Figura 49. Variación de indicadores evaluados en los estudios de caso

- Al comparar CE1 y CE2 se puede apreciar que los indicadores evaluados mejoran significativamente en el estudio de caso dos, principalmente el monitoreo, la revisión de los criterios de éxitos, y el desempeño del equipo.

- Aunque en el estudio CE2 se tome un gran lapso para sensibilizar a los estudiantes sobre la importancia de la colaboración y el desempeño de cada uno en un equipo de desarrollo de software, cabe destacar la importancia del tablero taskboard para el modelo Childprogramming – C, este elemento se convirtió en el mejor aliado para la asignación de las tareas, el seguimiento y el control de las actividades.
- El desempeño de los líderes juega un papel fundamental en el desarrollo del modelo Childprogramming-C, si se cuenta con niños líderes comprometidos el desarrollo de las aplicaciones será más amena.
- Aunque muchos aspectos mejoraron se continúa encontrando estudiantes que no están interesados en el desarrollo de las actividades porque siempre buscan trabajar solos.

6. Capítulo 6. Conclusiones y trabajos futuros

A continuación, se presentan cada una de las conclusiones resultantes de este trabajo de investigación, donde se planteó un modelo para el desarrollo de software desde la perspectiva de la ingeniería de la colaboración a partir del modelo original de Childprogramming:

- La propuesta de un modelo desde la perspectiva de la Ingeniería de la Colaboración basado en ChildProgramming, permitió diseñar y estructurar actividades desde el enfoque colaborativo, con el objetivo de guiar la ejecución de las tareas en un entorno donde los profesionales que coordinan un proceso no necesariamente sean ingenieros y/o expertos en colaboración, además, el modelo propuesto promueve el desarrollo del pensamiento computacional de los niños, el mejoramiento de la comunicación, el autoaprendizaje, la participación, y la responsabilidad de un niño como integrante de un equipo.
- La generación de los patrones de Colaboración “liderar” y “propagación de conocimiento”, es muy pertinente para la Ingeniería de la Colaboración, porque son nuevas guías para la ejecución de un proceso. Estos thinklets fueron validados por expertos en colaboración.
- Esta investigación propone dos Childthinklets para procesos colaborativos desarrollados por niños, como un aporte a la Ingeniería de la Colaboración los cuales son: “Analysis in Pairs” y “Divide et impera”, estos thinklets integran aspectos de las técnicas colaborativas de discusión y técnicas colaborativas de resolución de problemas y se convierten en una opción que incrementa el número de thinklets que hasta el momento es limitado.
- En esta investigación el método CSACE (Case Study based Analysis in Collaboration Engineering) y la Metodología para el Desarrollo de Procesos Colaborativos, fueron una valiosa combinación para establecer las necesidades de procesos colaborativos

desde las interacciones de los niños, así como para la evaluación empírica de la efectividad del proceso colaborativo final del modelo.

- La teoría de aprendizaje plantea una variedad de técnicas colaborativas de aprendizaje que en este trabajo fueron analizadas con el apoyo de un grupo de docentes. Como resultado de este análisis se propone integrar a la Metodología para el Desarrollo de Procesos Colaborativos dichas técnicas como herramientas didácticas para la ejecución de los procesos colaborativos, que no solo puede ser aplicables a procesos de enseñanza de desarrollo de software, sino también a entornos empresariales y sistemas colaborativo virtuales.
- Las actividades propuestas para sensibilizar a los niños hacia la importancia del trabajo colaborativo, permitieron mejorar las características de la colaboración tales como: la interdependencia positiva, la igual participación, la responsabilidad individual y la definición de estrategias durante el desarrollo de software. Este tipo de habilidades podría permitirles a los futuros profesionales, insertarse de forma más adecuada a los entornos laborales, en los cuales el trabajo se desarrolla habitualmente en equipos de trabajo.
- Uno de los elementos claves para poder mejorar las características del trabajo colaborativo durante el proceso de desarrollo de software siguiendo el modelo ChildProgramming, fue el tablero taskboard que permitió hacer el seguimiento y la evaluación de las tareas por realizar, las tareas en progreso y las tareas hechas. De igual manera, la gamificación por medio de la asignación de puntos, motivó a los estudiantes durante todo el desarrollo de sus retos de programación.
- La participación y aporte de un grupo de docentes durante el desarrollo de Childprogramming-C, permitió capturar requerimientos y soluciones para consolidar un modelo de desarrollo de software más estable desde diferentes aspectos de la Ingeniería del Software, la Ingeniería de la Colaboración, la gamificación y la pedagogía. El método de análisis y evaluación con expertos (Docentes), se debe seguir incorporando en la evolución de Childprogramming, con el objetivo de lograr un modelo

propuesto a partir de las necesidades reales de los usuarios, de una manera dinámica y metódica.

- Las dinámicas y mecánicas de juego seleccionadas en los estudios de caso fueron fundamentales, para que los equipos conformados por niños vivieran el proceso de desarrollo de software como una actividad lúdica. La gamificación mejoró el rendimiento de los equipos y la colaboración entre sus integrantes. Para este tipo de actividades es fundamental que los docentes e investigadores realicen una correcta definición y planeación de las dinámicas y mecánicas de juego, porque el proceso de aprendizaje se acelera en el momento que un equipo inicia a ganar puntos y los demás se motivan a no quedarse atrás.
- En los diferentes estudios de caso se pudo observar que los niños líderes juegan un rol fundamental dentro de los equipos. Por esta razón, es necesario descubrir líderes potenciales antes de iniciar los retos de programación.
- El incremento de la colaboración en los equipos de trabajo no solo está definido por la estructuración de las actividades del modelo ChildProgramming desde el enfoque la Ingeniería de la Colaboración, es necesario definir un proceso más completo que involucre a docentes, técnicas pedagógicas y de aprendizaje colaborativo.
- Aunque se planteen actividades que fomenten el trabajo colaborativo dentro de los equipos de desarrollo de software, se encontrarán integrantes que no desean trabajar colaborativamente. Por lo tanto, se deben continuar planteando estrategias que permiten su integración, participación y motivación, en este tipo de situaciones el docente es el principal interventor para buscar una solución.

6.1 Trabajos futuros

Como trabajos futuros se tiene:

- Aunque la ISO/IEC 25010 ofrece normas y estándares directos para la evaluación de usabilidad de productos software, este trabajo también realizó un acercamiento a la evaluación de la capacidad de aprendizaje como subcaracterística de la usabilidad en un proceso de desarrollo de software, lo cual es el inicio para continuar investigando sobre cómo mejorar la calidad no solo de los productos software, sino también de los procesos de enseñanza de programación de software.
- Es necesario continuar realizando más estudios de caso en otros contextos escolares con el fin de refinar el modelo propuesto, teniendo en cuenta los roles de docente y estudiante en las fases del proceso.
- Aunque en este trabajo se proponen dos patrones de colaboración y dos thinklets de colaboración es importante ofrecer más documentación de los mismos, así como realizar las validaciones pertinentes.
- Aunque Childprogramming se continuará formalizando y documentando como un modelo de procesos desde la Ingeniería del Software, la Ingeniería de la Colaboración y la mejora de procesos, este trabajo permitió identificar la necesidad de estructurar el modelo pensando también en el usuario final, por tanto, es importante continuar en el diseño de guías prácticas y más didácticas que orienten a los docentes u otras personas que deseen aplicar el modelo a un grupo de niños para enseñar a programar.
- Este trabajo propone nuevas prácticas colaborativas para mejorar el proceso de Childprogrammig, aunque es un valioso aporte, se abre la iniciativa para continuar en la mejora y propuesta de más prácticas.

- Es importante trabajar en la construcción de un lenguaje de modelamiento que permita realizar los modelos de facilitación del proceso MDF de las actividades colaborativas.
- Mediante el desarrollo de este trabajo se abre una puerta para evaluar las estrategias de sensibilización planteadas, en otros entornos de desarrollo de software, como por ejemplo en los equipos virtuales de desarrollo de software.

6.2 Publicaciones y reconocimientos

Como resultado del trabajo de investigación fueron realizadas algunas publicaciones, no obstante, queda pendiente generar más publicaciones donde sean presentados los resultados finales logrados. A continuación, se presentan las publicaciones realizadas hasta el momento:

- “ChildProgramming-C: as an improvement of the collaborative dimension of the ChildProgramming model”. Revista Categoría C de Colciencias: Entre Ciencia e Ingeniería. ISSN 1909-8367. Colombia. Disponible en: <http://biblioteca.ucp.edu.co/OJS/index.php/entrecei/article/view/3551/3864>
- Analizando la Capacidad de Aprendizaje como característica de Usabilidad, en el Modelo ChildProgramming. 18° Conferencia Internacional de Interacción Humano computador 2017. Septiembre, 2017 - Cancún (Mexico). Memorias del evento Scientia et cognito. V1. Motiel y Soriano Editorial 2017. Página 251.
- ChildProgramming – C, extendiendo ChildProgramming desde la Ingeniería de la Colaboración. XI Congreso Colombiano de Computación 2016 - Popayán – Colombia. Memorias del evento.
- “Ganadora de V encuentro Nacional de Experiencias Significativas de uso de TIC en el aula”, Ministerio de Educación de Colombia. Junio 2017. Propuesta: “Programando retos”. Experiencia significativa de programación de software con Childprogramming y colaboración en la Institución Educativa agroempresarial Huasanó.
- Ganadora de Simposio de Maestría y doctorado. Congreso Colombiano de Computación 2017. Universidad Autónoma de Occidente. Cali (Colombia).

Artículo en revisión:

- “Analyzing collaboration in the gamification process of Childprogramming”. Artículo enviado a: Revista Comunicar de España. En Journal Citation Reports 2017 (JCR) Q1. E-ISSN: 1988 – 3293. <https://www.revistacomunicar.com/>.

Referencias

- [1] G.-J. De Vreede and R. O. Briggs, "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks," *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 0, no. C, p. 17, 2005.
- [2] J. Archer-Kath, D. W. Johnson, and R. T. Johnson, "Individual versus Group Feedback in Cooperative Groups," *The Journal of Social Psychology*, vol. 134. pp. 681–694, 1994.
- [3] "E. Turban, «Management Support System, Decision Support and Expert System,» New Jersey, Prentice Hall, 1995, pp. 10-127.," p. 1995, 1995.
- [4] P. S. Taylor *et al.*, "Agile Software Development," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 1, no. 6, pp. 679–680, 2009.
- [5] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: some issues and experiences," *Commun. ACM*, vol. 34, no. 1, pp. 39–58, 1991.
- [6] C. Collazos, "How to take advantage of ' cooperative learning ' in the classroom," *Educ. y Educ.*, vol. Volumen 9, no. 4128, pp. 61–76, 2006.
- [7] J. A. H. Alegría and C. A. Collazos, "Analyzing and Evaluating Collaborative Processes using Case Studies in the Software Development Context," *Proc. XV Int. Conf. Hum. Comput. Interact.*, pp. 1–2, 2014.
- [8] G. Kolfshoten and G.-J. Vreede, "The Collaboration Engineering Approach for Designing," in *Groupware: Design, Implementation, and Use*, vol. 4715, 2007, pp. 95–110.
- [9] J. M. Wing, "Computational thinking and thinking about computing," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 366, no. 1881, pp. 3717–3725, Oct. 2008.
- [10] R. F. Zúñiga Muñoz and J. A. Hurtado Alegría, "Explorando los procesos de abstracción computacional en niños: un estudio de caso," *ResearchGate*, no. September 2014, pp. 1–6, 2015.
- [11] J. Wing, "Computational Thinking Benefits Society," *Soc. Issues Comput.*, 2014.
- [12] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav, "Computational thinking in compulsory education: Towards an agenda for research and practice," *Educ. Inf. Technol.*, vol. 20, no. 4, pp. 715–728, 2015.
- [13] J. Hurtado, C. Collazos, S. Cruz, and O. Rojas, "Child Programming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Lúdica, la Colaboración y la Agilidad," *Rev. Univ. RUTIC*, vol. 1, pp. 9–14, 2012.
- [14] S. T. Cruz and O. E. Rojas, "Un Modelo Para la Enseñanza de la Programación de Software en niños a través de Estrategias Colaborativas," Universidad del Cauca, 2013.
- [15] C. A. Collazos *et al.*, "Evaluating collaborative learning processes using system-based measurement," *Educational Technology and Society*, vol. 10, no. 3. pp. 257–274, 2007.

- [16] J. George, L. Gan, and B. Jessica, "Learning cooperative learning via cooperative learning," *Hawker Brownlow Education Learning*, 2007.
- [17] "Sitio web del grupo de investigación IDIS (Investigación y Desarrollo en Ingeniería de Software) . Available: <http://www.unicauca.edu.co/idis/>."
- [18] J. L. Jurado and C. A. Collazos, "La mejora de procesos en la gestión de proyectos , una perspectiva desde la Ingeniería de la Colaboración," *Ingenium*, vol. 7, no. 15, pp. 51–59, 2013.
- [19] Granollers, and M. G. Mata, "Usabilidad Y La Accesibilidad," *Agenda*. 2010.
- [20] G. L. Kolschoten, R. O. Briggs, G. J. de Vreede, P. H. M. Jacobs, and J. H. Appelman, "A conceptual foundation of the thinkLet concept for Collaboration Engineering," *Int. J. Hum. Comput. Stud.*, vol. 64, no. 7, pp. 611–621, 2006.
- [21] G.-J. de Vreede, R. O. Briggs, and A. P. Massey, "Collaboration Engineering: Foundations and Opportunities," *J. Assoc.*, vol. 10, pp. 121–137, 2009.
- [22] P. Mendoza and P. Galvis, "Juegos Multiplayer: Juegos Colaborativos Para La Educación1," *Informática Educ.*, vol. 11, no. 2, pp. 223–239, 1998.
- [23] "How to take advantage of 'cooperative learning' in the classroom."
- [24] D. W. Johnson and R. T. Johnson, *Learning together and alone*, vol. Prentice H. 1994.
- [25] M. Sapon-Shevin, B. J. Ayres, and J. Duncan, "Cooperative learning and inclusion.," *Creativity and collaborative learning: The practical guide to empowering students, teachers, and families (2nd ed.)*. pp. 209–221, 2002.
- [26] G. Stahl, "Global Introduction to CSCL Gerry Stahl." 2010.
- [27] G. Stahl, "Group Cognition," *Technology*, p. 359, 2006.
- [28] G. Beatriz, "Catálogo de prácticas colaborativas para apoyar la enseñanza de la programación a estudiantes de género femenino en programas de Ingeniería de Sistemas.," Universidad del Cauca, 2017.
- [29] C. S. González, C. A. Collazos, and R. García, "Desafío en el diseño de MOOCs: incorporación de aspectos para la colaboración y la gamificación," 2016.
- [30] R. O. Briggs, A. P. Massey, and G.-J. de Vreede, "Collaboration Engineering : Foundations and Opportunities : Editorial to the Special Issue on the Journal of the Association of Information Systems Journal of the Association for Information Systems Collaboration Engineering : Foundations and Opportunitie," no. March 2016, 2009.
- [31] N. C. R. R. J. Jay F Nunamaker Jr, Robert O Briggs, *Collaboration Systems: Concept, Value, and Use*, vol. 5, no. 3. London, 2015.
- [32] R. O. Briggs, R. O. Briggs, G. De Vreede, G. De Vreede, J. F. Nunamaker Jr., and J. F. Nunamaker Jr., "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems.," *J. Manag. Inf. Syst.*, vol. 19, no. 4, pp. 31–64, 2003.
- [33] A. F. Solano, "Metodología para la evaluación colaborativa de la usabilidad de sistemas software interactivos," Universidad del Cauca, 2015.
- [34] G. L. Kolschoten, R. O. Briggs, J. H. Appelman, and G.-J. De Vreede, "ThinkLets as Building Blocks for Collaboration Processes: A Further Conceptualization," *10th Collab. Res. Int. Work. Groupw.*, no. SEPTEMBER, pp. 137–152, 2004.

- [35] A. Solano, "Metodología para la evaluación Colaborativa De La Usabilidad De Sistemas," Universidad del Cauca, 2014.
- [36] R. O. Briggs, G.-J. De Vreede, J. F. . J. Nunamaker, and D. Tobey, "ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS)," *Proc. 34th Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 0, no. c, pp. 1–9, 2001.
- [37] A. Solano and C. Collazos, "Modelo para el diseño de actividades colaborativas desde un enfoque práctico," *Rev. Univ. RUTIC*, vol. 1, no. 2, pp. 55–64, 2012.
- [38] G. L. Kolfshoten, R. O. Briggs, J. H. Appelman, and G.-J. De Vreede, "ThinkLets as Building Blocks for Collaboration Processes: A Further Conceptualization," *10th Collab. Res. Int. Work. Groupw.*, pp. 137–152, 2004.
- [39] G. L. Kolfshoten, G. J. de Vreede, R. O. Briggs, and H. G. Sol, "Collaboration 'engineerability,'" *Gr. Decis. Negot.*, vol. 19, no. 3, pp. 301–321, 2010.
- [40] G. M. de Vreede, Gert Jan, Luis Guerrero, "Groupware: Design, Implementation, and Use," vol. 3198, p. 2004, 2004.
- [41] V. G.-J. Briggs, Robert O, *ThinkLets Collaboration: Building Block for concerted Collaboration*. 2001.
- [42] K. Sheingold, *Mirrors of Minds: Patterns of Experience in Educational Computing.*, no. 14. 1987.
- [43] "Beza, Olga. 'Gamification – How games can level up our everyday life.' VU University, Amsterdam (2011).," p. 2011, 2011.
- [44] A. García and O. Fabián, "ChildProgramming-G: Extendiendo ChildProgramming con Técnicas de Gamificación," Universidad del Cauca Facultad, 2014.
- [45] H. Orejuela Muñoz, H. F. O. Muñoz, A. García, J. A. H. Alegria, and C. A. C. Ordoñez, "Analizando y Aplicando la Gamificación en el Proceso ChildProgramming," *Rev. Colomb. Comput. - RCC*, vol. 14, no. 2, pp. 7–23, 2014.
- [46] "Sridharan, Mithun, Aditya Hrishikesh, and Lena Sunali Raj. 'An academic analysis of Gamification.' UX Magazine (2012): 1-13.," p. 2012, 2012.
- [47] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to Gamefulness: Defining 'Gamification,'" *Proc. 2011 Annu. Conf. Ext. Abstr. Hum. factors Comput. Syst. - CHI EA '11*, p. 2425, 2011.
- [48] P. Hagglund, "Taking gamification to the next level," p. 37, 2012.
- [49] "Lee, Joey J., and Jessica Hammer. 'Gamification in education: What, how, why bother?.' Academic exchange quarterly 15.2 (2011): 146.," vol. 2, p. 2011, 2011.
- [50] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir. Softw. Eng.*, 2009.
- [51] A. S. Alegría, Y. M. Alegría, and C. C. Ordoñez, "THINKLET: Key Element in the Collaborative Methods Generation for Evaluate Software Usability," *Cienc. e Ing. Neogranadina*, vol. 20, pp. 87–106, 2010.
- [52] R. Briggs, G. Kolfshoten, and V. Gert-Jan, "Defining key concepts for collaboration engineering," in *AMCIS 2006 Proceedings*, 2006, pp. 121–128.
- [53] G. L. Kolfshoten and G.-J. Vreede, "The Collaboration Engineering Approach for

- Designing,” *Groupw. Des. Implementation, Use*, vol. 4715, pp. 95–110, 2007.
- [54] L. Ma, J. D. Ferguson, M. Roper, M. Wood, and J. Wilson, “A collaborative approach to learning programming: a hybrid learning model,” in *6th Annual Higher Education Academy Subject Network for Information and Computer Science Conference*, 2005, pp. 75–80.
- [55] J. H. W. Mollocondo, Y. Mamani, R. Linares, M. Ibarra, “ChildThink: Propuesta de un Modelo Didáctico para la Enseñanza de la Programación en Niños,” in *isit2015*, 2015.
- [56] N. Omar and A. Jesús, “Métodos de Enseñanza de Algoritmos Centrado en dos Dimensiones,” *4. Simp. Int. - Sist. Telemáticos Organ. Intel.*, p. 16, 2009.
- [57] V. Tundjungsari, “E-learning model for teaching programming language for secondary school students in Indonesia.,” *13th Int. Conf. Remote Eng. Virtual Instrum.*, pp. 262–266, 2016.
- [58] a Pears *et al.*, “A survey of literature on the teaching of introductory programming,” *SIGCSE Bull.*, vol. 39, no. 4, pp. 204–223, 2007.
- [59] V. B. Barreto, A. L’Erario, and J. A. Fabri, “Ensino de Programacao para Alunos do Ensino Médio Utilizando o Robo Lego Mindstorms,” in *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*, 2015.
- [60] E. V Howard, D. Evans, C. Bishop-clark, D. Colton, and K. a Grant, “A qualitative look at Alice and pair-programming,” 2009.
- [61] J. Lin, L. Yen, M. Yang, and C. Chen, “Teaching computer programming in elementary schools: A pilot study,” *Natl. Educ. Comput. Conf.*, pp. 1–8, 2005.
- [62] S. Crook, “Embedding Scratch in the Classroom,” *Int. J. Learn. Media*, vol. 1, pp. 17–21, 2009.
- [63] M. Resnick *et al.*, “Scratch: Programming for All.,” *Commun. ACM*, vol. 52, pp. 60–67, 2009.
- [64] E. A. C. Bittner and J. M. Leimeister, “Why shared understanding matters - Engineering a collaboration process for shared understanding to improve collaboration effectiveness in heterogeneous teams,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2013, pp. 106–114.
- [65] H. Patel, M. Pettitt, and J. R. Wilson, “Factors of collaborative working: A framework for a collaboration model,” *Applied Ergonomics*. 2012.
- [66] G. L. Kolfshoten and G.-J. Vreede, “The Collaboration Engineering Approach for Designing,” in *Groupware: Design, Implementation, and Use*, vol. 4715, 2007, pp. 95–110.
- [67] A. Solano, C. César, and M. Yeny, “Evaluación de Usabilidad de Software desde una Perspectiva Colaborativa,” in *Conferencia Latinoamericana de Medios Audiovisuales en Red – LACNEM*, 2010.
- [68] J. Whitehead, “Collaboration in Software Engineering : A Roadmap Collaboration in Software Engineering : A Roadmap,” *Futur. Softw. Eng. FOSE-2007*, pp. 214–225, 2007.
- [69] C. A. Collazos, “Diseño de Actividades de Aprendizaje Colaborativo Asistidas por Computador,” *Rev. Educ. en Ing.*, vol. 9, no. 17, pp. 143–149, 2014.

- [70] L. Guerrero, R. Alarcón, and C. Collazos, "Indicadores de cooperación en el trabajo grupal," in *Memorias de la Conferencia Latinoamericana de Informática, CLEI 2000*, 2000, p. 10.
- [71] L. N. Aballay, M. B. Herrera, C. A. Collazos, and S. V. Aciar, "Medición de habilidades de trabajo en equipo en un modelo distribuido y colaborativo," *Rev. Tek.*, vol. 15, no. 1, p. x-y, 2015.
- [72] K. Huotari and J. Hamari, "Gamification" from the perspective of service marketing," *CHI 2011 Work. Gamification*, no. January, pp. 11–15, 2011.
- [73] H. Julio. Chimunja Ana, Collazos César, "Child Programming-C : Extendiendo Child Programming desde la perspectiva de la Ingeniería de la Colaboración," in *11 Congreso Colombiano de Computación.*, 2016.
- [74] A. M. Chimunja, C. A. Collazos, and J. A. Hurtado, "A. M. Chimunja, C. A. Collazos, J. A. Hurtado Recibido enero 19 de 2017 - Aprobado mayo 31 de 2017," no. 22, pp. 67–75, 2017.
- [75] M. Coto *et al.*, "Modelo Colaborativo y Ubicuo para apoyar los procesos de enseñanza-aprendizaje a nivel Iberoamericano," *Rev. Educ. a Distancia*, vol. 2, no. 48, pp. 7–22, 2015.
- [76] P. Deemer, G. Benefield, and C. Larman, "the Primer," *Development*, pp. 1–22, 2010.
- [77] L. J. Osterweil, "Software processes are software too," *Proc. 9th Int. Conf. Softw. Eng.*, vol. 3, pp. 2–13, 1987.
- [78] N. B. Moe and T. Dybå, "The use of an electronic process guide in a medium-sized software development company," *Softw. Process Improv. Pract.*, vol. 11, no. 1, pp. 21–34, 2006.
- [79] L. Scott, L. Carvalho, R. Jeffery, J. D'Ambra, and U. Becker-Kornstaedt, "Understanding the use of an electronic process guide," *Inf. Softw. Technol.*, vol. 44, no. 10, pp. 601–616, 2002.
- [80] "Wang, Y., King, G.A.: Software engineering processes: principles and applications. CRC Press, Boca Raton (2000)," p. 2000, 2000.
- [81] "I. 2. SQuaRE, «Software Product Quality Requirements and Evaluation – Quality Model.,» ISO, 2012.," p. 2012, 2012.
- [82] ISO, "ISO/IEC CD 25010: Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Software and Quality in Use Models (Resolution 937)," p. 42, 2008.
- [83] M. N. ri Mahrin, D. Carrington, and P. Strooper, "Investigating factors affecting the usability of software process descriptions," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5007 LNCS, pp. 222–233, 2008.
- [84] B. Curtis, M. I. Kellner, and J. Over, "Process modeling," *Commun. ACM*, vol. 35, no. 9, pp. 75–90, 1992.
- [85] "Kellner, M.I., Becker-Kornstaedt, U., Riddle, W.E., Tomal, J., Verlage, M.: Process guides: effective guidance for process participants. In: Proc. of the Fifth International Conference on the Software Process: Computer Supported Organizational Work, pp.

- 1," vol. 25, p. 1998, 1998.
- [86] B. E. F. Cross Patricia, Major Claire Howell, "Collaborative learning techniques : A Handbook for college faculty," p. 2005, 2005.
 - [87] N. I. Scagnoli, "Estrategias para motivar el aprendizaje colaborativo en cursos a distancia," *Investig. Cienc.*, vol. 14, no. 36, pp. 39–47, 2005.
 - [88] P. Dillenbourg, M. Baker, a. Blaye, and C. O'Malley, "The evolution of research on collaborative learning," *Learn. Humans Mach. Towar. an Interdiscip. Learn. Sci.*, pp. 189–211, 1996.
 - [89] E. Johnson y Johnson Johnson, D., Johnson, R., Holubec, *Cooperation in the classroom*. 1998.
 - [90] E. F. Barkley, C. H. Major, and K. P. Cross, *Collaborative learning techniques: A handbook for college faculty*. 2014.
 - [91] S. Burbano, C. Collazos, and U. Hernandez, "Monitoreo y evaluación del proce de colaboración en ambientes de aprendizaje colaborativo," Popayán, 2003.
 - [92] C. Collazos, L. Guerrero, J. Pino, and S. Ochoa, "Evaluating collaborative learning processes," *Groupw. Des. Implementation, Use*, vol. 2440, pp. 173–194, 2002.
 - [93] L. A. Guerrero, R. Alarcón, C. Collazos, J. A. Pino, and D. A. Fuller, "Evaluating cooperation in group work," *Proc. - 6th Int. Work. Groupware, CRIWG 2000*, no. February, pp. 28–35, 2000.