

UN MODELO PARA LA ENSEÑANZA DE LA PROGRAMACIÓN DE SOFTWARE EN NIÑOS A TRAVÉS DE ESTRATEGIAS COLABORATIVAS



**Samith Tatiana Cruz Sánchez
Oscar Eduardo Rojas Ordoñez**

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

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo IDIS – Investigación y Desarrollo en Ingeniería de Software
Línea de Investigación en Ingeniería del Software
Popayán, Abril 2013**

UN MODELO PARA LA ENSEÑANZA DE LA PROGRAMACIÓN DE SOFTWARE EN NIÑOS A TRAVÉS DE ESTRATEGIAS COLABORATIVAS



Monografía de Trabajo de Grado para Optar al Título de
Ingeniero de Sistemas

Samith Tatiana Cruz Sánchez
Oscar Eduardo Rojas Ordoñez

Director: PhD. Julio Ariel Hurtado Alegría
Co-Director: PhD. César Alberto Collazos Ordoñez

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo IDIS – Investigación y Desarrollo en Ingeniería de Software
Línea de Investigación en Ingeniería del Software
Popayán, Abril 2013

NOTA DE ACEPTACIÓN

PRESIDENTE DEL JURADO

JURADO

Popayán, Abril 2013

AGRADECIMIENTOS

Quiero agradecer a Dios por ser el gran arquitecto de mi vida, mi guía y mi fortaleza en el camino.

A mi madre Ana Lucia por ser mi impulso hacia este logro, mi compañía, mi gran apoyo, mi refugio, mi gran amor; a mi padre Carlos Alberto quien desde la distancia siempre me apoyo y creyó en mí.

A mis sobrinos Jennifer y Santiago por llenar mi corazón de alegría en los momentos de dificultad.

A mis amigos por permanecer siempre a mi lado y mantener para mí una palabra en el momento preciso, a mis compañeros por todos los momentos compartidos durante este largo periodo de aprendizaje.

A mis profesores del departamento de sistemas por toda la paciencia, la dedicación y sobretodo el conocimiento replicado en mí, especialmente a la Ingeniera Sandra Roa y el Ingeniero Erwin Meza por sus aportes valiosos y significativos para este trabajo.

A mis tutores Ingeniero Julio Ariel Hurtado, Ingeniero César Collazos por su confianza, por su entrega y su gran disposición para el logro de este trabajo.

A la Universidad del Cauca brindarme los espacios para hacer de mí una excelente profesional.

SAMITH TATIANA CRUZ SÁNCHEZ

A Dios por ser el centro de mi existencia, haberme permitido llegar hasta este punto llenándome de fortaleza y sabiduría para lograr mis objetivos,

A mis padres por ser la base fundamental en todo lo que soy, en mi educación, tanto académica, como de la vida y por su incondicional apoyo durante este largo tiempo,

A mi novia por ser parte de mi vida, brindándome su amor, su gran paciencia y un apoyo más para continuar superando y alcanzando mis metas,

A mis profesores y tutores por aportarme sus conocimientos en mi formación profesional,

A mis compañeros y amigos por el valioso tiempo que compartieron junto a mí durante este proceso de formación,

A todos los demás muchas gracias.

OSCAR EDUARDO ROJAS ORDOÑEZ

TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	2
1.2 OBJETIVOS DEL PROYECTO	3
1.2.1 <i>Objetivo General</i>	3
1.2.2 <i>Objetivos Específicos</i>	3
1.3 RESULTADOS OBTENIDOS.....	4
1.4 ORGANIZACIÓN DEL DOCUMENTO	4
2. MARCO TEORICO Y ESTADO DEL ARTE	5
2.1 MODELOS PEDAGOGICOS	5
2.1.1 <i>Pedagogía Activa: Método Montessori</i>	6
2.1.2 <i>Modelo Constructivista: Teoría de Jean Piaget</i>	6
2.1.3 <i>Pedagogía Conceptual</i>	7
2.1.4 <i>Aprendizaje Significativo</i>	7
2.1.5 <i>Teoría Sociocultural: Teoría de Lev Vygotsky</i>	8
2.2 TRABAJO GRUPAL.....	8
2.2.1 <i>Beneficios del trabajo grupal</i>	9
2.2.2 <i>El Trabajo en Grupo en las Aulas de Clase</i>	9
2.3 APRENDIZAJE COLABORATIVO	10
2.3.1 <i>El Aprendizaje Colaborativo desde la Parte Cognitiva</i>	10
2.3.2 <i>Aportes del Aprendizaje Colaborativo</i>	11
2.3.3 <i>Elementos Básicos del Aprendizaje Colaborativo</i>	11
2.3.4 <i>Técnicas de Aprendizaje Colaborativo</i>	12
2.4 INGENIERIA DE COLABORACIÓN.....	13
2.4.1 <i>Patrones de Colaboración</i>	13
2.4.2 <i>Thinklets</i>	13
2.5 METODOLOGIAS AGILES DE DESARROLLO DE SOFTWARE.....	14
2.5.1 <i>Manifiesto Ágil</i>	16
2.5.2 <i>Revisión de Metodologías</i>	17
2.5.2.1 <i>Programación Extrema (XP)</i>	17
2.5.2.2 <i>Scrum</i>	19
2.5.2.3 <i>Crystal Methodologies</i>	20
2.5.2.4 <i>Dinamic Systems Development Method (DSDM)</i>	21
2.5.2.5 <i>Adaptative Software Development (ASD)</i>	21
2.5.2.6 <i>Feature Driven Development (FDD)</i>	22
2.5.2.7 <i>Evolutionary Project Management (EVO)</i>	22
2.6 AMBIENTE DE DESARROLLO	22
2.6.3 <i>Desde la perspectiva de la Colaboración</i>	22
2.6.4 <i>Desde la perspectiva del aprendizaje</i>	23
2.6.2 <i>Desde la perspectiva del Desarrollo de Software</i>	23
2.6.4 <i>Desde la perspectiva de la Programación de Software</i>	23
2.7 ESTADO DEL ARTE.....	23
2.7.1 <i>Uso de Estrategias de Programación en Jardines Infantiles y Escuelas</i>	24

2.7.2 Una mirada Cualitativa a la Programación en Pareja y Alice	24
2.7.3 Niños Tecnologías Cuentacuentos: Las Diferencias en Elaboración y Recuerdos	24
2.7.4 Los Grados de Comprensión: Comprensión de los Niños de un Entorno de Programación Visual..	25
2.7.5 La enseñanza de Programación de Computadoras en las Escuelas de Primaria: Un Estudio Piloto	25
2.7.6 Fuentes Divertidas: Presentación de 4to y 5to Grado de Programación Usando Squeak	26
2.7.7 Experiencias con Scratch en el Aula	26
2.7.8 Scratch en Colombia	27
2.7.9 Impacto de la Programación con Scratch en estudiantes con su propio proceso de aprendizaje..	27
2.8 SÍNTESIS Y DISCUSIÓN	29
3. EL MODELO CHILD PROGRAMMING	30
3.1 CONTEXTO DEL MODELO PROPUESTO CHILD PROGRAMMING	30
3.1.1 Análisis de Modelos Estudiados	31
3.1.2 Modelo Seleccionado	32
3.1.3 Diagnóstico Inicial y Caracterización	34
3.2 ARQUITECTURA CONCEPTUAL DEL MODELO CHILD PROGRAMMING	35
3.2.1 Marco Conceptual del Modelo Child Programming	36
3.2.2 Actores del Modelo Child Programming	41
3.2.3 Dimensiones del Modelo Child Programming	42
3.2.4 Elementos Child (Child Elements).....	43
3.2.4.1 Roles	43
3.2.4.2 Conceptos	44
3.2.4.3 Prácticas.....	46
3.2.4.4 Especificación de las prácticas	47
3.3 PROCESO CHILD (CHILD PROCESS)	49
3.3.1 Pre - Juego.....	50
3.3.2 Juego	50
3.3.3 Post – Juego	51
3.4 HERRAMIENTAS DE SOPORTE CHILD	52
3.4.1 Modelado en EPFC (Eclipse Process Framework Composer)	52
3.4.2 Descripción Sitio Web ChildProgramming	54
3.4.3 Descripción Ambiente de Desarrollo ChildProgramming	55
3.4.3.1 Ambiente de Desarrollo.....	55
3.4.3.2 Ambiente de Programación	56
3.4.3.3 Ambiente Virtual.....	57
4. EXTRACCIÓN Y EVALUACIÓN DEL MODELO CHILD PROGRAMMING: ESTUDIOS DE CASO	59
4.1 METODOLOGIA.....	59
4.1.1 Instrumentos de Evaluación.....	60
4.2 CONTEXTO DE LA INVESTIGACIÓN	61
4.3 ESTUDIOS DE CASO	62
4.3.1 ESTUDIO DE CASO 1	62
4.3.1.1 Pregunta de Investigación	62
4.3.1.2 Objetivo del Estudio.....	63
4.3.1.3 Selección del Estudio	63
4.3.1.4 El Estudio de Caso y los Sujetos de Investigación	63
4.3.1.5 Indicadores y Mediciones	64
4.3.1.6 Ejecución del Estudio de Caso	66

4.3.1.7 Resultados Cuantitativos	66
4.3.1.8 Resultados Cualitativos.....	71
4.3.1.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso	71
4.3.1.9 Amenazas de Validez	71
4.3.2 Estudio de CASO 2.....	72
4.3.2.1 Pregunta de Investigación	72
4.3.2.2 Objetivo del Estudio.....	72
4.3.2.3 Selección del Estudio	72
4.3.2.4 El Estudio de Caso y los Sujetos de Investigación	73
4.3.2.5 Indicadores y Mediciones	73
4.3.2.6 Ejecución del Estudio	74
4.3.2.7 Resultados Cuantitativos	75
4.3.2.8 Resultados Cualitativos.....	81
4.3.2.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso	81
4.3.2.9 Amenazas de Validez	82
4.3.3 Estudio de Caso 3.....	83
4.3.3.1 Pregunta de Investigación	83
4.3.3.2 Objetivo del Estudio.....	83
4.3.3.3 Selección del Estudio	83
4.3.3.4 El Estudio de Caso y los Sujetos de Investigación	84
4.3.3.5 Indicadores y Mediciones	84
4.3.3.6 Ejecución del Estudio	85
4.3.3.7 Resultados Cuantitativos	86
4.3.3.8 Resultados Cualitativos.....	93
4.3.3.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso	93
4.3.3.9 Amenazas de Validez	94
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	96
5.1 CONCLUSIONES	96
5.2 RECOMENDACIONES	98
5.3 TRABAJO FUTURO.....	98
BIBLIOGRAFIA	100

LISTA DE TABLAS

Tabla 1. Clasificación de Trabajos Relacionados con Child Programming	29
Tabla 2. Comparativo de Modelos Pedagógicos Estudiados.....	32
Tabla 3. Relación de Conceptos y Prácticas relevantes en cada Modelo.....	33
Tabla 4. Relación de Clases Child Programming	41
Tabla 5. Comparativo de Roles Scrum - Child Programming. Fuente Propia	43
Tabla 6. Relación de Conceptos Definidos por Child Programming	46
Tabla 7. Relación de Prácticas definidas por Child Programming	46
Tabla 8. Indicadores y Mediciones Estudio de Caso 1	64
Tabla 9. Conjunto de “ <i>Prácticas Empíricas Child</i> ”	70
Tabla 10. Registro de Frecuencias	70
Tabla 11. Indicadores y Mediciones Estudio de Caso 2	74
Tabla 12. Conjunto de Prácticas Observadas en el Estudio.....	80
Tabla 13. Conjunto de Nuevas Prácticas Potenciales “ <i>Prácticas Child</i> ”	80
Tabla 14. Análisis Cualitativo Equipos Experimentales y Equipos de Control	81
Tabla 15. Indicadores y Mediciones Estudio de Caso 3	84
Tabla 16. Prácticas Resultantes ChildProgramming	92
Tabla 17. Análisis Cualitativo Equipos Experimentales y Equipos de Control – Estudio de Caso 3	93

LISTA DE FIGURAS

Figura 1. Modelos Pedagógicos.....	5
Figura 2. Distribución del riesgo en un desarrollo ágil.....	15
Figura 3. Relación de Prácticas en XP [34].....	19
Figura 4. Arquitectura Conceptual de ChildProgramming [57].....	30
Figura 5. Arquitectura Child Programming Propuesta	35
Figura 6. Modelo Conceptual Child Programming.....	36
Figura 7. Definición Conceptual de la Clase Child	37
Figura 8. Definición Conceptual de la Clase Team	37
Figura 9. Definición Conceptual de la Clase Collaborative Process	38
Figura 10. Definición Conceptual de la Clase Facilitador	38
Figura 11. Definición Conceptual de la Clase Abstrata ChildLetElement.....	38
Figura 12. Definición Conceptual de la Clase ChildLet Task.....	39
Figura 13. Definición Conceptual de la Clase Practice.....	39
Figura 14. Definición Conceptual de la Clase Concept	40
Figura 15. Definición de la Clase Role	40
Figura 16. Definición de la Clase Thinklet.....	40
Figura 17. Definición de la Clase Collaborative Pattern	41
Figura 18. Composición Componente Cognitivo- Fuente Propia.....	45
Figura 19. Prácticas Child Programming- Fuente Propia	46
Figura 20. Plantilla Práctica Cognitiva.....	47
Figura 21. Plantilla Práctica Ágil	48
Figura 22. Plantilla Práctica Colaborativa.....	49
Figura 23. Ciclo de Vida propuesto por Child Programming. Fuente Propia.....	49
Figura 24. Diagrama de Flujo Proceso Child Programming. Fuente Propia	52
Figura 25. Contenido del Método Asociado al Proceso Colaborativo Child.....	53
Figura 26. WBS ChildProgramming	54
Figura 27. Captura Usuario Anónimo - Página de Inicio.....	55
Figura 28. Ambiente de Desarrollo ChildProgramming. Fuente Propia	55
Figura 29. Adaptación del Entorno de Programación ChPScratch	57
Figura 30. Ambiente Virtual de Desarrollo ChildProgramming	58
Figura 31. Procedimiento Metodológico Seguido para esta Investigación – Basado en Yin Fuente Propia	60
Figura 32. Instrumentos de Evaluación- Fuente Propia.....	61
Figura 33. Gráfico de Distribución de Comportamiento por Equipos	67
Figura 34. Gráfico de Distribución de Productividad de los Equipos	68
Figura 35. Gráfico de Distribución de Calidad de los Equipos.....	68
Figura 36. Frecuencias de Organización y Dinámicas de Equipo	69
Figura 37. Imágenes de diferentes Equipos en el Desarrollo de la Actividad 1.	71
Figura 38. Distribución de Comportamiento Equipos Experimentales y de Control.....	75
Figura 39. Gráfico de Distribución de Productividad de Equipos Experimentales.....	76
Figura 40. Gráfico de Distribución de Calidad de Equipos Experimentales	77

Figura 41. Distribución de Adopción de Prácticas Equipos Experimentales y de Control .	78
Figura 42. Gráfico de Dispersión Adopción Vs Comportamiento.....	78
Figura 43. Gráfico de Dispersión Adopción Vs Productividad	78
Figura 44. Gráfico de Dispersión Adopción Vs Calidad.....	79
Figura 45. Imágenes de Diferentes Equipos en el Desarrollo de la Actividad 2 – Fases 1, 2, 3 de Jigsaw.....	82
Figura 46. Imágenes de Diferentes Equipos en el Desarrollo de la Actividad 2 – Fases 4 y 5 de Jigsaw.....	82
Figura 47. Distribución de Comportamiento Equipos Experimentales y de Control Estudio de Caso 3	86
Figura 48. Gráfico de Distribución de Productividad de Equipos Experimentales Estudio de Caso 3	87
Figura 49. Gráfico de Distribución de Calidad de Equipos Experimentales Estudio de Caso 3	88
Figura 50. Distribución de Adopción de Prácticas Equipos Experimentales y de Control Estudio de Caso 3	88
Figura 51. Distribución de Adopción de Nuevas Prácticas Equipos Experimentales y de Control Estudio de Caso 3	89
Figura 52. Gráfico de Dispersión	90
Figura 53. Gráfico de Dispersión	90
Figura 54. Gráfico de Dispersión Adopción de Nuevas Prácticas – Calidad.....	90
Figura 55. Regiones de la Gráfico de Dispersión	91
Figura 56. Imágenes Proceso de Desarrollo de los Equipos – Estudio de Caso 3	94
Figura 57. Imágenes Proceso de Desarrollo de los Equipos - Estudio de Caso 3.....	94

1. INTRODUCCIÓN

La tecnología es parte fundamental de la vida cotidiana de las personas del mundo actual, especialmente para los niños que han nacido al momento se vienen presentando grandes avances en el área de las TIC. El diseño de tecnologías para los niños ha recibido un aumento de atención en los últimos años [1]. Durante este tiempo se ha incrementado notoriamente la interacción de los niños con las tecnologías de software y la inclusión de éstas en su vida cotidiana, a través de dispositivos de comunicación como el celular, de entretenimiento como los juegos individuales o grupales, las tecnologías de la educación, entre otros, los cuales garantizan inclusión, entretenimiento y aprendizaje. Actualmente, se está pasando del aprendizaje tradicional en la escuela, de los juegos y las experiencias cotidianas a nuevas experiencias con tecnología de última generación, haciendo de los niños uno de los principales grupos de usuarios consumidores de estos nuevos cambios [2], transformándolos en una población de usuarios frecuentes y con experiencia [3].

En los últimos años han surgido metodologías para el desarrollo de nuevas tecnologías y son los usuarios quienes se han convertido en miembros activos, en verificadores o participantes en investigaciones y es importante resaltar como los niños se están convirtiendo en piezas fundamentales de las investigaciones en desarrollo de tecnología [4]. De acuerdo a Druin [4], a menudo surgen preguntas como: ¿Son los niños capaces de contribuir a través del proceso de investigación y desarrollo?, y la respuesta es quizá más significativa, los niños sí deben contribuir y de hecho pueden ser miembros o estar vinculados en un equipo de experiencia en investigación. Al igual que los científicos en el campo de la computación o los educadores que aportan según sus experiencias, los niños también pueden partir desde sus propias experiencias para cumplir en función de lo que necesita lograr un equipo [4]. Un ejemplo de ello son los equipos inter-generacionales [5] que han incluido miembros de diferentes edades, disciplinas y experiencias, para lo cual los niños se han convertido en piezas fundamentales como los educadores, los científicos en computación y los artistas [4].

Una instancia particular de la pregunta anterior es ¿Serían los niños capaces de contribuir al desarrollo de software? Aunque la mayoría de los niños que utilizan computadores, no son programadores, a muchos de ellos les encantaría programar su propio software y personalizar el que tienen [2]. Por lo general desean realizar programas que son similares a los que utilizan tales como juegos o aplicaciones educativas que son altamente interactivas y de riqueza gráfica. Aunque para los niños, las ideas sobre la programación no son muy simples, su aprendizaje se convierte en una herramienta poderosa para desarrollar sus habilidades lógico-matemáticas y sociales [2]. Sin embargo, para este tipo de tareas las herramientas adecuadas para su edad no son suficientes y no están totalmente soportadas por metodologías que involucren a los niños en un verdadero desarrollo de software [2].

En esta dirección, se presenta el proyecto “Un Modelo para la Enseñanza de la Programación de Software en Niños a través de Estrategias Colaborativas” cuyo objetivo

es formalizar un modelo para soportar el desarrollo de software orientado a niños a partir de un conjunto de experiencias prácticas que reúnen dinámicas colaborativas y ágiles establecidos por los equipos de trabajo conformados por niños.

1.1 PLANTEAMIENTO DEL PROBLEMA

El tema de la programación de software en los niños ha sido durante mucho tiempo una controversia y una polémica en el ámbito de la educación. Las discusiones en el área se han centrado en cuestiones como la forma de crear lenguajes para programación con niños, o si los niños pueden aprender temas particulares en la programación [6].

Aunque la tarea no es fácil, dado que el hablar de instrucciones y secuencias como núcleo central de la programación no es simple y menos para los niños quienes tienen un paradigma o un modelo mental propios en su forma de pensar y ver las cosas. En cierta medida, los niños apenas están adquiriendo las nociones de sistemas de numeración y hacen un esfuerzo enorme con la simbología del lenguaje [7], por ello esperar que se enfrenten a los sistemas simbólicos necesarios para afrontar programas de computador, es más difícil aún.

De allí, que enseñar programación a niños no solo se vuelve interesante por sí misma, sino que la didáctica y los resultados psicopedagógicos de ésta enseñanza pueden influenciar positivamente el desarrollo de las capacidades lógicas del niño. De acuerdo a expertos en el campo de la educación, la actividad de la programación, además del interés que tiene como instrumento para la resolución de problemas de diversa índole, puede ser un poderoso recurso didáctico, no solo para el aprendizaje de conceptos y técnicas de campo específico de la informática, sino para el desarrollo de destrezas más generales y valiosas de tipo heurístico [8].

Además, a edades tempranas los niños manejan mucha información teniendo en cuenta la relación que guardan unos con otros en sus diversos espacios de juego y esparcimiento, es decir, se produce un constante intercambio de ideas e información en diversas disciplinas [9], este patrón de interacción es obviamente más fácil de aplicar en educación primaria, generando en ellos fuertes sentimientos de pertenencia y cohesión, permitiéndoles identificar metas comunes y atribuciones compartidas lo que los hace sentirse parte de, estimulando su productividad y responsabilidad influyendo directamente en su autoestima y desarrollo. De aquí la importancia de resaltar el trabajo colaborativo el cual juega también un papel preponderante en el aprendizaje de los niños basado en estas características.

Por ello se plantea para este proyecto la siguiente pregunta de investigación: ***¿Cómo guiar a un equipo pequeño (5±2) de niños en edades comprendidas entre los 8 y 10 años en el desarrollo de una solución software en forma efectiva?***

Para dar respuesta a la pregunta anterior, se planteó el presente trabajo de grado el cual plantea un modelo que reúne una serie de experiencias que han permitido conocer, estudiar, formalizar y evaluar las dinámicas que emergen en equipos de desarrollo de software conformados por niños.

Este proyecto implicó un trabajo de investigación, en el que se han aplicado conceptos de ingeniería de software, específicamente el desarrollo de software basado en metodologías ágiles de desarrollo, ingeniería de la colaboración específicamente conceptos de patrones de colaboración y thinklets, aprendizaje colaborativo, entornos de programación diseñados para niños, entre otras áreas básicas que resultaron necesarias para la investigación y formulación del modelo.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Obtener y formalizar 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.

1.2.2 Objetivos Específicos

- Caracterizar el modelo cognitivo de escolares entre 8 y 10 años de la población de estudio, así como las dinámicas individuales y de grupo emergentes del trabajo en equipo.
- Extraer un modelo inicial de acuerdo a la literatura existente en métodos ágiles, modelos de aprendizaje colaborativo y un caso de estudio preliminar.
- Formular incrementalmente el modelo de desarrollo software emergente a partir de la maduración del modelo inicial a través de casos de estudio.
- Evaluar incrementalmente el modelo a través de un conjunto de experiencias con niños en la Institución Educativa Técnico Industrial sede San Camilo de la ciudad de Popayán.
- Adaptar un ambiente colaborativo de desarrollo software que brinde soporte al modelo.

1.3 RESULTADOS OBTENIDOS

- **MONOGRAFÍA DE GRADO:** este documento describe detalladamente el proceso llevado a cabo para el desarrollo del proyecto, la o las herramientas seleccionadas, las experiencias de campo en las que se construyó y aplicó el modelo, los resultados, los problemas que se presentaron, las respectivas soluciones, los principales aportes, las conclusiones y recomendaciones para el desarrollo de futuras investigaciones.
- **ANEXOS:** este documento contiene los aportes necesarios y complementarios del proyecto no incluidos en la monografía.
- **ARTICULOS INVESTIGATIVOS:** (3) documentos que resumen los principales características y productos del proyecto, uno de ellos publicado en la “Revista Universitaria en Telecomunicaciones, Informática y Control” – RuTic de la Universidad del Cauca y dos que serán publicados en una revista indexada o presentados en un evento nacional y/o internacional.
- **Modelo de Proceso especificado en SPEM,** el cual ha sido especificado en la herramienta EPF (Eclipse Process Framework).
- **SITIO WEB:** la documentación principal arrojada por esta investigación estará alojada en el sitio web <http://www.unicauca.edu.co/childprogramming> como soporte para futuras investigaciones.

1.4 ORGANIZACIÓN DEL DOCUMENTO

El documento se encuentra organizado de la siguiente manera:

Capítulo 2: incluye el marco teórico y el estado del arte relacionados con el desarrollo de software con niños a partir de herramientas y prácticas colaborativas en el aula.

Capítulo 3: presenta el modelo ChildProgramming para la adaptación al entorno académico de estudio, teniendo en cuenta las dimensiones definidas para su especificación.

Capítulo 4: presenta los resultados obtenidos, a partir la aplicación de estudios de caso de donde ChildProgramming fue extraído y evaluado.

Capítulo 5: presenta las conclusiones, recomendaciones y trabajos futuros planteados sobre la tesis.

2. MARCO TEORICO Y ESTADO DEL ARTE

Este capítulo contiene las bases teóricas sobre las cuales se encuentra enmarcado el presente proyecto, estas contemplan el trabajo grupal, el aprendizaje colaborativo, la ingeniería de la colaboración, las metodologías ágiles de desarrollo. En esta etapa se definen los núcleos temáticos para esta investigación.

2.1 MODELOS PEDAGOGICOS

Los modelos pedagógicos representan formas particulares de interrelación entre los parámetros pedagógicos: metas, contenidos de enseñanza, relación profesor – alumno, métodos y concepto de desarrollo, además son representaciones esenciales de las corrientes pedagógicas [68].

Existen diferentes clasificaciones de modelos pedagógicos como los describe la figura 1.

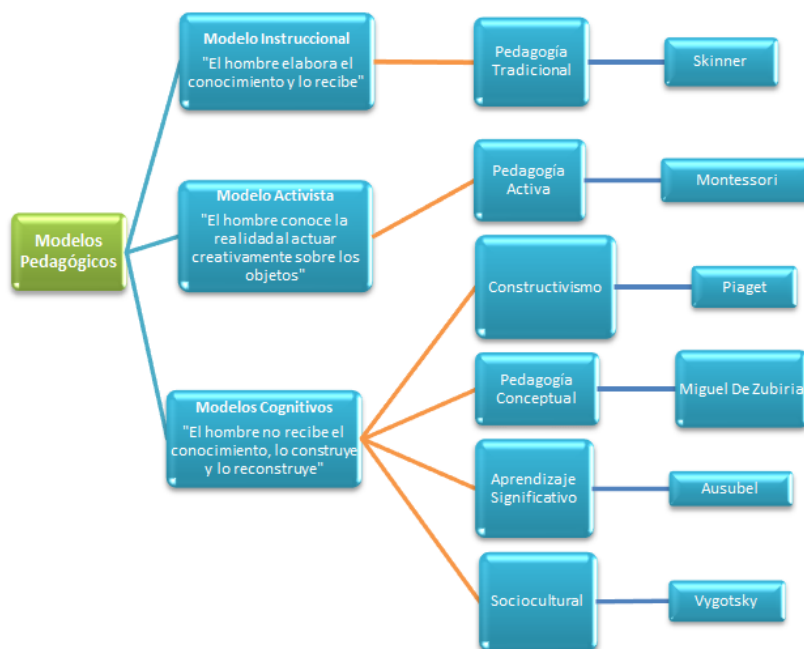


Figura 1. Modelos Pedagógicos¹

Como parte de esta investigación hemos considerado estudiar los modelos Activistas y Cognitivos, dada la relevancia de estos en la construcción de conocimiento a partir de experiencias propias de los individuos.

¹ <http://www.joaquinparis.edu.co/DATA/MODELOS/PAGINAS/abc.htm#CONCEPTUAL>

A continuación presentamos una descripción de cada uno de los modelos en consideración, citando del modelo activista el postulado de María Montessori², y de los modelos cognitivos los postulados de Jean Piaget³, Miguel De Zubiria⁴, David Ausubel⁵ y Lev Vygotsky⁶.

2.1.1 Pedagogía Activa: Método Montessori

El Método Montessori postula que los niños aprenden a hablar, leer y escribir de la misma manera como lo hacen al gatear, caminar, correr de forma espontánea [69]. Este postulado permite llegar a descubrir y ayudar a los niños para alcanzar su potencial como ser humano, a través de los sentidos, un ambiente preparado y utilizar la observación científica de un profesor entrenado [70].

Cada método desarrollado, se basó en sus observaciones, en lo que los niños hacían naturalmente, por si mismos sin ayuda de adultos, por tanto este método va hacia la capacidad del educador de armar y respetar al niño como persona y ser sensible a sus necesidades. El educador ejerce una figura de guía, que potencia o propone desafíos, cambios y/o novedades, el ambiente no incita a la competencia entre compañeros, en cambio se respeta y valora el logro de cada alumno en su momento y ritmo oportuno [69].

Los principios básicos fundamentales de la Pedagogía Montessori son: la libertad, la actividad y la individualidad. El método está inspirado en el humanismo integral, postulando la formación de los seres humanos como personas únicas y plenamente capacitadas para actuar con libertad, inteligencia y dignidad.

El método propone períodos sensibles de la edad en que los niños demuestran capacidades inusuales en adquirir habilidades particulares, es decir, cuando el interés de los niños se focaliza a una parte específica de su ambiente

2.1.2 Modelo Constructivista: Teoría de Jean Piaget

El modelo constructivista ha repercutido en la educación y ha motivado a muchos cambios en el diseño curricular. Este modelo permite a los alumnos construir conocimiento en lugar de recibirlo de forma pasiva como una fiel copia de un libro o de un profesor específicamente. Este modelo sugiere que los alumnos deben construir representaciones

² **María Montessori:** (1870 - 1952). Educadora, Científica, Médica, Psiquiatra, Filósofa, Psicóloga. Italiana.

³ **Jean Piaget:** (1896 - 1980). Epistemólogo, Psicólogo, Biólogo. Suizo.

⁴ **Miguel De Zubiria:** (1951). Psicólogo. Fundador y Director Científico de la Fundación Internacional de Pedagogía Conceptual Alberto Merani. Colombiano

⁵ **David Ausubel:** (1918 - 2008). Psicólogo y Pedagogo. Estadounidense.

⁶ **Lev Vygotsky:** (1896 - 1934). Psicólogo. Ruso.

del aprendizaje, haciéndolo suyo y desarrollando el conocimiento desde sus propias palabras y experiencias de manera que les sea útil para sus vidas cotidianas[71].

“La actitud pedagógica constructivista debe preocupar que los alumnos no sean receptores conformistas sino agentes activos de su propia educación” [72].

Jean Piaget fue de los primeros psicólogos que postularon la teoría de que nacemos como procesadores de información, activos y exploratorios y construimos nuestro conocimiento en lugar de tomarlo ya hecho en respuesta a la experiencia o la instrucción.

La teoría de Piaget propone que desde el nacimiento a la madurez nuestros procesos de pensamiento cambian de manera radical, pero de forma lenta para no imponer un sentido al mundo; identificó cuatro factores que pueden influir en los cambios del pensamiento: la maduración, la actividad, las experiencias sociales y el equilibrio [73].

Piaget propuso cuatro etapas de desarrollo cognitivo, que se conocen como: periodo sensorio-motor (de 0 a 2 años de edad), el periodo pre-operacional (de 2 a 7 años de edad), el periodo de las operaciones concretas (de 7 a 12 años de edad) y el periodo de las operaciones formales (de 12 años a la adultez).

2.1.3 Pedagogía Conceptual

Propuesta por Miguel De Zubiria Samper validada a través de la Fundación Alberto Merani, es un enfoque pedagógico que tiene como propósito formar individuos plenos afectivamente (apasionados, alegres, amorosos), cognitivamente (brillantes) y expresivamente (talentosos). Este modelo, establece una clara definición entre Instrumento del Conocimiento y Operaciones Intelectuales, por lo que actúa deliberadamente en la formación de cada uno de ellos.

La teoría de la Pedagogía Conceptual se fundamenta en la teoría evolutiva de la mente humana, que la explica como unidad dinámica con tres sistemas: afectivo, cognitivo y expresivo, armados por instrumentos y operaciones mentales, para potenciar la inteligencia de los estudiantes. Como método educativo aplica el modelo del hexágono el cual identifica en todo acto educativo seis componentes secuenciales: propósito, enseñanza, evaluación, secuencia, didáctica y recursos [74].

2.1.4 Aprendizaje Significativo

El aprendizaje significativo de Ausubel se caracteriza como un punto de vista de transmisión, el factor más importante que influye en el aprendizaje significativo de cualquier idea nueva en el estado de la estructura cognitiva del individuo existen en el momento del aprendizaje. Ausubel enfatiza en la enseñanza de cuerpos de conocimientos

organizados, estructurados alrededor de conceptos clave y sugiere formas en que los profesores podrían estructurar el contenido para sus estudiantes.

Esta teoría radica en presentar el material en forma que animen a los alumnos a darle el sentido relacionándolo con lo que ya conocen. Los alumnos aprenden significativamente cuando encuentran sentido a lo que aprenden y este sentido se encuentra cuando son capaces de establecer una relación entre el nuevo conocimiento y que ya posee [75].

2.1.5 Teoría Sociocultural: Teoría de Lev Vygotsky

La teoría de Lev Vygotsky, que su teoría del desarrollo cognitivo ocurre a partir de las conversaciones e intercambios que el niño sostiene con miembros más conocedores de la cultura, adultos o compañeros más capaces. En este sentido los maestros son considerados mediadores para que el alumno tenga la capacidad de integrarse al grupo, son considerados guías que ofrecen información y apoyo necesario para que el niño crezca de manera intelectual [75].

Vygotsky como parte de su teoría describe la “zona de desarrollo próximo” definida como la distancia entre el nivel actual de desarrollo de un alumno determinado por su capacidad de resolver individualmente un problema y su nivel de desarrollar potencial, determinado por la posibilidad de resolver un nuevo problema cognitivo con la ayuda de una persona con más capacidad.

La importancia del aprendizaje colaborativo está sustentada en la teoría de Lev Vygotsky, donde plantea que en la medida que un sujeto se mueva de su nivel real actual o un posible potencial inmediato, hay adquisición de conocimientos, apropiación de habilidades e incorporación de actitudes y valores y por tanto existirá ahí educación y desarrollo. Así, el trabajo en equipo facilita la identificación por parte del profesor y también de los miembros de cada equipo de la zona de desarrollo potencial de cada uno y del equipo en su totalidad.

2.2 TRABAJO GRUPAL

El hombre es un ser social por naturaleza, es necesario reconocer y fomentar esta característica fundamental para el desarrollo de cualquier actividad humana de tipo cooperativa [10]. Particularmente, en las organizaciones, 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 con sus aportes contribuyen a lograr mejores resultados [11]. 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 [12].

2.2.1 Beneficios del trabajo grupal

De acuerdo a Turban [12], el trabajo grupal cuenta con una serie de ventajas sobre el trabajo individual:

- **Un grupo entiende mejor un problema que una sola persona:** los diferentes puntos de vista que pueden darse en un grupo de individuos suele ser mejor que la visión de una sola persona.
- **Existe una responsabilidad compartida:** el trabajo realizado por el grupo no recae en un solo individuo, todos los integrantes del grupo son los responsables del resultado final y de las decisiones que se tomen como grupo.
- **Facilita la detección de errores:** la perspectiva de más de una persona sobre algo siempre tendrá mayores puntos de juicio y evaluación que el de una sola persona que puede pasar por alto ciertos detalles importantes.
- **Conocimiento grupal:** un grupo presenta un mayor conocimiento que una sola persona, lo cual ofrece mejores alternativas para la resolución de problemas.
- **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.

2.2.2 El Trabajo en Grupo en las Aulas de Clase

El trabajo en grupo trasladado a las aulas escolares, incrementa la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos. Las enormes posibilidades que aportan las interacciones del alumnado como fuente de construcción de conocimiento es enorme, muchos autores como Vygotsky, plantean que los alumnos pueden aprender más y mejor si se les permite afrontar juntos los procesos de aprendizaje, sobre todo cuando se les brindan objetivos a los que deben llegar trabajando como un equipo [13]. Además, el trabajo en grupo posibilita la realización social creando un clima de confianza que favorece la incorporación de todos los alumnos y crea condiciones que sirven de soporte fundamental no solo para el desarrollo de habilidades intelectuales, sino también, para la formación de valores éticos [14].

Actualmente en la educación, el trabajo en grupo juega un papel importante como ingrediente esencial en todas las actividades de enseñanza y aprendizaje. Se puede afirmar que todos los proyectos que utilizan métodos o técnicas de enseñanza y aprendizaje innovadoras incorporan en los niños habilidades y experiencias en las que aprenden y al igual se forman como personas [18].

2.3 APRENDIZAJE COLABORATIVO

El aprendizaje colaborativo es un enfoque que trata de conformar pequeños grupos de tal forma que los estudiantes trabajen juntos para maximizar su propio aprendizaje y el de los demás, basado en la colaboración de sus integrantes. Esta clase de aprendizaje no se antepone al trabajo individual, ya que puede observarse como una estrategia de aprendizaje complementaria que permite fortalecer el desarrollo global del alumno. Una recopilación de estudios sobre aprendizaje colaborativo permite citar la siguiente definición, adecuada de Johnson, D. y Johnson, R. citada Izuzquiza y Gómez [15]:

“El aprendizaje colaborativo es un conjunto de métodos de instrucción o entrenamiento para uso en grupos pequeños, así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro de grupo es responsable tanto de su aprendizaje como del de los restantes miembros del grupo”.

Por otra parte Leidner y Jarvenpaa [16] señalan que el aprendizaje colaborativo, además de ayudar a desarrollar el pensamiento crítico en los estudiantes, también contribuye a mejorar las relaciones interpersonales, ya que implica que cada miembro aprenda a escuchar, comprender y comunicar las propias ideas u opiniones a los demás, con un enfoque positivo y constructivista. Algunos estudios como el de Scagnoli [17] sustentan que el aprendizaje colaborativo aumenta la satisfacción y motivación del participante y lo prepara como investigador. Investigaciones realizadas en niveles primarios y secundarios de la educación han comprobado que los alumnos aprenden mejor en situaciones no competitivas y de colaboración, que en situaciones adonde se enfatiza la individualidad y la competencia.

2.3.1 El Aprendizaje Colaborativo desde la Parte Cognitiva

Dentro del ámbito educativo actual las propuestas de aprender de forma colaborativa traen consigo algunas ventajas aparte de la educativa, dentro de las cuales una de ellas es el fomento de habilidades sociales, de comunicación, aceptación de los puntos de vista distintas, etc. El propósito de este tipo de aprendizaje en la parte educativa, es ampliar gradualmente la zona cognitiva para que se convierta en zona de desarrollo real, es decir, que esas actividades (incluyendo las actividades mentales), puedan incluirse dentro del rango de las que el individuo puede realizar en forma autónoma y nuevas actividades, de superior nivel de complejidad, sean las se realizan con ayuda [48]. El aprendizaje colaborativo involucra desde el punto de vista cognitivo el uso de modelos y entrenamiento para que el alumno retenga la información en la memoria y la incorpore en las estructuras cognitivas que posee.

2.3.2 Aportes del Aprendizaje Colaborativo

La práctica del aprendizaje colaborativo dentro de la enseñanza de una temática en particular es fundamental para promover la interacción entre los miembros del equipo y así lograr que socialicen conocimientos entre los mismos, además, de valorar los aportes individuales de los miembros del grupo en las tareas, motivar a que los estudiantes sean más responsables y ante todo lograr que los estudiantes desarrollen habilidades tanto personales como de trabajo en equipo, por ejemplo: debatir, cuestionar, tomar decisiones, liderar, coordinar y hacer que el estudiante se autoevalúe sobre su propio desempeño continuo, tanto a nivel personal y de grupo.

Lucero [19], ha estructurado las ventajas del aprendizaje colaborativo de acuerdo a la ejecución de tareas, las relaciones interpersonales, al aprendizaje y al desarrollo personal. A Continuación se describen cada una de esas ventajas.

Con respecto a la ejecución de tareas de tipo grupal, el aprendizaje colaborativo:

- Promueve el logro de objetivos con mayor calidad y variedad por el aporte de todos los integrantes del grupo.
- Valora el conocimiento de los demás miembros del grupo.
- Muestra el compromiso de cada integrante.

Con respecto a las relaciones interpersonales, el aprendizaje colaborativo:

- Estimula la solidaridad y respeto mutuo.
- Se fortalecen las habilidades sociales y de comunicación.

Respecto al aprendizaje:

- Favorece la generación de conocimiento debido a que el estudiante lleva a cabo investigaciones.
- Ayuda a desarrollar un pensamiento crítico.

Respecto al desarrollo personal:

- Fortalece la seguridad en sí mismo.
- Disminuyen el aislamiento.
- Ayuda a perder el miedo a la frustración y las críticas.

2.3.3 Elementos Básicos del Aprendizaje Colaborativo

Para que una actividad implique una verdadera experiencia de aprendizaje colaborativo es necesario que existan ciertos elementos básicos [19]:

- **Interdependencia Positiva:** este es el elemento central, abarca las condiciones organizacionales y de funcionamiento que deben darse al interior del grupo. Los

integrantes del equipo deben estar en función de los demás, es decir necesitarse los unos a los otros y obtener la confianza en razón de lo que entiende cada persona. Considera aspectos de interdependencia en el establecimiento de metas, tareas, recursos, roles y premios.

- **Promover la Interacción:** las formas de interacción y de intercambio verbal entre las personas del grupo, movidas por la interdependencia positiva son las que afectan los resultados de aprendizaje. El contacto permite realizar el seguimiento y el intercambio entre los diferentes miembros del grupo; el alumno aprende de ese compañero con el que interactúa día a día, o el mismo le puede enseñar, cabe apoyarse y apoyar. En la medida en que se posean diferentes medios de interacción, el grupo podrá enriquecerse, aumentar sus esfuerzos y retroalimentarse.
- **Responsabilidad Individual:** cada miembro del grupo debe asumir íntegramente su tarea y, además, tener los espacios para compartirla con el grupo y recibir sus contribuciones.
- **Habilidades Personales y de Grupo:** la vivencia del grupo debe permitir a cada miembro de éste el desarrollo y la potencialización de sus habilidades personales. De igual forma permitir el crecimiento del grupo y la obtención de habilidades grupales como: escucha, participación, liderazgo, coordinación de actividades, seguimiento y evaluación.
- **Procesamiento Grupal:** hace referencia a los procesos de reflexión en torno al trabajo realizado, a fin de determinar, analizar y mejorar el grado de efectividad de los aportes que cada individuo hace para alcanzar los objetivos del grupo.

2.3.4 Técnicas de Aprendizaje Colaborativo

Hay muchas variaciones dentro del modelo de aprendizaje colaborativo. Algunos de los métodos/técnicas/estructuras más populares son [20], para más detalle ver Anexo A:

- **Escucha Enfocada:** esta técnica puede usarse para generar ideas, descripciones o definiciones de conceptos. Los estudiantes formaran equipos pequeños para discutir las ideas o seleccionar aquéllas en los que estén todos de acuerdo
- **Rompecabezas (Jigsaw):** en esta actividad se emplea para reemplazar una exposición con una actividad colaborativa.
- **Anotaciones en Pares:** para esta técnica los estudiantes en pareja, revisan y aprenden del mismo artículo, capítulo o concepto, e intercambian ensayos para su lectura y reflexión.

- **Mesa Redonda:** esta actividad puede ser usada para generar ideas o repuestas a una sola pregunta o grupo de preguntas. El profesor hace la pregunta o preguntas. Cada equipo usa un lapicero y papel. El primer estudiante escribe una respuesta, la dice en voz alta y pasa el papel al siguiente estudiante que repite el procedimiento. Esto continúa hasta que se termina el tiempo.
- **Solución de Problemas Pensando en Voz Alta:** en esta técnica, los estudiantes trabajan en parejas para resolver un problema. Una de las personas actúa como el solucionador de problemas y la otra como la persona que escucha. Las personas que escuchan motivan a sus compañeros a seguir hablando y seguir generando soluciones o pistas si el solucionador no ha generado las ideas suficientes. Los roles se intercambian para el siguiente problema.
- **Entrevista de Tres Pasos:** en este caso las entrevistas de tres pasos son básicamente una actividad de introducción que les permite a los equipos recién formados conocerse. Los docentes pueden asignar los roles a los estudiantes para explorar conceptos a profundidad. En esta actividad, el profesor puede dar preguntas de entrevista o información a cada estudiante.

2.4 INGENIERIA DE COLABORACIÓN

La Ingeniería de Colaboración es un acercamiento al diseño de procesos colaborativos reutilizables. Los procesos colaborativos necesitan ser explícitamente diseñados, estructurados y manejados. Este es el eje central del área denominada “Ingeniería de Colaboración”, en el cual se diseñan procesos repetitivos colaborativos, los cuales se pueden transferir a grupos, usando técnicas y tecnologías de colaboración [49].

2.4.1 Patrones de Colaboración

En las investigaciones realizadas alrededor de la Ingeniería de Colaboración, se han establecido una serie de patrones, relacionados con la forma en la cual un grupo trabaja colaborativamente hacia sus metas. Estos patrones, denominados “patrones de colaboración” se definen en términos del movimiento del grupo desde su estado inicial hasta su estado final [49]. Cada patrón tiene subpatrones que se pueden relacionar con las actividades, en la descripción del proceso genérico [50][51] Ver Anexo B.

2.4.2 Thinklets

Las descripciones relacionadas con la manera como se llevan a cabo actividades, cuando las personas trabajan de manera colaborativa, han sido denominadas como Thinklets; que

son técnicas de facilitación repetibles, transferibles y predecibles para asistir a un grupo en alcanzar su objetivo acordado. Citado en [52] y tomado de [53].

Los Thinklets se detallan completamente y son modificables. Pueden usarse para construir procesos compuestos; son recetas creadas para ser fácilmente utilizadas, ya que son fácilmente aprendidos, recordados y pueden adaptarse a un diseño de proceso [49]. Se presentan a continuación algunos de los propósitos de los Thinklets [50]:

- Soportar el diseño de procesos colaborativos.
- Servir como un lenguaje común entre los usuarios.
- Servir como lenguaje de partida para ejecución de la técnica.
- Servir como instrumento de investigación para comparar diferentes técnicas de facilitación.

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 grupales [54]. Un Thinklet viene documentado de la siguiente manera: [55]

- *Nombre:* el cual deber ser metafórico o representativo relacionado con los patrones que crea el Thinklet.
- *Criterio:* Permite decidir cuándo escoger o no el Thinklet.
- *Información General del Thinklet:* como entradas y salidas. Las entradas son deben ser suministradas al momento que el Thinklet es usado y las salidas son los entregables que se deben generar como resultado de su ejecución.
- *Como usar el Thinklet:* incluye información relacionada con los pasos que conforman el Thinklet.
- *Un Historia exitosa:* que ayude a clarificar las circunstancias bajo las cuales el Thinklet es útil.
- *Una explicación del nombre:* esto hace más fácil recordar el nombre del Thinklet.

En el Anexo C, se describen detalladamente algunos de los Thinklets propuestos por los investigadores en Ingeniería de Colaboración [55].

2.5 METODOLOGIAS AGILES DE DESARROLLO DE SOFTWARE

Para asegurar el éxito durante el desarrollo de software no es suficiente contar con el modelado y las herramientas para el desarrollo, hace falta un elemento importante y es la metodología de desarrollo, la cual provee de una dirección a seguir para la correcta aplicación de las técnicas y las prácticas [21].

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software [30]. De acuerdo a Hurtado et al [31], toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistemas, etc.). Tradicionalmente el desarrollo de software se ha realizado empleando metodologías que se ajustan al proceso de forma estática, aunque son inminentes las características de dinamismo y variabilidad que se pueden dar en un desarrollo [22]. Hoy en día en las organizaciones se tiende hacia el rápido desarrollo de aplicaciones [23], lo que permite generar mayor actividad y rendimiento en los procesos.

De esta manera surgen las metodologías ágiles, para el desarrollo de software en una época aguda de la crisis del software, con el propósito de articular soluciones a esa crisis [25].

Las metodologías ágiles de desarrollo de software son uno de los temas emergentes en ingeniería del software de gran interés tanto para la industria, como de las comunidades de investigación. Esta tal su impacto que actualmente existen tres conferencias⁷ internacionales de alto nivel y específicas sobre el tema. Por ello, hablar de agilidad denota facilidad para el movimiento, actividad y destreza en el movimiento de allí el enfoque práctico de estas hacia el desarrollo de productos software. Algunas de las ventajas del trabajo con metodologías ágiles son [22]:

- Capacidad de respuesta a cambios a lo largo del desarrollo dado que percibe los cambios como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente.
- Entregas continuas y en plazos breves de software funcional, lo que permite al cliente verificar en el lugar el desarrollo del proyecto e ir comprobando si satisface sus necesidades.
- Desarrollo en ciclos de corta duración que favorece que los riesgos y dificultades se repartan a lo largo del desarrollo del producto, principalmente al comienzo del mismo y permite ir aprendiendo de estos riesgos y dificultades, figura 2.

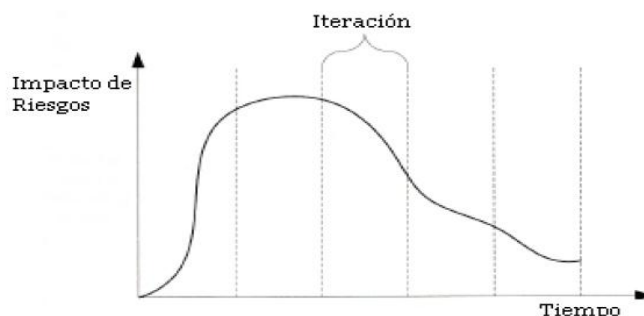


Figura 2. Distribución del riesgo en un desarrollo ágil

⁷ <http://xp2011.org/>, <http://agile2011.agilealliance.org/>, <http://www.agileconference.org/>

- Trabajo conjunto entre el cliente y el equipo de desarrollo con una comunicación directa que pretende evitar malentendidos.
- Importancia de la simplicidad, eliminando el trabajo innecesario que no aporta valor.
- Mejora continua de los procesos y el equipo de desarrollo, entendiendo el éxito como la unión de factores, técnicos, personales y organizacionales.

En febrero de 2001, 17 figuradas destacadas en el campo del desarrollo ágil se unieron para tratar la unificación de sus metodologías en una reunión que se llevó a cabo en Utah-EEUU, donde nace el término “ágil” aplicado al desarrollo de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software de forma rápida y respondiendo a los cambios que puedan surgir a lo largo de un proyecto [26]. Los firmantes del manifiesto ágil fueron: Kent Beck (XP), Mike Beedle, Arie van Bennekum (DSDM), Alistair Cockburn (Crystal), Ward Cunningham (XP), Martin Fowler (XP), James Grenning (XP), Jim Highsmith (ASD), Andrew Hunt (Pragmatic Programming), Ron Jeffries (XP), Jon Kern (FDD), Brian Marick, Robert C. Martin (XP), Steve Mellor, Ken Schwaber (Scrum), Jeff Sutherland (Scrum) y Dave Thomas (Pragmatic Programming) [25]. Tras esta reunión se creó “The Agile Alliance” [27], organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, que se encarga de resumir esta filosofía.

2.5.1 Manifiesto Ágil

El manifiesto ágil para el desarrollo de software se convierte en un cambio respecto al desarrollo, dado que fue creado para contrarrestar los métodos orientados a los procesos y la documentación, e intenta sugerir nuevos enfoques de desarrollo de software donde cobran mayor importancia las personas y los aspectos humanos [28]. El manifiesto ágil es una colección de cuatro valores y doce principios que buscan incrementar el valor:

- A las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es tan importante construir un buen equipo como construir un buen entorno.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- La colaboración continua con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo.
- La habilidad de responder a los cambios que puedan surgir a lo largo proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) esto determina también el éxito o fracaso del mismo.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los principios son:

- I. *La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporten un valor.*
- II. *Dar la bienvenida a los cambios de requisitos. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
- III. *Entregar frecuentemente software que funcione, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.*
- IV. *Los miembros del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*
- V. *Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*
- VI. *El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*
- VII. *El software que funciona es la medida principal de progreso.*
- VIII. *Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*
- IX. *La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*
- X. *La simplicidad es esencial.*
- XI. *Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*
- XII. *En intervalos regulares, el equipo debe reflexionar sobre cómo ser más efectivo y según estas reflexiones ajustar su comportamiento.*

Estos principios marcan el ciclo de vida de un desarrollo ágil, así como las prácticas y procesos a utilizar [29].

2.5.2 Revisión de Metodologías

A continuación se describen algunas de las metodologías ágiles más destacadas enfatizando principalmente en las metodologías XP (Programación Extrema) y Scrum como base de este estudio.

En el Anexo D, se describe de forma más detalla cada una de las metodologías descritas a continuación.

2.5.2.1 Programación Extrema (XP)

La programación extrema (XP⁸) en inglés (Extreme Programming) es una metodología ágil atribuida a Kent Beck, Ron Jeffries y Ward Cunningham [32]. El objetivo principal de XP

⁸ www.extremeprogramming.org

es buscar la satisfacción del cliente tratando de mantener todo el tiempo su confianza en el producto. XP está dirigido a grupos pequeños y medianos de construcción de software donde los requisitos cambian rápidamente o son de alto riesgo [25].

Una de las características de XP es que se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo [21].

▪ **Prácticas de la Programación Extrema (XP)**

XP busca disminuir la curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el proceso evolutivo funcione [33]. Esto se consigue gracias a la aplicación disciplinada de prácticas que ayudan en el desarrollo de software [34]. Las prácticas en XP se describen a continuación:

- ✓ **El juego de la planificación:** permite una comunicación frecuente entre el cliente y los programadores, el equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y el cliente decide sobre el ámbito y tiempo de las entregas y de cada iteración.
- ✓ **Entregas pequeñas:** Producir rápidamente versiones del sistema que sean operativas aunque no cuenten con toda la funcionalidad del sistema.
- ✓ **Metáfora:** historia compartida que describe cómo debería funcionar el sistema.
- ✓ **Diseño Simple:** diseñar la solución tan simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- ✓ **Pruebas:** la producción de código va dirigida por las pruebas unitarias, establecidas por el cliente antes de codificar y son ejecutadas constantemente.
- ✓ **Refactorización:** se realiza constantemente con el fin de remover duplicación del código, mejorar la legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.
- ✓ **Programación en parejas:** la programación debe realizarse con trabajo en parejas dado que se reduce la tasa de errores, se mejora en diseño y se da mayor satisfacción de los programadores.
- ✓ **Propiedad colectiva del código:** en un proyecto XP, cualquier par de programadores puede mejorar cualquier parte del código en cualquier momento.
- ✓ **Integración continua:** cada pieza del código es integrada en el sistema una vez esté lista.
- ✓ **Cuarenta horas por semana:** se debe trabajar máximo cuarenta horas por semana.

- ✓ **Cliente en el sitio de trabajo:** el cliente tiene que estar disponible y presente todo el tiempo en el equipo, este es uno de los principales factores de éxitos en proyectos XP.
- ✓ **Estándares de codificación:** es decisivo para poder planear con éxito la propiedad colectiva del código. Permite que todo el equipo se sienta cómodo con el código escrito.

El mayor beneficio de la aplicación de las prácticas se consigue con una aplicación conjunta y equilibrada puesto que se apoyan unas con otras, el mérito de XP es combinarlas de forma efectiva y combinarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo figura 3.

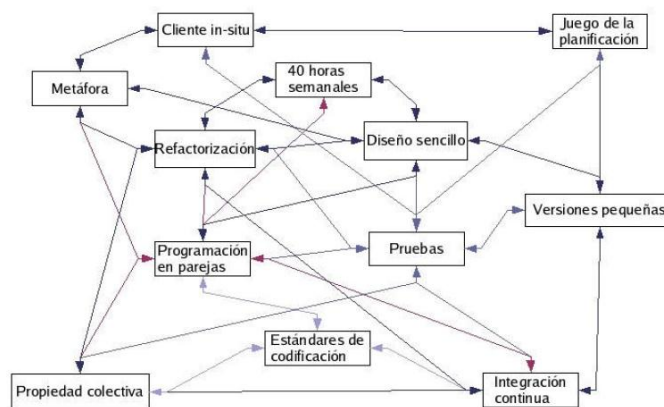


Figura 3. Relación de Prácticas en XP [34]

2.5.2.2 Scrum

Como metodología ágil específicamente aplicada a la ingeniería del software, Scrum fue aplicado por Jeff Sutherland y elaborado más formalmente por Ken Schwaber [35]. Scrum define un marco para la gestión de proyectos, ha sido utilizado con éxito durante los últimos diez años. Especialmente recomendado para proyectos con un rápido cambio de requisitos. Sus principales características son [34]:

- El desarrollo de software se realiza mediante iteraciones, denominadas *Sprint*, con una duración aproximada de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- Se realizan reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Scrum no está concebido como un método independiente, es un complemento de otras metodologías, como XP o el Proceso Unificado. Como método, scrum enfatiza valores y prácticas de gestión y no incluye prácticas para los requisitos, implementación y demás,

de allí su gran insuficiencia, pero al mismo tiempo su capacidad para complementar otras metodologías [25].

▪ **Prácticas de Scrum**

Scrum requiere ciertas prácticas y herramientas de administración en varias fases del scrum para evitar el caos causado por los imprevistos y por la complejidad [25].

A continuación, se describen algunas prácticas:

- ✓ **Pedido del producto:** define todo lo que se necesita en el producto final basado en el conocimiento inicial.
- ✓ **Estimación de esfuerzo:** es un proceso iterativo, en el que el dueño del producto y el equipo scrum son responsables de realizar la estimación del esfuerzo.
- ✓ **Sprint:** es el procedimiento de adaptación al cambio de las variables ambientales (requisitos, tiempo, recursos, conocimiento, tecnología, etc.). El equipo scrum se organiza para producir un nuevo incremento ejecutable del producto en un sprint que dura aproximadamente treinta días.
- ✓ **Reunión de planeación:** la reunión es organizada por el scrum master y consta de dos fases: los clientes, usuarios, gerentes, el propietario del producto y equipo scrum participan en la primera fase de la reunión para decidir sobre los objetivos y funcionalidad del nuevo sprint. La segunda fase de la reunión se realiza entre el scrum master y el equipo scrum, se enfoca en como el incremento del producto es implementado durante el proceso.
- ✓ **Pedido del sprint:** es el punto de partida para cada sprint. Esta es una lista de ítems del pedido del producto seleccionados para ser implementados en el próximo sprint.
- ✓ **Reunión diaria scrum:** las reuniones diarias o scrum diario son organizadas para mantener una revisión constante del progreso del equipo y también sirve como reunión de planeamiento.
- ✓ **Revisión a reunión de sprint:** en el último día de reunión sprint el equipo scrum y el scrum master presentan los resultados del sprint a la gerencia. Clientes, usuarios y propietario del producto participan de la reunión, ellos evalúan el incremento del producto y toman decisiones acerca de la siguiente actividad.

2.5.2.3 Crystal Methodologies⁹

Se trata de un conjunto de metodologías para el desarrollo de software desarrolladas por Alistair Cockburn [36]. El equipo de desarrollo en crystal es un factor clave, por lo que se

⁹ <http://www.crystallmethodologies.org>

deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas, estas políticas dependerán del tamaño del equipo, estableciendo una clasificación por colores, cada color denota la complejidad, cuanto más oscuro sea el color más pesado es el método.

Los métodos se llaman Crystal como facetas de una gema: cada faceta es otra versión del proceso y todas se sitúan en torno a un núcleo idéntico. Los parámetros son Comodidad (C), Dinero Discrecional (D), Dinero Esencial (E) y Vidas (L). Crystal Clear es la metodología más documentada y consiste en valores, técnicas y procesos.

La metodología Crystal no prescribe las prácticas de desarrollo, las herramientas o los productos que pueden usarse, permitiéndole combinarse con otros métodos como Scrum, XP entre otros.

2.5.2.4 Dinamic Systems Development Method (DSDM)¹⁰

Define un marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD¹¹ unificada. Sus principales características son: un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases [28].

Dado el enfoque hacia proyectos de características RAD esta metodología encuadra perfectamente en el movimiento de metodologías ágiles.

2.5.2.5 Adaptative Software Development (ASD)¹²

Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad y se entrega al cliente [37].

¹⁰ <http://www.dsdm.org>

¹¹ **RAD:** Desarrollo Rápido de Aplicaciones, metodología de desarrollo de software que utiliza una planificación mínima a favor de la creación rápida de prototipos.

¹² <http://www.adaptivesd.com>

2.5.2.6 Feature Driven Development (FDD)¹³

Sus impulsores son Jeff De Luca y Peter Coad. FDD define un proceso iterativo que consta de cinco pasos. Las iteraciones son cortas (2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software [28]. FDD no requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso.

2.5.2.7 Evolutionary Project Management (EVO)¹⁴

Evo, fue creado por Tom Gilb, es el método iterativo ágil más antiguo. El modelo de Evo consiste en cinco elementos mayores: metas, valores y costos, soluciones, estimación de impacto, plan evolutivo y funciones.

En proyectos evolutivos, las metas se desarrollan tratando de comprender de quienes vienen (participantes), qué es lo que son (medios y fines) y cómo expresarlos (cuantificables, medibles y verificables). Se procura pasar el menor tiempo posible en tareas de documentación [39].

2.6 AMBIENTE DE DESARROLLO

El ambiente es definido por la real academia de la lengua como las “Condiciones o circunstancias físicas, sociales, económicas, entre otras, de un lugar, de una reunión, de una colectividad o de una época. Según el Diccionario de Psicología¹⁵ define el ambiente como el espacio vital en el que se desarrolla el sujeto. Conjunto de estímulos que condicionan al individuo desde el momento mismo de su concepción.

2.6.3 Desde la perspectiva de la Colaboración

Los ambientes colaborativos son los espacios para el desarrollo de un aprendizaje específico en cuanto al trabajo en comunidad y el fortalecimiento de la identidad [77]. El aprendizaje en ambientes colaborativos busca propiciar espacios en los cuales se dé la discusión entre participantes al momento de explorar conceptos que interesa dilucidar o situaciones problemáticas que se desea resolver; se busca que la combinación de situaciones e interacciones sociales pueda contribuir hacia un aprendizaje personal y grupal efectivo.

¹³ <http://www.featuredrivendevelopment.com>

¹⁴ <http://www.spipartners.nl/data/Evo99.PDF>

¹⁵ http://www.psicoactiva.com/diccio/diccio_a.htm

2.6.4 Desde la perspectiva del aprendizaje

Un ambiente de aprendizaje es un espacio en el que los estudiantes interactúan, bajo condiciones y circunstancias físicas, humanas, sociales y culturales propicias, para generar experiencias de aprendizaje significativo y con sentido. Dichas experiencias son el resultado de actividades y dinámicas propuestas, acompañadas y orientadas por un docente [79]. En los ambientes escolares es evidente el desarrollo humano que los potencia en las tres dimensiones: socio afectiva (emociones, sentimientos, actitudes, valores), cognitiva (nociones, proposiciones, conceptos, argumentos) y físico-creativa (comunicar, crear, innovar).

2.6.2 Desde la perspectiva del Desarrollo de Software

El ambiente de desarrollo corresponde al entorno en el cual el proceso es conducido para construir software abarcando los requisitos, el diseño, la implementación, la prueba, la gestión de la configuración y la gestión del proyecto [67]. Desde la perspectiva de desarrollo ágil de software se requiere entornos de desarrollo que soporten las prácticas de trabajo que requieren la facilitación de los principios del postulado ágil [78], es decir, la comunicación y la interacción cara a cara, la entrega continua de resultados, la colaboración con el cliente y planificación de tareas orientada a la generación de valor [27]. El ambiente en el que se lleva a cabo el desarrollo de software ágil es importante ya que puede contribuir significativamente dado que un ambiente de trabajo que favorezca la comunicación personal puede ayudar a facilitar la interacción entre los individuos, el ambiente físico pueden ayudar a los desarrolladores centrarse en el desarrollo y la ubicación de un equipo de desarrollo de cerca del cliente puede aumentar la colaboración con clientes, hasta llegar a arreglos flexibles para proyecto. Estos patrones proveen un importante punto de partida para relacionar la experiencia y facilitar el desarrollo ágil de software, para ello están divididos en cuatro categorías: el contexto, el lugar, el espacio y el detalle [78].

2.6.4 Desde la perspectiva de la Programación de Software

El estándar IEEE610.12, correspondiente al glosario de terminología en ingeniería del software, define un entorno integrado de programación al conjunto integrado de herramientas de software.

2.7 ESTADO DEL ARTE

A continuación se presentan algunos estudios enfocados en la enseñanza de la programación de software apoyados por técnicas de aprendizaje colaborativo utilizando algunos entornos de programación enfocado a los niños.

2.7.1 Uso de Estrategias de Programación en Jardines Infantiles y Escuelas

El objetivo del trabajo de Morgado et. al. [41] fue encontrar la forma, las herramientas y/o técnicas que permitieran a los niños que se encontraban en un rango de 4 y 5 años de edad utilizar los conceptos de lenguajes de programación para expresar las reglas y jugar con ellas. Esta experiencia fue realizada con 7 niños, 5 de ellos con 4 años de edad y 2 de 5 años, en Vila Real (Portugal), realizada en un jardín infantil entre los meses de mayo y junio del año 2000. La experiencia fue dividida en sesiones, cada sesión de trabajo estaba organizada para un tiempo límite determinado, suponiendo que los niños se aburrirían con sesiones de trabajo más largas, sin embargo, desde la primera sesión esta suposición fue errónea, a los niños les encantó la herramienta que fue utilizada, ToonTalk¹⁶ y temas tratados fueron clave para futuros experimentos con ToonTalk.

2.7.2 Una mirada Cualitativa a la Programación en Pareja y Alice

Por su parte, Howard et. Al. [42] en su trabajo dan a conocer su experiencia con la participación de 89 estudiantes de seis de las distintas secciones de una clase de introducción a la informática para completar un módulo de programación de 2,5 semanas utilizando Alice¹⁷, un lenguaje de programación gráfica interactiva. Los alumnos de dos secciones completaron el módulo de programación de Alice de manera individual y los estudiantes de las otras cuatro secciones utilizaron el paradigma de programación en pares, en el que dos programadores trabajan en el mismo programa al mismo tiempo usando el mismo equipo. Al final del módulo, los 89 estudiantes escribieron un ensayo reflexivo sobre su experiencia con Alice. Estos datos cualitativos indicaron que los estudiantes de programación, entendían conceptos básicos de programación y entendieron la relación entre los algoritmos y las historias de Alice.

2.7.3 Niños Tecnologías Cuentacuentos: Las Diferencias en Elaboración y Recuerdos

El trabajo de Boltman y Druin [40] propone un estudio enfocado en el trabajo colaborativo utilizando la herramienta KidPad¹⁸ dada la capacidad que tiene esta para permitir a los niños crear una estructura de sus propias historias y poder codificar estas en un nivel muy alto. El grupo de niños en este estudio fue dividido en partes iguales entre las dos escuelas participantes ubicadas en Suecia e Inglaterra, respectivamente. Dentro de las escuelas, los participantes fueron asignados aleatoriamente, un total de 72 niños en edades comprendidas entre los 6 y 7 años, 36 de ellos en cada localidad. La muestra estuvo conformada por los niños con exposición previa, poca o ninguna a KidPad, la

¹⁶ <http://www.toontalk.com/English/computer.htm>

¹⁷ <http://www.alice.org>

¹⁸ <http://cs.umd.edu/hcil/kiddesign/kidpad.shtml>

tecnología utilizada en este estudio que no era totalmente conocida por los niños. El resultado arrojado de esta experiencia evidencia un impacto educativo de la tecnología en los niños permitiéndoles desarrollar sus habilidades narrativas. Además, ilustra como KidPad en efecto, involucra fácilmente los niños con la tecnología, lo que permite deducir su aplicación en diferentes áreas.

2.7.4 Los Grados de Comprensión: Comprensión de los Niños de un Entorno de Programación Visual

Rader et. al. [43] también presenta en su trabajo a los niños como una nueva generación innovadora y con características propias para trabajar en entornos de desarrollo altamente visuales. El estudio reporta la experiencia de una profesora de ciencias que trabaja con un grupo de estudiantes de 2 y 3 años de edad con un proyecto al que denominaron Teatro de Ciencias y sus alumnos de 4 y 5, trabajando con KidSim, posteriormente denominado Cocoa [44]. Para ello se consideraron los requisitos básicos de instrucción para que los niños aprendieran a programar en este entorno visual. El método utilizado para evaluar la comprensión de los estudiantes sobre el funcionamiento del programa fue desarrollar una serie de 10 tareas que utilizan las características básicas del entorno de programación. El entorno de KidSim les permitió llegar a la conclusión de que los niños fácilmente dominan el dibujo y la animación de personajes en mundos imaginarios. Los niños más pequeños proponen retos importantes en el dominio del medio ambiente, lo suficiente buenos para crear programas útiles en su estudio de las ciencias, aunque se les dificulta un poco el dominio de la herramienta. A los niños más grandes les fue mucho mejor, pero a menudo no alcanzaron a comprender en su totalidad los aspectos importantes de KidSim. El principal interés de este estudio fue la programación como un vehículo para apoyar el desarrollo conceptual de otras áreas.

2.7.5 La enseñanza de Programación de Computadoras en las Escuelas de Primaria: Un Estudio Piloto

En el estudio de Lin et. al. [46], 81 estudiantes entre 10 y 12 años de edad, aprendieron a usar Stagecast Creator¹⁹, HANDS²⁰ y Visual Basic²¹ para crear juegos simples y animaciones. El estudio se llevó a cabo en tres escuelas primarias, en cada escuela el experimento duro dos semanas, tres horas al día y cinco días a la semana. En primer lugar presentaron a los estudiantes la herramienta Stagecast Creator, luego HANDS y Visual Basic. Con cada paquete de programación los estudiantes aprendieron a crear juegos simples y animaciones. Uno de los autores de este trabajo fue el instructor y tres restantes eran profesores asistentes y observadores. Los estudiantes trabajaron individualmente en sus propios equipos la mayor parte del tiempo, pero de vez en cuando

¹⁹ <http://www.stagecast.com>

²⁰ **HANDS:** Human-centered Advances for the Novice Development of Software

²¹ <http://www.microsoft.com>

se unieron para hacer proyectos más grandes. Los resultados de este estudio indicaron que los niños puedan aprender y disfrutar de la programación de computadores.

2.7.6 Fuentes Divertidas: Presentación de 4to y 5to Grado de Programación Usando Squeak

Sánchez-Ruiz y Jamba en su estudio [47] discuten una serie de actividades diseñadas para mejorar las habilidades informáticas de los estudiantes de primaria que les permita inclinarse hacia a carreras de Tecnología de la Información. El objetivo de las sesiones incluye el aumento de la comprensión de los estudiantes formales de los mecanismos internos de una computadora y un aumento de su conjunto de habilidades abstractas mediante la adición de programación de computadoras a la misma.

En el estudio el grupo de estudiantes eran niños de cuarto y quinto de primaria quienes participaron en estas sesiones cuya duración era de una hora una vez por semana. En general, los estudiantes disfrutaron de los proyectos informáticos que no formaban parte de su plan de estudios formal. Sin embargo, el aprendizaje se realiza de manera individual y sin una guía metodológica. Al finalizar el primer año de lecciones, aproximadamente el 91% de los estudiantes fueron capaces de seguir la ejecución de un programa informático simple imperativo. Después del segundo año de lecciones, aproximadamente el 91% de los estudiantes fueron capaces de escribir correctamente los programas de software que utilizan las habilidades matemáticas asociadas a los puntos en un sistema de coordenadas de dos dimensiones en el entorno de desarrollo Squeak²².

2.7.7 Experiencias con Scratch en el Aula

Crook [45] documenta un ejemplo del uso de Scratch²³ en el aula como parte del plan de estudios de las TIC para los niños de 7 y 8 años en una escuela primaria 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. Inicialmente la escuela enseñó Scratch como parte del programa normal de la TIC durante una hora por semana en un laboratorio de computación. La experiencia se dio con 23 niños entre los 7 y 8 años de edad, donde todos ellos disfrutaron las sesiones. Los niños compartían el computador, con diferentes proyectos, inicialmente en parejas pero se quejaban que otros niños se levantaban a copiar su trabajo hasta que el profesor les explicó la importancia de compartir e intercambiar conocimientos y técnicas de programación como parte del proceso. Todos los proyectos realizados fueron almacenados en el mismo disco duro y así era fácil para el profesor mostrar las soluciones particulares sobre la pizarra interactiva y animar a los niños a compartir las

²² <http://squeak.org/>

²³ http://info.scratch.mit.edu/es/About_Scratch

soluciones del proyecto. Los investigadores concluyen que Scratch ofrece una oportunidad para enseñar a los niños desde edades tempranas, como programar un computador y crear potentes aplicaciones multimedia y juegos, además de participar en los procesos de colaborar en la planificación, diseño, creación y fijación de una aplicación informática, mejorando así las habilidades de comunicación entre los niños.

2.7.8 Scratch en Colombia

En Colombia, Ceballos y Nieto [58], docentes de informática del Instituto de Nuestra Señora de la Asunción en Cali, trabajan desde hace varios años con el entorno de programación de computadores Scratch con el propósito de mejorar significativamente las habilidades de pensamiento de orden superior de sus estudiantes. Los docentes inicialmente se capacitaron en Scratch y posteriormente, empezaron a llevar Scratch al aula tratando de dosificar las actividades en los grados 3° a 5°. Para el trabajo se plantearon cuatro objetivos claros en el área de informática para estos grados, el primero lograr el desarrollo del pensamiento algorítmico de los estudiantes, cuando un estudiante recibe un problema de programación propuesto por los docentes, él debe escribir y encontrar la manera de solucionarlo y de llevar la secuencia lógica de los pasos a realizar en Scratch para convertir esa solución en un programa de computador, el segundo objetivo incentivar en los estudiantes el pensamiento crítico con el fin de que sea capaz de responder cualquier pregunta y argumentar la forma como realizó su tarea, otro de los objetivos planteados es fomentar el desarrollo de competencias de colaboración y comunicación, así los estudiantes tienen el espacio propicio para explicar a sus compañeros como resolvieron el problema y el último objetivo propuesto es desarrollar la habilidad para solucionar problemas de forma creativa, permitirle a los niños encontrar permanentemente nuevas formas de hacer las cosas y que vayan compartiendo lo que van descubriendo. La estrategia propuesta al momento de poner en marcha un proyecto en Scratch es usar una serie de plantillas (análisis y diseño) de forma sencilla que les facilite su trabajo.

2.7.9 Impacto de la Programación con Scratch en estudiantes con su propio proceso de aprendizaje

Ferrer et. al [62] en su estudio analizan un grupo de estudiantes con su propio proceso de aprendizaje quienes introducen dentro del área matemática una actividad relacionada con la programación básicamente con Scratch. Se trabajó con dos grupos de alumnos: un grupo está en una etapa más temprana del uso de Scratch, mientras que el otro está más avanzada. La metodología utilizada es una explicación de enfoque mixto método de investigación donde los datos cuantitativos y cualitativos se complementan entre sí. Se utilizó un cuestionario, (aprendizaje con escala Scratch) adaptado para poner a prueba los estudiantes de autoaprendizaje de gestión del aprendizaje y la intencionalidad, y los grupos de enfoque para recuperar los datos cualitativos. Hay una meta secundaria

académica para esta investigación: definir y comparar correctamente los conceptos de aprendizaje auto-regulado y auto-dirigida añadiendo puntos de vista sobre cuestiones de la literatura actual. El estudio se realizó con estudiantes de 12 a 13 años de edad quienes fueron divididos en dos grupos (19 niños principiantes y 22 niños avanzados).

La tabla 1 presenta una clasificación de los trabajos relacionados, identificando las características específicas para cada uno de los aspectos tratados en este trabajo.

Propuesta	Prácticas Cognitivas	Prácticas Ágiles	Prácticas Colaborativas	Metodología de Desarrollo Explicita	Reporte de Experiencias	Entorno de Programación Utilizado
Morgado et. al. [41]	Si <ul style="list-style-type: none"> Seguir Órdenes Usan Símbolos y guías. 	Ninguna Expuesta	Si <ul style="list-style-type: none"> Aceptan Reglas. 	Ninguna Expuesta	Ninguno Expuesto	ToonTalk
Howard et. al. [42]	Si <ul style="list-style-type: none"> Seguir Órdenes Entienden Conceptos 	Si <ul style="list-style-type: none"> Programación en Pares 	Si <ul style="list-style-type: none"> Aceptan Reglas 	Ninguna Expuesta	Ninguno Expuesto	Alice
Boltman y Druin [40]	Si <ul style="list-style-type: none"> Seguir Órdenes 	Ninguna Expuesta	Si <ul style="list-style-type: none"> Aceptan Reglas Trabajo en Equipo 	Ninguna Expuesta	Ninguno Expuesto	KidPad
Rader et. al. [43]	Si <ul style="list-style-type: none"> Seguir Órdenes 	Ninguna Expuesta	Ninguna Expuesta	Ninguna Expuesta	Ninguno Expuesto	KidSim
Lin et, Al. [46]	Si <ul style="list-style-type: none"> Seguir Órdenes 	Ninguna Expuesta	Si <ul style="list-style-type: none"> Aceptan Reglas 	Ninguna Expuesta	Ninguno Expuesto	StageCast Creator, HANDS, Visual Basic
Sánchez-Ruiz y Jamba [47]	Si <ul style="list-style-type: none"> Seguir Órdenes Entienden Conceptos 	Ninguna Expuesta	Si <ul style="list-style-type: none"> Aceptan Reglas 	Ninguna Expuesta	Ninguno Expuesto	Squeak
Crook [45]	Si <ul style="list-style-type: none"> Seguir Órdenes 	Si <ul style="list-style-type: none"> Programación en Pares Integración Continua 	Si <ul style="list-style-type: none"> Aceptan Reglas Participación continua 	Ninguna Expuesta	Ninguno Expuesto	Scratch
Ceballos y Nieto [60]	Si <ul style="list-style-type: none"> Seguir Órdenes Entienden Conceptos Explican los pasos a seguir 	Si <ul style="list-style-type: none"> Comunicación Valuntad 	Si <ul style="list-style-type: none"> Aceptan Reglas Participación Continua 	Ninguna Expuesta	Si <ul style="list-style-type: none"> Replicado a través de la Fundación Gabriel Piedrahita Uribe - FGPU²⁴ 	Scratch
Ferrer et. al [62]	Si <ul style="list-style-type: none"> Entienden Conceptos 	Ninguna Expuesta	Ninguna Expuesta	Ninguna Expuesta	Ninguno Expuesto	Scratch
Child Programming ²⁵	Si	Si	Si	Si	Si Reporte de	Scratch

²⁴ Fundación Gabriel Piedrahita Uribe: <http://www.eduteka.org/fgpu>

²⁵ Child Programming Web: <http://www.unicauca.edu.co/childprogramming>

	<ul style="list-style-type: none"> • Seguir Órdenes • Expresar dudas y/o comentarios • Entender Conceptos 	<ul style="list-style-type: none"> • Trabajo en Pares • Utilizar el Espacio Informativo • Diseñar de forma Simple e Incremental • Entregable del Producto 	<ul style="list-style-type: none"> • Aceptar y Respetar Reglas • Trabajo en Equipo • Voluntad para Trabajar en Equipo 	<ul style="list-style-type: none"> • Basado en la Metodología de Desarrollo Ágil Scrum aplicada a un entorno con niños. • Metodología ChildProgramming 	Caso de Estudio. [59][60]	
--	--	---	--	--	---------------------------	--

Tabla 1. Clasificación de Trabajos Relacionados con Child Programming

Teniendo en cuenta la tabla 1, ChildProgramming se diferencia de los trabajos allí descritos porque aporta un conjunto de prácticas ágiles, colaborativas y cognitivas aplicadas a un entorno de escolares de 8 a 10 años, que son organizados en equipos para realizar 3 actividades de tipo colaborativo que incluyen el acercamiento de estos al desarrollo de software, donde emplearan como herramienta de programación Scratch. Estas actividades serán reportadas como experiencias prácticas para ser aplicadas en otros entornos.

2.8 SÍNTESIS Y DISCUSIÓN

En este capítulo hemos presentado un estado del arte sobre el dominio de desarrollo de software enfocado en la programación para niños a través de entornos propios para ellos. Es importante enseñar programación a los niños y de gran valor el esfuerzo que vienen haciendo otros investigadores desde distintas perspectivas.

Las propuestas presentadas tienen enfoques hacia la programación para niños, pero la mayoría de ellas no sintetiza en una metodología apropiada o adaptada al entorno escolar para permitir que los niños puedan apropiarla y desarrollarla de manera natural. Es interesante ver como estos trabajos presentan algunas proyecciones de futuros ingenieros de software y es más interesante aún que en algunos casos los niños sean quienes toman la iniciativa por aprender.

De allí se parte con la idea que los niños empiecen a aprender a trabajar de forma grupal, no de manera individual como es acostumbrado, quizá darles algunas pautas para mejorar la colaboración entre ellos y así obtener mejores resultados en sus desarrollos.

3. EL MODELO CHILD PROGRAMMING

El objetivo general del presente trabajo es obtener y formalizar un modelo de proceso para soportar el desarrollo del software orientado a los niños a partir de un conjunto de experiencias prácticas fundamentadas en estrategias colaborativas y prácticas ágiles, con este modelo de proceso, se presentarán los principales elementos que lo conforman y su descripción detallada.

El modelo de proceso²⁶ Child Programming es el principal producto de esta investigación, resultado de un proceso de abstracción de conceptos, prácticas y actividades colaborativas, evidenciadas en niños con edades comprendidas entre los 8 y 10 años. El modelo de proceso Child Programming refleja como los niños involucrados en este estudio, a través del trabajo colaborativo, prácticas ágiles y elementos cognitivos, logran obtener un conocimiento propio y particular en el área del desarrollo de software.

3.1 CONTEXTO DEL MODELO PROPUESTO CHILD PROGRAMMING

Se denomina ChildProgramming [57], por la combinación de palabras Child (Niños) y Programming (Programación), “Programación para Niños” y nace de un trabajo preliminar, base inicial de esta investigación, el cual incluye una estrategia integrada para la enseñanza del desarrollo de software en niños que incluye una base conceptual y una metodología de desarrollo de software orientada a niños junto con un entorno de desarrollo integrado para la construcción de software y un programa de formación basado en la lúdica, la colaboración y la agilidad. En la figura 4 se representa la arquitectura de esta propuesta.

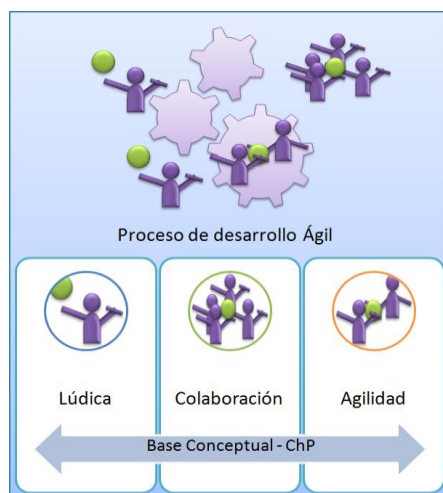


Figura 4. Arquitectura Conceptual de ChildProgramming [57]

²⁶ **Modelo de Proceso:** se define como un conjunto estructurado de elementos que describen las características de procesos efectivos y de calidad. Los modelos de un proceso son usados para documentar y dar soporte a los procedimientos de una organización en una forma consistente.

Partiendo de esta base arquitectónica este trabajo, aborda la definición de un proceso de desarrollo basado en la agilidad como eje central y define algunos elementos fundamentales del marco conceptual. Este proceso guía a los niños en un esfuerzo colaborativo de desarrollo de software, que mantenga visible las rutas de actividades a seguir y que sin forzarlo a seguirlo rigurosamente, motivándolos a seguir ciertas prácticas que potencien su trabajo como equipo. Por ello, Child Programming define en forma concreta pero general, procurando garantizar colaboración, agilidad y libertad en el marco de un proyecto de software enfocado a los niños.

3.1.1 Análisis de Modelos Estudiados

En la tabla 2 se presenta un resumen comparativo de los modelos estudiados con el fin de determinar el que se ajuste a las características de esta investigación y que nos permita hacer una posterior selección para soportar el modelo ChildProgramming.

Modelo Pedagógico	Metodología	Propósito	Diseño Curricular	El Alumno y el Profesor
Modelo Constructivista (Jean Piaget)	El método principal comprende actividades tendientes a la construcción del conocimiento, se parte de los pre conceptos de los estudiantes, de sus intereses y de actividades como la indagación, la experimentación, y de experiencias que van desde lo concreto para llegar a lo abstracto	La autoconstrucción del conocimiento y el acceso a niveles intelectuales superiores. Siendo la finalidad de la escuela promover la construcción del conocimiento en las dimensiones: vida cotidiana, escolar y social; el conocimiento, lo cognitivo, axiológico y procedimental. formando personas autónomas, críticas y capaces de transformar la realidad	Los contenidos se adecuan al nivel de desarrollo de los estudiantes y a sus capacidades psicológicas según el nivel desarrollo, los contenidos se van organizando en el aula.	El alumno es constructor de su propio conocimiento, determina su ritmo de aprendizaje. El Profesor es un orientador, guía, facilitador, acompañante del proceso de construcción de conocimiento.
Modelo Sociocultural (Lev Vygotsky)	El aprendizaje se da primero la asimilación de conocimientos generales y abstractos para luego familiarizarte con los conocimientos particulares y concretos	El propósito es la formación de las estructuras del pensamiento como resultado de un proceso de construcción social en el marco de un contexto cultural e histórico concreto.	Se analiza la realidad social y en forma transversal e integral se selecciona, apropia y evalúa los conocimientos.	El alumno reconstruye los conocimientos ya elaborados por la ciencia y la cultura y en dicho proceso el lenguaje hace las veces de mediador. El maestro contribuye a organizar las áreas para que los estudiantes promuevan su desarrollo intelectual.
Modelos de Aprendizaje Significativo (David Ausubel)	El método, responde a la premisa "averigua lo que el alumno sabe y enseña consecuentemente", se realiza aprendizaje cooperativo, aprendizaje receptivo y por descubrimiento.	El aprendizaje es significativo cuando los nuevos conocimientos se vinculan de manera clara y estable con los conocimientos previos que poseía el individuo, así que, el qué, para qué y cuándo del proceso depende de lo que el alumno ya sabe.	Los contenidos y secuencias curriculares dependen de las experiencias y conocimientos previos, incluyendo una selección rigurosa de los contenidos en los grados y asignaturas, escogiendo lo esencial de cada ciencia y lo significativo.	El alumno debe manifestar una actitud positiva hacia el aprendizaje, debe estar motivado y ser disciplinado frente al estudio. El maestro, cumple un papel de diagnosticador e investigador, debe ser hábil en la determinación de experiencias y conocimientos previos que manejan los estudiantes

<p>Modelo Activista (María Montessori)</p>	<p>El proceso de aprendizaje enseñanza la acción y la experiencia son la condición que garantiza el aprendizaje mediante el uso de métodos activo inductivo, deductivo, lúdico, científico, de construcción del conocimiento; basados en las experiencias y capacidad de los alumnos. Presupone la formación de conceptos por el contacto directo con los objetos, se aprende haciendo, se educa para el trabajo, se realizan proyectos de aula y los contenidos giran alrededor de centro de intereses.</p>	<p>Permitirle al niño actuar y pensar libre, espontáneamente y con autonomía.</p>	<p>Se organiza partiendo de lo simple y concreto hacia lo complejo y abstracto. Tanto la naturaleza como la vida hacen parte del Contenido curricular orientándolas según el predominio de intereses y necesidades del estudiante.</p>	<p>El estudiante cumple un rol activo, creativo, recursivo, autónomo, de interés por el aprendizaje, elaborando su conocimiento al interactuar con los objetos, es líder y responsable. El maestro cumple un papel de orientador, motivador, guía, acompañante, dinamizador y promotor de procesos como la investigación.</p>
<p>Modelo Pedagogía Conceptual (Miguel De Zubiria)</p>	<p>El método de enseñanza, se caracteriza por el modelo hexágono, el cual está constituido por seis componentes interrelacionados en el quehacer pedagógico: propósitos, contenidos, secuenciación, métodos, recursos, y evaluación.</p>	<p>Formar individuos plenos, acorde a la triple naturaleza humana activa, cognitiva y expresiva. La naturaleza activa se refiere al ser apasionado alegre u amoroso; la cognitiva, se refiere al ser brillante u expresivamente talentoso y la expresiva a través de cualquier lenguaje.</p>	<p>El diseño del currículo, se realiza alrededor del modelo del hexágono el cual contiene seis componentes que deben ser desarrollados teniendo en cuenta el carácter tridimensional de la naturaleza humana.</p>	<p>El alumno se desempeña como participe en la construcción de su propio pensamiento y reconstruye en forma activa los conceptos de las ciencias incorporándolos a sus estructuras de pensamiento. El maestro por su parte, orienta los procesos mentales, guía, acompaña, organiza estrategias, es un mediador.</p>

Tabla 2. Comparativo de Modelos Pedagógicos Estudiados

3.1.2 Modelo Seleccionado

Teniendo en cuenta la revisión conceptual de los modelos estudiados y los aportes que cada uno de estos hace para la construcción de conocimiento y la motivación para el aprendizaje de los niños, a continuación en la tabla 3 presentamos los aspectos relevantes en cuanto a conceptos y prácticas relevantes en cada modelo teniendo en cuenta su propósito.

Modelo Pedagógico	Conceptos	Prácticas
<p>Modelo Constructivista</p>	<ul style="list-style-type: none"> - Cumplimiento - Comunicación - Reglas - Condiciones - Entendimiento - Concepto - Cooperación 	<ul style="list-style-type: none"> -Aceptar Condiciones -Cumplir las Reglas del Juego -Desarrollar la actividad en equipo -Comprometerse para trabajar en equipo. -Entender el Tema de la Actividad
<p>Modelo Sociocultural</p>	<ul style="list-style-type: none"> - Cumplimiento - Reglas - Condiciones - Entendimiento - Comentario - Concepto - Interacción -Cooperación 	<ul style="list-style-type: none"> -Aceptar Condiciones -Cumplir las Reglas del Juego -Desarrollar la actividad en equipo -Comprometerse para trabajar en equipo -Entender el Tema de la Actividad
<p>Modelos de Aprendizaje</p>	<ul style="list-style-type: none"> -Inquietudes 	<ul style="list-style-type: none"> -Preguntar lo que no se entiende

Significativo	-Motivación	
Modelo Activista	-Comunicación -Instrucciones -Entendimiento -Tema -Concepto -Tareas	-Cumplir las Reglas del Juego -Entender el Tema de la Actividad
Modelo Pedagogía Conceptual	-Entendimiento -Concepto	-Preguntar lo que no se entiende -Entender el Tema de la Actividad

Tabla 3. Relación de Conceptos y Prácticas relevantes en cada Modelo

Realizando el análisis de cada modelo podemos determinar:

- El Modelo Constructivista propuesto por Piaget aporta a ChildProgramming desde sus bases conceptuales la construcción de conocimiento en diversas dimensiones permitiendo a los niños formarse como personas críticas, autónomas, capaces de transformar realidades. Piaget propone una clasificación del conocimiento de acuerdo a las edades por etapas, teniendo en cuenta los cambios, características y capacidades del pensamiento en cada una de estas, esta clasificación es de nuestro interés específicamente el “estadio de las operaciones concretas” donde se ubican los niños en edades entre los 7 y los 11 años, rango de edad escogido por sus características para esta investigación. Además, dentro de sus postulados se puede relacionar con conceptos y prácticas desde la cognición y la colaboración propias de ChildProgramming.
- ChildProgramming toma del Modelo Socio-Cultural de Vygotsky la forma de aprendizaje de los niños partiendo de la asimilación de conocimientos abstractos para luego familiarizarlos con conocimientos particulares de sus experiencias. Vygotsky aporta desde su teoría conceptos y prácticas como base fundamental para el trabajo en equipo y la colaboración entre integrantes de los mismos.
- ChildProgramming no tiene en cuenta el Modelo de Aprendizaje Significativo de Ausubel, dado que esta teoría parte de que los niños adquieren un nuevo conocimiento y lo vinculan de forma clara a sus conocimientos previos, partiendo de que este depende de lo que los niños ya saben.
- El Modelo Activista de María Montessori propone una enseñanza a partir de la acción y las experiencias como una condición que garantiza el aprendizaje pero se enfoca en uno de sus principios que es la individualidad y para ChildProgramming concebido como un modelo para el aprendizaje colaborativo, no es apropiada esta teoría.
- El Modelo de Pedagogía Conceptual de De Zubiria, a pesar de contar con características importantes del quehacer pedagógico como son los propósitos, los contenidos, la secuenciación, los métodos, recursos y evaluación como apoyo fundamental en cualquier proceso de enseñanza y aprendizaje, no clasifica los

individuos acorde a un proceso de construcción específico de allí que ChildProgramming no lo toma como base para esta investigación.

De aquí que tomamos como base para esta investigación dadas las características de sus aportes el modelo constructivista de Piaget y el modelo sociocultural de Vygotsky.

3.1.3 Diagnóstico Inicial y Caracterización

Basados en los Modelos antes descritos como parte de esta investigación se realizó un diagnóstico inicial para identificar en la población de estudio aspectos como:

- Capacidad Lógico - matemática
- Lenguaje
- Trabajo en equipo
- Afinidades con áreas

Para realizar este diagnóstico se diseñó el “Instrumento para identificar características individuales de los niños”, Ver Anexo H y así identificar aspectos relacionados con su edad, género y gustos, además de obtener información sobre estos aspectos. En el Anexo L se pueden observar los resultados obtenidos después de la aplicación de este instrumento.

A continuación se listan algunas de las características más evidentes en los niños teniendo en cuenta el rango de edad de los 8 a 10 años [76].

Desarrollo Físico

- Los niños realizan actividades deportivas diversas de forma espontánea.
- Escriben bien letras y números
- Organizan sus textos escolares
- Distinguen los alimentos

Desarrollo Cognitivo (Desarrollo del Pensamiento)

- Entienden conceptos de fechas
- Realizan las 4 operaciones matemáticas básicas
- Comprenden fracciones
- Buscan explicaciones lógicas a lo que los rodea
- Logran memorizar datos.
- Siguen órdenes.
- Usan símbolos, ubican lugares en mapas
- Entienden lo que otros piensan

Desarrollo del Lenguaje

- Hablan sobre variedad de temas
- Saben leer y escribir
- Leen instrucciones y siguen pasos
- Entienden gráficos
- Opinan y expresan dudas y comentarios

Desarrollo Socio-emocional

- Comprenden los sentimientos de otros
- Se identifican con creencias
- Comparten con niños y niñas independiente del genero
- Inventan historias y las recrean
- Les gusta los grupos sociales
- Cuentan con un grupo pequeño de amigos
- Participan en actividades

Se corrobora de esta forma para esta investigación el estadio de las operaciones concretas propuesto por Piaget con los datos arrojados por el diagnostico lo cual faculta nuestro trabajo y la presente investigación alrededor de la programación de software en la que se requiere este tipo de habilidades, sin embargo, la construcción del modelo aprovecho algunos aspectos de todas estas las fuentes.

3.2 ARQUITECTURA CONCEPTUAL DEL MODELO CHILD PROGRAMMING

La arquitectura propuesta para el modelo de procesos ChildProgramming está basada en la definida previamente en Hurtado et, al [57]. Si bien el producto principal de ésta tesis es el proceso, fue necesario desarrollar un marco conceptual para soportarlo. Así la arquitectura original ha sido adaptada a ésta propuesta, a partir de la experiencia en el trabajo de campo y sus componentes han sido formalizadas con herramientas de soporte dispuestas para este proyecto. A continuación se describen los principales componentes de la arquitectura propuesta.

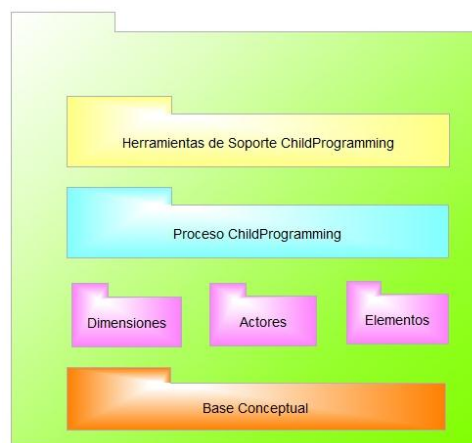


Figura 5. Arquitectura Child Programming Propuesta

3.2.1 Marco Conceptual del Modelo Child Programming

El marco conceptual del modelo Child Programming define los elementos de referencia para el desarrollo de ésta investigación, así como la organización de la propuesta. Este marco incluye los siguientes elementos:

- **Modelo Conceptual Child Programming**

La figura 6 se presenta el modelo conceptual de esta propuesta, el cual fue extraído de las definiciones iniciales de la propuesta y evolucionado a través de su aplicación empírica. El concepto central es el Proceso Colaborativo (Collaborative Process) el cual describe las instancias del proceso en las que el modelo ChildProgramming aplica y los elementos relacionados.

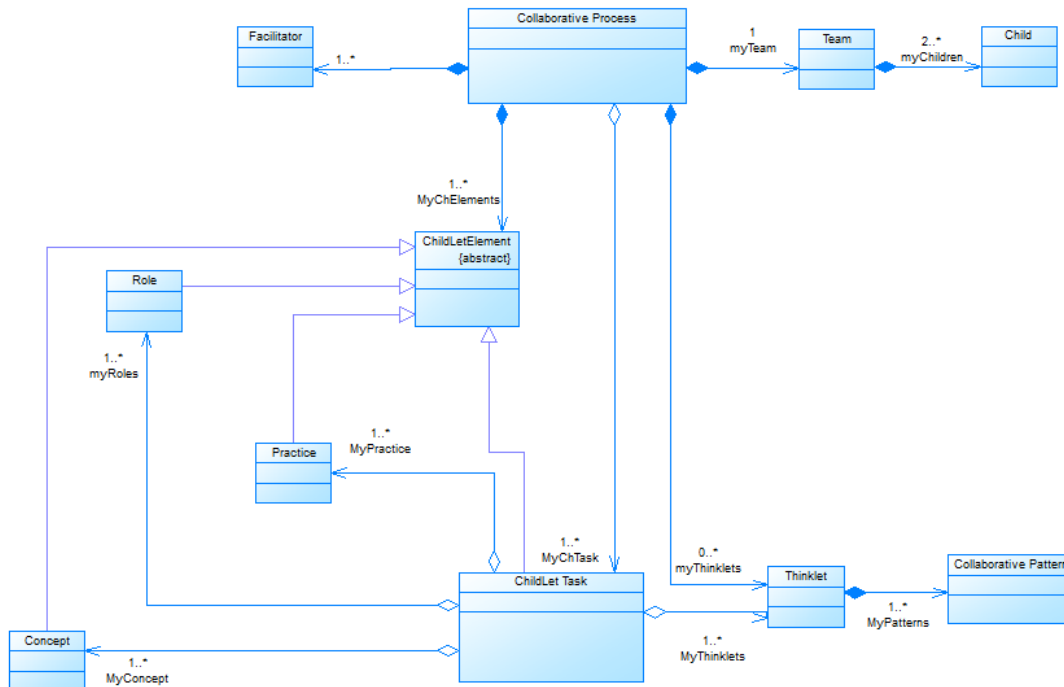


Figura 6. Modelo Conceptual Child Programming

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 de 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.

A continuación se describe cada uno de los conceptos que hacen parte del modelo abstracto de *ChildProgramming*.

- ✓ *Niño (Child)*: concepto que representa la información de un niño(a) que es integrante de un equipo de trabajo, identificado por el atributo *Cod_Child*. Este concepto permite la conformación de equipos (*Team*) y es la encargada de abstraer toda la información particular del niño como su nombre, grado y edad, con los atributos *Name_Child*, *Grade_Child*, *Age_Child*. *Child* representa al niño (alumno), actor principal en el proceso *Child Programming*.

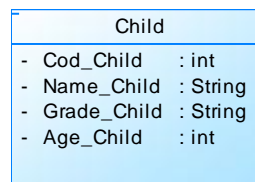


Figura 7. Definición Conceptual de la Clase Child

- ✓ *Equipo (Team)*: clase que contiene la información de un equipo. Un equipo es identificado por el atributo *Id_Team*. Un Equipo es el que ejecuta un Proceso Colaborativo (*Collaborative Process*) y abstrae toda la información del equipo a través los atributos: *Name_Team* y *Description_Team*. Esta clase tiene una relación de composición fuerte con la clase *Collaborative Process*. *Team* representa el equipo al que pertenecen los niños en el proceso *Child Programming*.

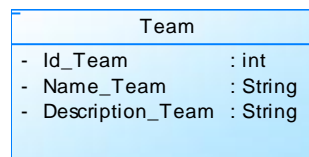


Figura 8. Definición Conceptual de la Clase Team

- ✓ *Proceso Colaborativo (Collaborative Process)*: clase que abstrae toda la información referente al proceso colaborativo que sigue *Child Programming*, un proceso colaborativo se identifica con el atributo *Id_CollaborativeProcess* y abstrae toda la información del proceso a través de los atributos *Name_Process* y *Goal_Process*. *Collaborative Process* representa el proceso colaborativo en el que participan los equipos dentro de *Child Programming*.

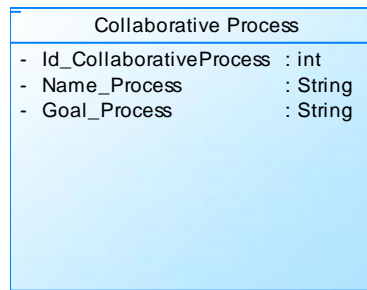


Figura 9. Definición Conceptual de la Clase Collaborative Process

- ✓ *Facilitador (Facilitator)*: clase que contiene la información de un facilitador. Un facilitador es la persona encargada de definir y conducir las actividades prácticas a realizar. Es identificado por el atributo *Id_Facilitator*. El facilitador y el equipo dan inicio al Proceso Colaborativo (*Collaborative Process*).

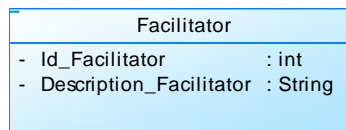


Figura 10. Definición Conceptual de la Clase Facilitator

- ✓ *Elemento (ChildLetElement)*: esta es una clase abstracta donde se describen los atributos y métodos comunes para las clases heredadas (*Concept*, *Practice*, *ChildLet Task* y *Role*). Esta clase abstrae la información referente a los elementos que hacen parte del ChildLet²⁷ a través del identificador único *Id_ChElement* y de los atributos *Name_ChElement*, *Descripción_ChElement*. Un elemento es considerado una unidad fundamental dentro del proceso, la unión de varias unidades determina el desarrollo del Proceso Colaborativo.

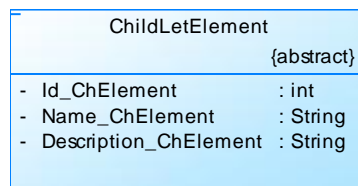


Figura 11. Definición Conceptual de la Clase Abstrata ChildLetElement

- ✓ *Tarea (ChildLet Task)*: esta es una clase heredada de *ChildLetElement*. Describe atributos propios de la clase como: *Inputs_Task* y *Outputs_Task*, un *ChildLet Task* se identifica a través del atributo *Id_ChTask*. Esta clase abstrae toda la

²⁷ **ChildLet**: representa la descripción de cómo llevar a cabo las tareas cuando los niños trabajan de forma colaborativa y ágil.

información de una tarea colaborativa propia del proceso. Una tarea específica una funcionalidad o acción que se quiera realizar, para childprogramming las tareas definen las acciones colaborativas que los equipos deben realizar para cumplir una misión. Es una generalización de la clase *ChildLetElement*, una versión más específica de esta clase, además, tiene una asociación de agregación con la clase *Collaborative Process*.

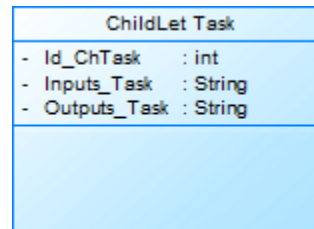


Figura 12. Definición Conceptual de la Clase ChildLet Task

- ✓ *Práctica (Practice)*: clase heredada de *ChildLetElement*, una *Practice* se identifica con el atributo *Id_Practice* y describe atributos propios como: *Inputs_Practice* y *Outputs_Practice*. Esta clase abstrae toda la información relacionada con las prácticas (ágiles, colaborativas o cognitivas) que durante el proceso los niños tendrán en cuenta como un refuerzo aplicable para dar solución a las tareas. Es una generalización de la clase *ChildLetElement*, una versión más específica de esta clase, además, tiene una asociación de agregación con la clase *ChildLet Task*.

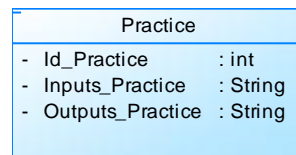


Figura 13. Definición Conceptual de la Clase Practice

- ✓ *Concepto (Concept)*: clase heredada de *ChildLetElement*, un *Concept* se identifica con el atributo *Id_Concept* y describe un atributo propio *Note_Concept*. Esta clase abstrae toda la información relacionada con los conceptos (de tipo colaborativo, ágil y cognitivo) que durante el proceso los niños tendrán en cuenta para solucionar sus tareas. Esta clase tiene una relación de agregación con la clase *ChildLet Task*. Esta clase es una generalización de la clase *ChildLetElement*, una versión más específica de esta clase, además, tiene una asociación de agregación con la clase *ChildLet Task*.

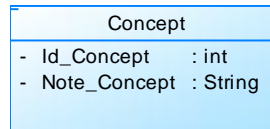


Figura 14. Definición Conceptual de la Clase Concept

- ✓ *Rol (Role)*: clase heredada de *ChildLetElement*, un rol se identifica con el atributo *Id_Rol*. Esta clase abstrae toda la información relacionada con los roles que durante el proceso los niños desempeñaran teniendo en cuenta las tareas asignadas. Esta clase es una generalización de la clase *ChildLetElement*, una versión más específica, además, tiene una asociación de agregación con la clase *ChildLet Task*. *Rol* representa el comportamiento que asumirá el niño como parte del desarrollo de la tarea dentro del proceso Child Programming.

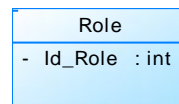


Figura 15. Definición de la Clase Role

- ✓ *Thinklet*: clase que contiene la información de un Thinklet, un Thinklet se identifica con el atributo único *Id_Thinklet*. Esta clase depende de la existencia de la clase *Collaborative Process* y abstrae toda la información del thinklet a través los atributos: *Name_Thinklet*, *Description_Thinklet*, *Inputs_Thinklet*, *Outputs_Thinklet*, *Steps_Thinklet*. Además, la clase thinklet tiene una relación de agregación con la clase *ChildLet Task*. *Thinklet* representa una unidad de construcción que especifica como un determinado patrón de colaboración debe utilizarse cuando el proceso está en ejecución.

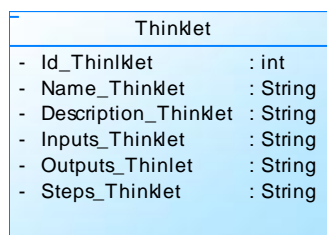


Figura 16. Definición de la Clase Thinklet

- ✓ *Patrón Colaborativo (Collaborative Pattern)*: clase que contiene la información de un collaborative pattern, un patrón de colaboración se identifica con el atributo *Id_CollaborativePattern*. Esta clase tiene una asociación de composición fuerte con la clase *Collaborative Process*. *Collaborative Pattern* representa una guía de ejecución del proceso.

Collaborative Pattern	
- Id_CollaborativePattern	: int
- Name_Pattern	: String
- Description_Pattern	: String

Figura 17. Definición de la Clase Collaborative Pattern

La tabla 4 relaciona las características del modelo conceptual antes descrito.

Clase	Clase Asociada	Tipo de Asociación	Rol	Cardinalidad
Team	Child	Composición	myChildren	2..*
Collaborative Process	Team	Composición	myTeam	1
	Facilitator	Composición	-----	1..*
	ChildLetElement	Composición	myChElementsmy	1..*
	ChildLetTask	Agregación	ChTask	1..*
Thinklet	Thinklet	Composición	myThinklets	0..*
	Thinklet	Composición		
ChildLetElement	Concept Practice Rol ChildLet Task	Generalización		
ChildLet Task	Concept	Agregación	myConcept	1..*
	Rol		myRoles	1..*
	Practice		myPractice	1..*
	Thinklet		myThinklet	1..*
Thinklet	Collaborative Pattern	Composición	myPatterns	1..*
Facilitator	Collaborative Process	Composición		
Child	Team	Composición		
Collaborative Pattern	Thinklet	Composición		

Tabla 4. Relación de Clases Child Programming

3.2.2 Actores del Modelo Child Programming

- **Profesor**²⁸: 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.
- **Estudiantes o Alumnos (Niños)**²⁹: participantes fundamentales, fuente principal de información y ejecutores de las actividades propuestas por el profesor.
- **Observadores (Investigador)**: personas encargadas de apoyar el proceso en cualquier momento teniendo en cuenta las disposiciones del profesor y observando de manera detallada del desarrollo de los niños en sus equipos para cada actividad propuesta.

²⁸ **RAE**: Persona que ejerce o enseña una ciencia o arte.

²⁹ **RAE**: Persona que cursa estudios en un establecimiento de enseñanza.

3.2.3 Dimensiones del Modelo Child Programming

- ✓ **Dimensión Cognitiva:** considerada como el esfuerzo que realizará un niño por comprender, analizar y apropiarse situaciones presentes en las tareas definidas por *ChildProgramming*, aportando al proceso los conceptos principales para el desarrollo de las mismas.

Esta dimensión permite a los niños afianzar su razonamiento lógico, la deducción, inducción, planificación y síntesis logrando adherirse al proceso y garantizar que los integrantes de los equipos de trabajo puedan explorar en sus conocimientos posibles formas de trabajo y además, puedan organizarse y adaptarse de manera que los logren alcanzar su objetivo dentro de la actividad propuesta [61].

- **Dimensión Ágil:** basados en la promulgación de los valores del manifiesto ágil antes mencionado, la dimensión ágil aporta a *ChildProgramming* una forma acorde de trabajo para alcanzar los objetivos donde se evidencie un trabajo de equipo que permita a los integrantes del mismo permanecer juntos a lo largo de la actividad, manteniendo un ambiente constante de cooperación y un entorno de apoyo que les permita a los equipos lograr finalizar el trabajo.

Child Programming a través de prácticas ágiles de desarrollo de software adiciona al proceso este componente de tal forma que los niños apropien estas prácticas a sus espacios de trabajo y puedan desarrollar sus actividades de manera simple y sencilla, aportando desde su percepción de niños dinámicas que garanticen un flujo de trabajo activo y ágil tal como lo proponen las teorías de desarrollo ágil descritas en el capítulo anterior.

- **Dimensión Colaborativa:** es el aprendizaje por descubrimientos es el eje central de los procesos educativos de hoy en día [57]. El aprendizaje colaborativo involucra experiencias que ayudan a la construcción de conocimiento de forma adecuada acorde al desarrollo de los niños en su entorno académico. Esta dimensión provee una forma efectiva de colaboración haciendo uso de prácticas y actividades que garanticen interacción y cooperación entre los integrantes de los equipos en torno a su trabajo.

Child Programming incorpora al proceso, estrategias colaborativas que aportan a la construcción del aprendizaje dentro del aula de clase, donde el alumno aprende más cuando genera su propio conocimiento desde la práctica y la interacción con otros compañeros. Dentro del proceso colaborativo se incluye la aplicación de técnica de aprendizaje colaborativo que permitan a los equipos de trabajo desarrollar sus actividades de forma dinámica. Ver Anexo A.

3.2.4 Elementos Child (*Child Elements*)

Child Programming se compone de elementos básicos que definen el proceso de desarrollo en torno a una actividad definida. Estos elementos parten del proceso de extracción descrito en el capítulo IV, los cuales son un complemento para la ejecución del proceso de desarrollo. A continuación se describe cada uno de ellos, los cuales fueron definidos en términos dados por los niños, de una forma entendible para ellos.

3.2.4.1 Roles

Child Programming ha definido los roles teniendo en cuenta el contexto de trabajo desarrollado con los niños. En la tabla 5 se indica los roles definidos por Child Programming.

Roles adaptados por Child Programming	
Internos del Proceso	
▪	Profesor
▪	Guía del Equipo
▪	Equipo de Trabajo
Externos al Proceso	
•	Investigadores/Observador Externo

Tabla 5. Comparativo de Roles Scrum - Child Programming. Fuente Propia

A continuación se describe cada uno de los roles propuesto por Child Programming.

- **Internos del Proceso**

- ✓ **Profesor:** es el responsable del proyecto de aprendizaje en el aula de clase, debe entregar a los equipos las pautas para la realización de la actividad, debe ser el encargado de entregar la misión a realizar, entrenar y monitorizar la metodología de trabajo y estar dispuesto a esclarecer dudas e inquietudes en cualquier momento del desarrollo. El profesor interviene en el proceso siempre, aunque no debe ser intrusivo en el trabajo del equipo. Además es el encargado de tomar las decisiones finales con respecto a la Misión, es quien define y determina los objetivos y requisitos teniendo en cuenta la temática a tratar.
- ✓ **Guía del Equipo:** es un aprendiz responsable de asegurar el desarrollo de la Misión acorde a las características y los requisitos de la misma. Debe estar pendiente de que el equipo este trabajando acorde a las prácticas, valores y reglas establecidas y que se avance en el desarrollo del trabajo según lo previsto. Trabaja igual que el resto del equipo y también es responsable de eliminar las

dificultades y de mantener un ritmo productivo como sea posible. El guía del equipo es escogido en consenso por los integrantes del mismo.

- ✓ **Equipo de Trabajo:** el equipo de trabajo es el conformado por (5±2) niños, quienes tienen la responsabilidad de organizarse para alcanzar las metas propuestas para la Misión y realizar cada tarea acorde a las características y especificaciones de las misma. El equipo de trabajo está implicado en la valoración del esfuerzo de sus integrantes, la priorización de cada tarea, la revisión de cada uno de sus resultados, mantener continua comunicación entre sí y sugerir soluciones o propuestas en cualquier momento específicamente en momento donde se presenten dificultades e inconvenientes en la realización de la Misión en las reuniones de trabajo continuas.

- **Externos al Proceso**

- ✓ **Investigador u Observador Externo:** es quien participa de la actividad observando el desarrollo de la misma, inicialmente con el profesor se encargan de describir la Misión y definir los objetivos de la misma, con el fin de determinar el objeto de la observación para la investigación a realizar.

3.2.4.2 Conceptos

Los conceptos en Child Programming son los elementos concretos que hacen parte de la dimensión Cognitiva expresados en un lenguaje acorde a los niños, los cuales incorporan conceptos claves del modelo y los conceptos derivados de las dimensiones Ágil y Colaborativa. Ver Anexo E.

Los conceptos son definiciones específicas las cuales fueron obtenidas a partir de la extracción definida en el capítulo IV, donde se describen y asocian estos al proceso de trabajo, para facilitar a los equipos una mejor comprensión de la actividad a desarrollar.



Figura 18. Composición Componente Cognitivo- Fuente Propia

En la tabla 6 se presentan los conceptos definidos por Child Programming con su respectiva descripción.

CONCEPTO	SIGNIFICADO
Conceptos Dimensión Cognitiva	
<i>Cumplimiento</i>	Terminar, lograr u obtener la realización de una misión o parte de ésta, alcanzando la meta definida.
<i>Regla</i>	Modo de hacer o desarrollar la actividad propuesta.
<i>Condición</i>	Aceptación de requisitos que deben cumplir los niños para realizar la actividad.
<i>Instrucción</i>	Conjunto de enseñanzas y prácticas adquiridas por los niños e impartidas por el profesor.
<i>Inquietud</i>	Curiosidad o interés que tengan los niños frente algún tema o explicación.
<i>Entendimiento</i>	Comprensión de temas o enseñanzas.
<i>Comunicación</i>	Dar a conocer algo a los demás integrantes del equipo alguna información específica.
<i>Comentario</i>	Hablar o dar opinión sobre algo en particular que alguno de los niños quiera expresar dentro del equipo.
<i>Tema</i>	Ideas, sobre algo en particular, un contenido cualquiera que se vaya a tratar en la actividad.
<i>Concepto</i>	Expresión que dan los niños en palabras sobre algo particular.
Conceptos Componente Ágil	
<i>Reunirse</i>	Unirse con los compañeros del equipo para dar inicio a la actividad.
<i>Realizar</i>	Hacer efectiva las tareas que completan la actividad.
<i>Utilizar</i>	Emplear todo el material o recursos disponibles para la actividad de tal forma que el equipo cuente con lo necesario para su desarrollo.
<i>Lugar de Trabajo</i>	Espacio ocupado por el equipo en el cual se desarrolla la actividad.
<i>Entregar</i>	Dar o facilitar la terminación de una tarea o actividad.
Conceptos Componente Colaborativo	
<i>Motivación</i>	Estimular el equipo para desarrollar las tareas de la manera efectiva.
<i>Interacción</i>	Compartir de forma coordinada con un compañero.

<i>Tareas</i>	Acción para realizar en un tiempo determinado.
<i>Cooperación</i>	Actuar conjuntamente con otro u otros compañeros de equipo para un mismo fin.

Tabla 6. Relación de Conceptos Definidos por Child Programming

3.2.4.3 Prácticas

Las prácticas en Child Programming se constituyen como el núcleo del proceso, donde los equipos de trabajo las aplican de forma efectiva para el cumplimiento de la Misión. En la figura 19 se muestra una representación de la relación transversal que tienen las prácticas descritas en Child Programming.



Figura 19. Prácticas Child Programming- Fuente Propia

En la tabla 7 se describen las prácticas definidas por Child Programming de acuerdo al componente que describe cada una de ellas, Ver Anexo F.

IDENTIFICADOR	NOMBRE DE LA PRÁCTICA	COMPONENTE AL QUE PERTENECE
ChP01	Aceptar las condiciones para desarrollar la actividad	Colaborativo
ChP02	Cumplir las reglas del juego	Cognitivo
ChP03	Desarrollar la actividad en equipo	Colaborativo
ChP04	Comprometerse para trabajar en equipo	Colaborativo
ChP05	Reunirse con un compañero(a) y realizar la tarea	Ágil
ChP06	Utilizar todo el lugar de trabajo con el equipo para informarse de la actividad	Ágil
ChP07	Hacer la tarea de forma sencilla, realizándola cada vez mejor	Ágil
ChP08	Preguntar lo que no se entiende	Cognitivo
ChP09	Entregar la tarea hecha	Ágil
ChP10	Entender el tema de la actividad	Cognitivo

Tabla 7. Relación de Prácticas definidas por Child Programming

3.2.4.4 Especificación de las prácticas

En Child Programming, para especificar como están formadas las prácticas se han definido unas plantillas de acuerdo al componente al que pertenece de la siguiente forma:

Plantilla Prácticas Cognitivas

- ✓ **Identificador:** que representa al valor único de tipo alfanumérico que la práctica.
- ✓ **Nombre de la Práctica Cognitiva:** corresponde al nombre de la práctica definida, cuyo nombre es específicamente dado por los niños.
- ✓ **Descripción:** describe de forma clara y sencilla de que se trata la práctica y además incluye algunas recomendaciones útiles en el momento de usarla.
- ✓ **Notas:** en caso de aclarar algo para la realización de la actividad que involucre la práctica.


IDENTIFICADOR:	ChP10
NOMBRE DE LA PRACTICA COGNITIVA:	"Entender el Tema de la Misión"
DESCRIPCIÓN:	 <p>Los integrantes del equipo deben tener claro los conceptos de la temática expuesta para la actividad propuesta, de tal manera que cada uno esté en la capacidad de dar a conocer el conocimiento que tiene sobre dicha temática a final de la actividad.</p> <p>Para entender los conceptos propuestos en la actividad es recomendable:</p> <ul style="list-style-type: none"> • El Equipo de trabajo una vez iniciada la actividad deben recibir del "Profesor" la guía para desarrollarla. • El "Profesor" debe distribuir en los equipos los contenidos del tema a tratar para la actividad propuesta. • El Equipo debe leer bien y comprender la información entregada según la temática correspondiente y dar inicio al desarrollo de la actividad. • Al finalizar la actividad el "Profesor" debe recibir de cada equipo las evidencias físicas (estructuras, carteleras, documentos, mapas conceptuales, descripciones, diseños, códigos de programas, test, etc) que demuestren la comprensión de los conceptos.
NOTAS:	<ul style="list-style-type: none"> ❖ Las evidencias que los equipos deben entregar son planificadas por "Profesor" de acuerdo a la temática a tratar en la actividad y bajo su libertad.

Figura 20. Plantilla Práctica Cognitiva

Plantilla Prácticas Ágiles

- ✓ **Identificador:** que representa al valor único de tipo alfanumérico que la práctica.
- ✓ **Nombre de la Práctica Ágil:** corresponde al nombre de la práctica definida, cuyo nombre es específicamente dado por los niños.
- ✓ **Descripción:** describe de forma clara y sencilla de que se trata la práctica y además incluye algunas recomendaciones útiles en el momento de usarla.
- ✓ **Proceso:** el cual incluye *entradas* y *salidas* que representan los insumos necesarios para aplicar la práctica, así como los insumos generados después de su aplicación.
- ✓ **Guía:** representa el flujo normal para la aplicación correcta de la práctica.
- ✓ **Sugerencias:** recomendaciones útiles para el desarrollo de la actividad aplicando la práctica.
- ✓ **Prácticas Relacionadas:** prácticas que tienen similitud con la práctica aplicada.

- ✓ **Conceptos Relacionados:** conceptos que se relacionan a la práctica y que son útiles para aclarar cualquier duda e inquietud.


IDENTIFICADOR:		ChP07
NOMBRE DE LA PRACTICA AGIL:		"Hacer la Tarea de Forma Sencilla, Realizándola cada vez Mejor"
DESCRIPCIÓN:		
		Consiste en realizar un diseño simple, fácil y entendible, evitando posibles complicaciones en el desarrollo de la actividad y realizando frecuentes incrementos.
PROCESO		
Entradas:		Salidas:
<ul style="list-style-type: none"> Listado de lluvia de ideas con posibles soluciones, referentes al diseño de la actividad. Documento con propuesta de diseño plasmada a través de un esquema, dibujo o gráfico que represente la solución de la actividad. 		<ul style="list-style-type: none"> Documento de diseño simple con una solución previa a la solución final. Un entregable tangible de la solución a la actividad propuesta (según sea el caso).
GUIA		
<ul style="list-style-type: none"> El "Equipo de Trabajo" deberá generar una lluvia de ideas acorde al diseño que se vaya a realizar y que será requerido. El "Equipo de Trabajo" deberá analizar la propuesta más indicada con el fin de determinar el diseño que se ajuste a la propuesta, este diseño debe ser simple y sencillo. Observar y discutir continuamente el diseño que se está realizando en el "Equipo" para encontrar posible errores y corregirlos. 		
SUGERENCIAS		
<ul style="list-style-type: none"> Leer detenidamente la tarea asignada, para comprender mejor de que se trata. Realizar un borrador en papel sea un gráfico, un dibujo o un esquema del diseño posible para resolver la tarea. 		
PRACTICAS RELACIONADAS		CONCEPTOS RELACIONADOS
<ul style="list-style-type: none"> ✓ "Reunirse con un compañero(a) y realizar la tarea" ✓ "Utilizar todo el lugar de trabajo con el equipo para informarse de la actividad" 		<ul style="list-style-type: none"> ✓ "Cumplir las reglas del juego" ✓ "Preguntarlo que se no entiende" ✓ "Entender el tema de la actividad"

Figura 21. Plantilla Práctica Ágil

Plantilla Prácticas Colaborativas

- ✓ **Identificador:** que representa al valor único de tipo alfanumérico que la práctica.
- ✓ **Nombre de la Práctica Colaborativa:** corresponde al nombre de la práctica definida, cuyo nombre es específicamente dado por los niños.
- ✓ **Descripción:** describe de forma clara y sencilla de que se trata la práctica y además incluye algunas recomendaciones útiles en el momento de usarla.
- ✓ **Proceso:** el cual incluye *entradas* y *salidas* que representan los insumos necesarios para aplicar la práctica, así como los insumos generados después de su aplicación.
- ✓ **Guía:** representa el flujo normal para la aplicación correcta de la práctica.
- ✓ **Sugerencias:** recomendaciones útiles para el desarrollo de la actividad aplicando la práctica.


IDENTIFICADOR:	ChP01
NOMBRE DE LA PRACTICA COLABORATIVA:	"Aceptar las Condiciones para Desarrollar la Misión"
DESCRIPCIÓN:	
 <p>Los equipos de trabajo conformados en el proceso tienen claro las condiciones, normas o reglas establecidas para la actividad y las asumen en todo momento.</p>	
PROCESO	
Entradas:	Salidas:
<ul style="list-style-type: none"> • Guía de la actividad propuesta. • Instrucciones del profesor. 	<ul style="list-style-type: none"> • Compromiso de los integrantes del equipo reflejado en su participación en el transcurso de la actividad.
GUÍA:	
<ul style="list-style-type: none"> • El profesor entrega al equipo la guía para desarrollar la actividad. • El profesor imparte las instrucciones y aclara las dudas que se generen durante la explicación, si las hay. • El equipo acepta las condiciones para desarrollar la actividad y empieza a trabajar. 	

Figura 22. Plantilla Práctica Colaborativa

3.3 PROCESO CHILD (CHILD PROCESS)

El proceso Child Programming está basado en la aplicación de prácticas ágiles, colaborativas y cognitivas, exploradas y empíricamente evaluadas con equipos de niños de edad escolar (4 y 5 de primaria). Su ciclo de vida se basa en un marco de trabajo que comprende las siguientes fases: Pre- Juego, Juego y Post-Juego. La figura 23 presenta el ciclo de vida del modelo propuesto, con sus fases y actividades principales.

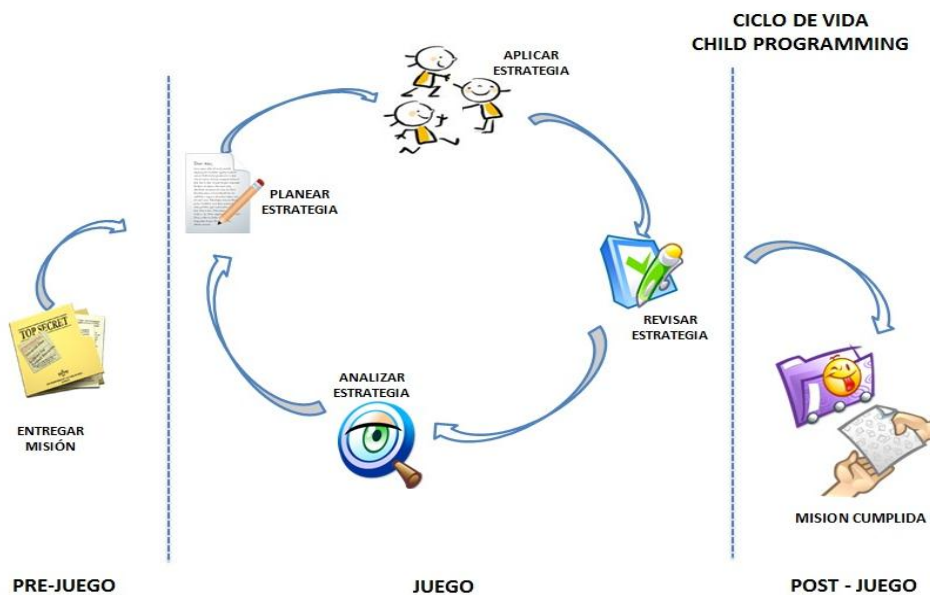


Figura 23. Ciclo de Vida propuesto por Child Programming. Fuente Propia

3.3.1 Pre - Juego

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.

En el contexto de Child Programming, en esta fase el *Profesor* hace entrega de “La Misión” al *Equipo de Trabajo* el profesor. En consenso con los niños se dió este nombre tal como se describe en el capítulo IV – Caso 2.

La misión contiene la descripción detalla de cada donde cada integrante de cada equipo podrá conocer el objetivo y las metas propuestas para la actividad. En esta estación también el *Profesor* entrega al *Guía del Equipo* el material necesario como apoyo para la realización y cumplimiento de la misión.

En esta estación los *Equipos de Trabajo* identifican las tareas y las priorizan teniendo en cuenta su criterio particular para la primera ronda.

3.3.2 Juego

En esta fase se realizan una serie de iteraciones denominadas por Child Programming *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: *Planear la Estrategia*, *Aplicar la Estrategia*, *Revisar la Estrategia* y *Analizar la Estrategia*.

- **Planear la Estrategia**

En esta estación la planificación de la estrategia involucra definir las tareas y su priorización. 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. Así mismo, 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.

- **Aplicar la Estrategia**

Después de establecer su planeación el *Equipo de Trabajo* entra en la ejecución de sus tareas, guiado por su *Guía de Equipo*, quien trabaja a la par con sus compañeros de equipo. 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*. 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.

- **Revisar la Estrategia**

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*.

- **Analizar la Estrategia**

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. 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*, sea porque haya sido acertada o porque no dio resultado y de forma inmediata reorganizan la forma de trabajar para la siguiente *Ronda*. Esta estación permite al *Equipo de Trabajo* y al *Guía del Equipo*, analizar posibles situaciones futuras y tomar medidas oportunas para mitigar errores o complicaciones potenciales.

3.3.3 Post – Juego

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. 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 refleja un proceso de aprendizaje y que haya permitido, de forma efectiva, cumplir con los objetivos de aprendizaje propuestos. En la figura 24 se detalla el diagrama de flujo de Child Programming.

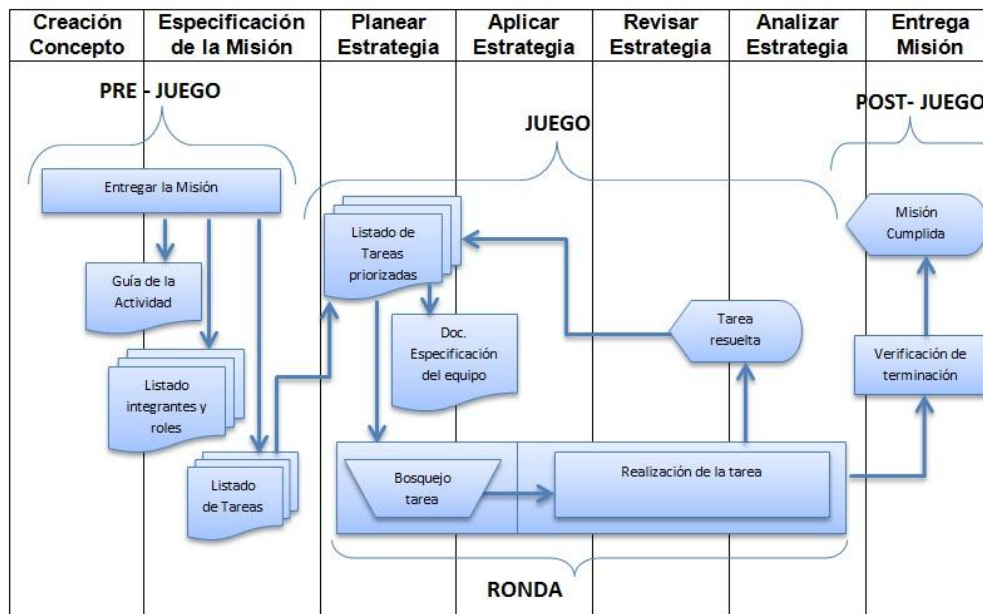


Figura 24. Diagrama de Flujo Proceso Child Programming. Fuente Propia

3.4 HERRAMIENTAS DE SOPORTE CHILD

3.4.1 Modelado en EPFC (Eclipse Process Framework Composer)

Modelar ChildProgramming, busca que quienes sean responsables de ejecutarlo tengan un marco de trabajo, donde se encuentre la información de las tareas, roles, productos de trabajo, patrones de colaboración, thinklets, etc, involucrados en la actividad colaborativa con el fin de tener un fácil acceso a la información para futuras implementaciones o aplicaciones.

Para este modelado se utilizó el estándar de metamodelado para la Ingeniería de Procesos de Software SPEM 2.0 (Software Process Engineering Metamodel) [63], a través de la herramienta EPF Composer³⁰ que permite definir, gestionar y reutilizar un repositorio de métodos y procesos.

A continuación se detalla la descripción del modelo ChildProgramming en EPFC:

Descripción del Modelado

- **Paquete de Contenido (*content package*)** donde se definen los elementos básicos del proceso ChildProgramming.

³⁰ <http://www.eclipse.org/epf/>

- Dentro del paquete de contenido **Tareas Child** se incluyen los elementos Child correspondientes: *roles, tareas, productos de trabajo, guías*.
 - ✓ **Roles:** corresponde a la descripción de los roles que participan en childprogramming, por ejemplo: *Guía_del_Equipo*.
 - ✓ **Tareas Child:** corresponde a la descripción de las tareas involucradas en childprogramming, las cuales determinan el desarrollo de la actividad propuesta, por ejemplo: *Analizar la Estrategia*.
 - ✓ **Productos de Trabajo:** corresponden a productos entregables, artefactos o resultados que ingresan o salen dentro del proceso, por ejemplo: *Entregable de la Tarea Asignada*.
 - ✓ **Guías:** en childprogramming las guías describen un conjunto de prácticas, conceptos, patrones, thinklets, etc, que aportan formas de trabajo dentro del desarrollo de una actividad, por ejemplo: *Cumplir las Reglas del Juego*.

En la figura 25 se presentan los elementos de contenido del proceso colaborativo *Child*.

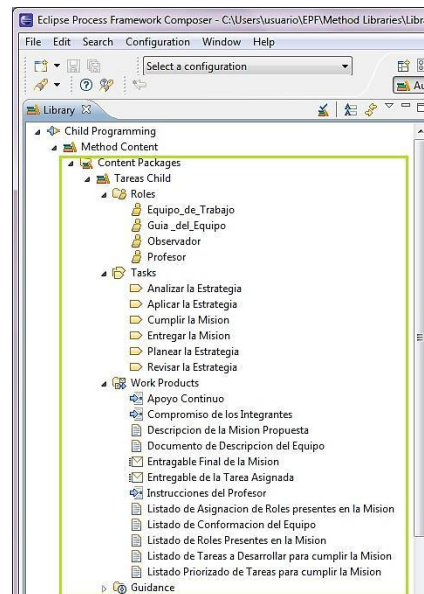


Figura 25. Contenido del Método Asociado al Proceso Colaborativo Child

Los elementos de contenido en el proceso de ChildProgramming (tareas, roles, productos de trabajo y guías), son categorizadas en categorías personalizadas (*Custom Categories*) establecidas para identificar de forma más detallada en la publicación del proceso.

Adicionalmente se realizó la configuración *Child*, la cual hace parte de la librería de métodos que permite restringir la vista de todos los elementos únicamente a los elementos requeridos.

Otra parte importante dentro del modelado es la creación del proceso. Una vez definidos los elementos del contenido, se creó un patrón de Proceso de Entrega (*Delivery Process*), definido por la estructura de desglose (*Work BreakDown Structure, WBS*) para representar el proceso de ejecución de *ChildProgramming*. En la figura 26 se puede observar la estructura de desglose definida en *ChildProgramming*.

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeat...	Multipl...
ChildProgramming_Proces	0			Delivery Pro...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pre-Juego	1			Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entregar la Mision	2			Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Juego	3	1		Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Analizar la Estrategia	4	2		Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aplicar la Estrategia	5	4		Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Planear la Estrategia	6	5		Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Revisar la Estrategia	7	6		Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Post-Juego	8	3		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cumplir la Mision	9			Task Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 26. WBS ChildProgramming

A partir del modelado que se realizó y haciendo uso de la herramienta EPFC, se generaron diferentes diagramas. En el Anexo M, puede ver algunos de los diagramas obtenidos.

3.4.2 Descripción Sitio Web *ChildProgramming*

ChildProgramming cuenta con un sitio web dinámico como resultado de la investigación, diseñado con el fin de permitir a los usuarios interactuar y generar experiencias a través de proyectos de aula, los cuales pueden ser compartidos en línea con otros usuarios y demás interesados, foros para compartir ideas, dudas y comentarios, ficha de contactos para dirigirse a los profesores o guías del proyecto quienes generaran respuestas oportunas a sus preguntas.

El sitio web *ChildProgramming* es un desarrollo que se publica bajo la dirección o URL <http://www.unicauca.edu.co/childprogramming>, cuyo dominio pertenece a la Universidad del Cauca. A continuación se describe brevemente las secciones que componen el sitio y algunas de las actividades que pueden realizarse. En el Anexo N, puede ver más detalle sobre la especificación técnica del sitio web.



Figura 27. Captura Usuario Anónimo - Página de Inicio

3.4.3 Descripción Ambiente de Desarrollo ChildProgramming

El ambiente en ChildProgramming es un elemento fundamental que sirve de soporte para el modelo, involucrando diferentes aspectos y desde diferentes perspectivas. Para ello el ambiente ChildProgramming se ha configurado de la siguiente manera:

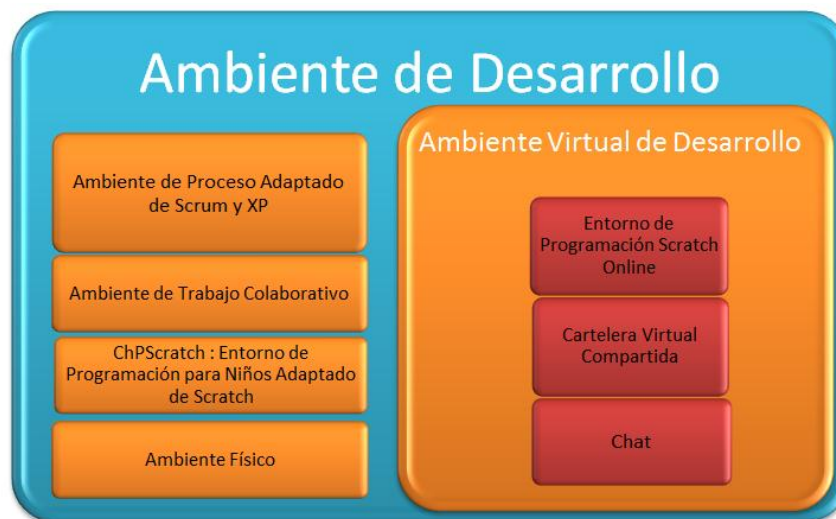


Figura 28. Ambiente de Desarrollo ChildProgramming. Fuente Propia

3.4.3.1 Ambiente de Desarrollo

El ambiente de desarrollo propuesto para ChildProgramming, está basado en aspectos de las metodologías de desarrollo ágil XP y Scrum y del Trabajo Colaborativo, los cuales

fueron adaptados al definir un espacio compartido de trabajo (tablero de tareas) de Scrum, reuniones de pie (reunión diaria) de Scrum, programación en parejas (pair programming) de XP y las fichas de control de tiempo del trabajo colaborativo. Además está compuesto por un entorno de programación adaptado desde Scratch, así como un ambiente virtual de desarrollo para soportar el desarrollo de software con grupos distribuidos.

Tomando como soporte los patrones descritos por Hansen [78] el modelo ChildProgramming define:

- **En cuanto al Contexto:** para proveer un mejor ambiente de trabajo, a cada equipo se brindará la confianza necesaria dándoles también, la posibilidad de decidir y equilibrar sus fuerzas acordes al desarrollo de sus actividades.
- **En cuanto al Lugar:** es necesario que los equipos mantengan una comunicación continua y efectiva por ellos se recomienda que estos sean pequeños tal como se define en este trabajo. Además, los integrantes de los equipos deben tener acceso de forma fácil al material que les permita escribir, debatir, pensar y visualizar, por ello ChildProgramming propone la cartelera de trabajo y papeles en general donde los integrantes de los equipos pueden escribir y participar de la información dentro del mismo espacio de trabajo y compartir sus experiencias, así como participar de pequeñas reuniones que les permitan conocer, aclarar alguna situación que se presente.
- **En cuanto al Espacio:** proveer espacios flexibles de trabajo para los equipos, este espacio debe estar dispuesto teniendo en cuenta el trabajo en equipo y el trabajo por parejas para lo cual debe ser flexible para los integrantes de los equipos, que les permita en cualquiera de los casos comunicarse y desarrollar sus actividades de forma adecuada. Se debe contar con los muebles y elementos necesarios. Las sillas deben ser adecuadas, cómodas y fáciles de manipular para que permitan la movilidad de los integrantes de los equipos, las mesas de trabajo también, deben ser flexibles y permitir la movilidad, los demás elementos útiles para el desarrollo del trabajo deben ser fáciles de manipular y deben ser fáciles de trasladar de un lugar a otro.
- **En cuanto al Detalle:** brindar a los integrantes de los equipos beneficios como momentos de descanso, que les permitan refrescar sus ideas y retomar sus trabajos con ánimo y disposición. Más detalle ver Anexo P.

3.4.3.2 Ambiente de Programación

En ChildProgramming se adaptó el ambiente de programación para niños Scratch denominado *ChPScratch* a partir de una configuración para trabajo en red, que permite realizar trabajo compartido para facilitar la separación de tareas y la prueba de soluciones conjuntas antes de ser integradas, además, que algunas de las funcionalidades fueron modificadas como los menús, el direccionamiento al sitio web de ChildProgramming y el

idioma. Se encuentra habilitada la opción para acceder al sitio web en cualquier momento para obtener más información del modelo de aprendizaje. La figura 29 muestra una adaptación del ambiente, que se encuentra más en detalles en el Anexo O.

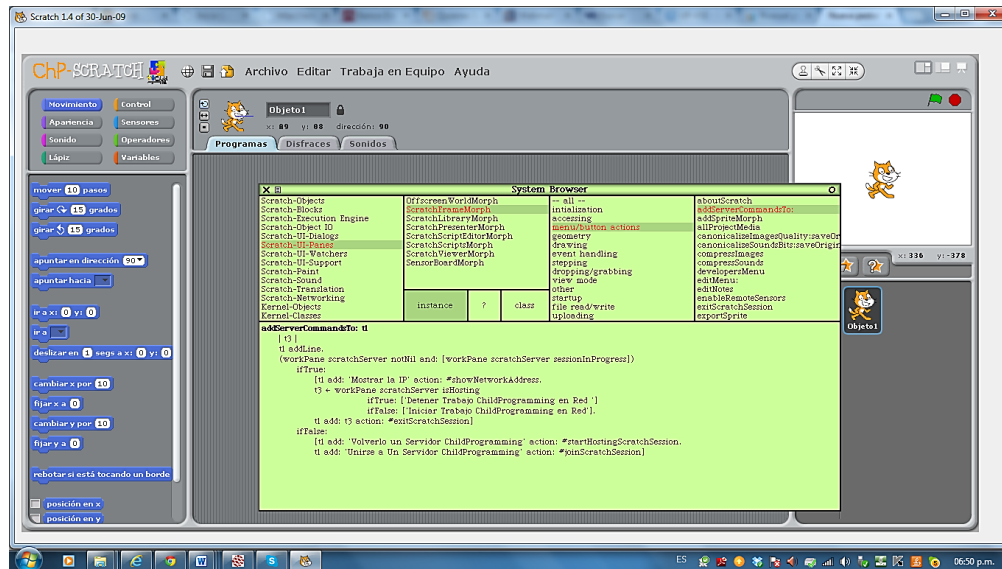


Figura 29. Adaptación del Entorno de Programación ChPScratch

3.4.3.3 Ambiente Virtual

El ambiente virtual que soporta ChildProgramming fue diseñado con el fin de brindar a los niños un espacio que recrea el ambiente de trabajo en el aula y les permite la interacción entre ellos únicamente cuando no sea posible la interacción física entre los miembros del equipo ChildProgramming, además, este espacio les permite acceder desde sus casas y por ejemplo desarrollar sus tareas escolares en equipo. Este espacio se encuentra en el sitio web diseñado en la pestaña “Desarrollo” a la cual pueden acceder una vez se hayan registrado previamente en el sitio. Una vez autenticado el niño, encontrará tres servicios que hemos integrado: el entorno de programación para niños Scratch, en el cual se puede desarrollar cualquier aplicación en línea y ejecutar cualquiera de las opciones del menú Scratch, la cartelera virtual compartida que es un espacio en el que los integrantes de cada equipo podrán registrar sus tareas y estar pendientes del avance y ejecución de cada una de estas en tiempo real, y finalmente se creó el Chat el cual va a permitir la comunicación e interacción directa de los niños con sus equipos de trabajo y con los demás equipos. La Figura 30 muestra el ambiente virtual diseñado para éste propósito, éste ambiente está descrito con mayor detalle en el Anexo Q.



Figura 30. Ambiente Virtual de Desarrollo ChildProgramming

4. EXTRACCIÓN Y EVALUACIÓN DEL MODELO CHILD PROGRAMMING: ESTUDIOS DE CASO

En este capítulo se recogen todas las experiencias en la aplicación de las actividades relacionadas con los estudios de caso aplicados para la extracción y evaluación del modelo Child Programming, para que pueda utilizarse como marco de referencia en instituciones educativas interesadas en trabajo colaborativo desde el punto de vista del proceso definido en este trabajo.

4.1 METODOLOGIA

Yin [64] considera el estudio de caso cómo método de investigación apropiado para temas que se consideran prácticamente nuevos. Chetty [65] indica que el método de estudio de caso es una metodología adecuada para investigar fenómenos en los que se busca dar respuesta a cómo y por qué ocurren los fenómenos, permite estudiar un tema determinado, permite explorar en forma más profunda y obtener un conocimiento más amplio sobre cada fenómeno, lo cual permite la aparición de nuevas señales sobre los temas que emergen y además juega un papel importante en la investigación.

Runeson et al. [66], plantean que el estudio de caso es una metodología de investigación que estudia un fenómeno contemporáneo en su contexto real, buscando mantener la integridad y las características significativas de los eventos, y es ejecutado cuando el investigador tiene poco control sobre los eventos y cuando los sujetos de estudio son más fáciles de observar en grupo que de manera aislada.

Trabajar con esta técnica en este caso permite al investigador generalizar desde una instancia concreta a un aspecto más general, ofrecer fuentes de datos de los que se pueden hacer análisis posteriores y así, generar futuros trabajos de investigación y en este caso aportar a partir de experiencias reales una contribución a cambiar prácticas a nivel educativo.

Tal como lo plantea Yin [64] es necesario formular el esquema metodológico de la investigación el cual soportará el trabajo a realizar y permitirá su posterior validación. La validez se va desarrollando a lo largo de todo el estudio, en cada una de sus etapas.

En la figura 31 se muestra el procedimiento metodológico utilizado para la investigación.

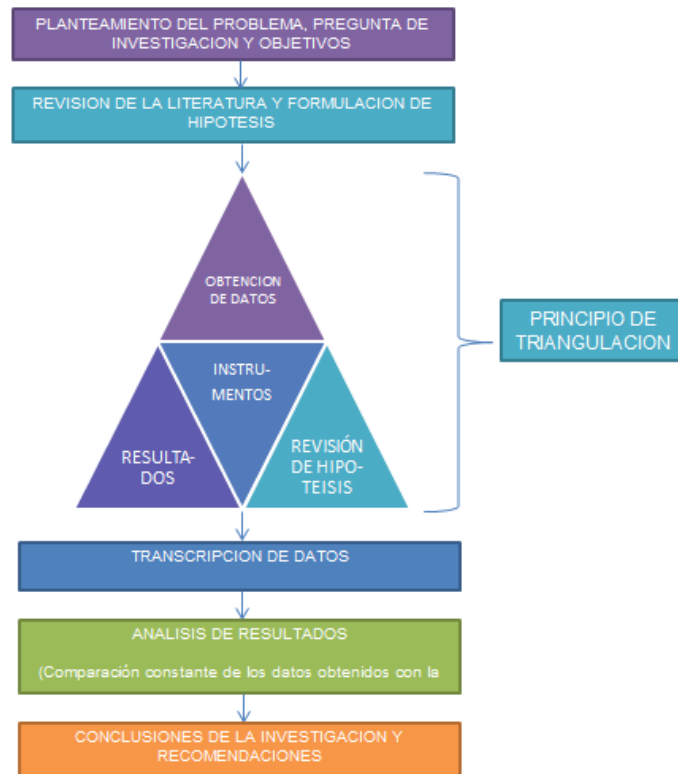


Figura 31. Procedimiento Metodológico Seguido para esta Investigación – Basado en Yin Fuente Propia

4.1.1 Instrumentos de Evaluación

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

- **Entrevistas:** 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. Se considera que este método es más eficaz que el cuestionario porque permite obtener una información más completa y para el caso resulta menos incómodo para los encuestados que son niños poder contestarlas porque les da más libertad de pensar en sus respuestas sin presión alguna.
- **Observación de campo:** este método establece una relación concreta e intensiva entre el equipo de investigación y el hecho social o los actores sociales de los que se obtienen datos que luego se sintetizan para desarrollar la investigación. La observación es un procedimiento de recolección de datos e información que consiste en utilizar los

sentidos para observar hechos y realidades sociales presentes y a la gente donde desarrolla normalmente sus actividades.

- **Archivos de datos:** los archivos tienen la responsabilidad de ser fuentes primarias de ilustración y mejoramiento para todos los sectores de la comunidad, en tanto contienen información de documentos que son texto de primera mano, que pueden y deben aprovecharse en el trabajo creador, por parte de investigadores y estudiosos.
- **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. Es una herramienta apropiada para detectar problemas o defectos que estén ocurriendo en la recolección de la información.
- **Documentos:** Resultantes del trabajo bajo investigación.



Figura 32. Instrumentos de Evaluación- Fuente Propia

4.2 CONTEXTO DE LA INVESTIGACIÓN

El desarrollo de esta investigación se realizó en las instalaciones de la Institución Educativa Técnico Industrial sede San Camilo de la ciudad de Popayán, fundada por los Hermanos Maristas, el 6 de Octubre de 1932. Ubicada sobre la comuna 4, en el barrio "San Camilo" y más específicamente en la Calle 10 # 9 - 82. Actualmente alberga a aproximadamente 435 estudiantes entre niñas y niños, con un énfasis humanístico y una modalidad académica. Los estudiantes están distribuidos de la siguiente manera:

- Nivel de Educación Preescolar (Transición): alumnos de 4 y 5 años.
- Nivel de Educación Básica (Ciclo Primaria): alumnos de 5 a 11 años de 1º a 5º grado.

La composición étnico-cultural de la población es homogénea, con mayoría de raza mestiza. Estudiantes procedentes de estratos medio 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 2 y 3, con dedicación en actividades comerciales independientes y empleados permanentes, con un ingreso medio bajo. Encontramos además una comunidad muy católica, marcando los planes de estudio con aspectos religiosos, desde el Preescolar hasta el grado 5º, con intensidad horaria fija en cada curso.

La institución cuenta actualmente con el personal administrativo y académico en cabeza del coordinador y el cuerpo de profesores conformado por 12 profesionales en la educación básica primaria, 1 docente del área de idiomas, 1 docente de área de tecnología, 1 docente del área de educación física y un profesional en psicología.

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 10 computadores de escritorio con menos de un año de uso donados por la alcaldía municipal de Popayán, 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, también, con menos de un año de uso y acceso a internet como parte de un convenio entre la institución, la alcaldía de Popayán y la empresa de telecomunicaciones de Popayán – Emtel.

Para cumplir los objetivos de esta investigación se han diseñado 3 estudios de caso (dos exploratorios y uno confirmatorio) dentro del mismo contexto de investigación a fin de obtener de cada uno de ellos los elementos y las características necesarias para formalizar el modelo Child Programming propuesto para este trabajo.

4.3 ESTUDIOS DE CASO

4.3.1 ESTUDIO DE CASO 1

Este estudio de caso está enfocado en la extracción de dinámicas propias de los equipos de trabajo, a través de una actividad colaborativa que permita evidenciar las prácticas y estrategias que los equipos emplean para dar una solución y las formas de organización y desempeño entorno a la realización de la actividad propuesta.

4.3.1.1 Pregunta de Investigación

Para conocer como los niños se organizan en equipos y resuelven situaciones de forma lógica a través de actividades lúdicas, este primer estudio de caso pretende resolver la siguiente pregunta:

¿Cómo se organizan y desempeñan los equipos de niños entre 8 y 10 años de edad de los grados cuarto y quinto de básica primaria de la Institución Educativa Técnico Industrial sede San Camilo, para resolver construcciones de tipo estructural?

4.3.1.2 Objetivo del Estudio

Explorar los mecanismos utilizados libremente a nivel de organización y de dinámica de equipos de niños con edades comprendidas entre los 8 y 10 años de edad, utilizados libremente para alcanzar la solución de un problema que requiere la construcción colectiva de una estructura tangible.

4.3.1.3 Selección del Estudio

Debido a que las actividades principales de este estudio van orientadas al trabajo específicamente con niños, son ellos los indicados para poner en práctica y evaluar esta propuesta. Así, la unidad de análisis para este caso son los equipos de trabajo. El estudio de caso es de tipo **Embebido** considerando 29 unidades de análisis con 177 sujetos de investigación. Las unidades de análisis seleccionadas son de tipo **Aleatoria**³¹ en su conformación y de tipo **Revelatorio**³² lo que lo hace suficientemente completo para evaluar las características y comportamientos de los sujetos dentro de sus equipos de trabajo.

4.3.1.4 El Estudio de Caso y los Sujetos de Investigación

Este estudio de caso es de tipo embebido [64], donde es considerada como unidad de análisis los 29 equipos de trabajo conformados por niños con edades comprendidas entre los 8 y 10 años, de los grados cuarto y quinto de básica primaria, quienes para efectos de este caso se constituyen como las fuentes primarias de información, en total 177. Este estudio de caso tiene un enfoque exploratorio dado que pretende evidenciar las formas de organización y desempeño en los equipos de trabajo.

Teniendo en cuenta las dinámicas que los niños aplican en sus cursos académicos en la institución, se diseñó el estudio de caso como cinco sesiones de trabajo, una sesión por cada curso, en la cual participaban los equipos definidos para cada curso así: cuarto A, 5 equipos, cuarto B, 6 equipos, cuarto C, 6 equipos, quinto A, 6 equipos, quinto B, 6 equipos, con el fin de observar en cada uno de ellos características propias de

³¹ Un estudio de caso es de asignación aleatoria de los sujetos para constituir los grupos que harán parte de la investigación.

³² Un estudio de caso revelatorio existe cuando un investigador tiene la oportunidad de observar y analizar un fenómeno previamente inaccesible a la investigación científica.

desempeño, organización y demás dinámicas de trabajo en equipo enfocadas a la abstracción, secuenciación y estructuración lógica de formas poliédricas.

Es de resaltar que todos los sujetos de investigación en este estudio de caso contaron con las mismas condiciones de trabajo y todos presentaron características similares en cuanto a conceptos lógicos, operacionales y formales del lenguaje, propios para cada grado académico, donde se aplicó el estudio de caso.

4.3.1.5 Indicadores y Mediciones

Para obtener la información necesaria para éste Estudio de Caso, particularmente, y dar respuesta a la pregunta de investigación fue necesario definir un conjunto de métricas e indicadores. La tabla 8 muestra un resumen de los indicadores y métricas definidos

PREGUNTA DE INVESTIGACIÓN	INDICADOR	MEDICIÓN	INSTRUMENTO
<p><i>Cómo se organizan y desempeñan los equipos de niños entre 8 y 10 años de edad de los grados cuarto y quinto de básica primaria de la Institución Educativa Técnico Industrial sede San Camilo, para resolver construcciones de tipo estructural?</i></p>	<p><i>Formas de Organización y Dinámicas que exhiben los equipos con mayor frecuencia y desempeño</i></p>	Nivel de Comportamiento observado en los equipos de trabajo	Observación, Protocolo de Observación
		Nivel de Productividad Alcanzada por los equipos de trabajo	Observación, Evaluación de Conformidad del Investigador
		Nivel de Calidad con respecto al producto entregado por los equipos de trabajo	Observación, Registro de Defectos del Artefacto Construido
		Frecuencia de formas de organización y dinámicas que exhiben los equipos con mayor grado de Comportamiento ($\geq 50\%$), Productividad ($\geq 50\%$) y Calidad ($\geq 50\%$)	Lista de Chequeo

Tabla 8. Indicadores y Mediciones Estudio de Caso 1

A continuación se describen en detalle los indicadores y la forma en que estos son calculados a través de las métricas identificadas:

- **Indicador**

- ✓ **Formas de Organización y Dinámicas:** representa las formas como se organizan y las dinámicas que presentan los equipos de trabajo en el desarrollo de la actividad, teniendo en cuenta los resultados obtenidos por la observación.

▪ **Mediciones**

- ✓ **Comportamiento:** es la forma de actuar o reaccionar de un individuo frente a una situación, problema o actividad. Trata de como también las personas u organismos proceden frente a los estímulos en relación con el entorno³³. Esta medición permite medir un conjunto de conductas comunes mediante el instrumento de observación (Protocolo de Observación) relacionados con la actividad, que los niños en sus equipos de trabajo exhiben con mayor frecuencia. La fórmula que hemos definido para valorar estas conductas está dada por (1):

$$C = \left\{ \frac{1}{8} [(NR_S * 100) + (NR_A * 50) + (NR_I * 10)] \right\} / 100 \quad (1)$$

Donde *C* corresponde a comportamiento, *NR_S* es el número de respuestas satisfactorias, *NR_A* es el número de respuestas aceptables, *NR_I* es el número de respuestas insatisfactorias. El factor (1/10) se relaciona con el número de características contenidas en el instrumento de observación (Protocolo de observación). El valor 100, 50 y 10 que operan con el *NR_S*, *NR_A*, *NR_I* corresponden a la valoración dada al número de observaciones con evaluación Satisfactoria, Aceptable e Insatisfactorio respectivamente. Para esta métrica se acepta como valores de conductas adecuadas aquellas que están por arriba de 50%.

- ✓ **Productividad:** se define como la manera acertada en que se han combinado y utilizado los recursos para cumplir con los objetivos específicos deseados, en el tiempo programado³⁴. Esta medición permite medir el cumplimiento, aplicación o evidencia de los equipos en su trabajo y en el producto obtenido en la actividad a través de las características definidas en el instrumento de observación (Evaluación de Conformidad del Investigador). Se obtiene a partir de la siguiente formula (2):

$$P = \left\{ \frac{1}{20} [(NR_5 * 100) + (NR_4 * 80) + (NR_3 * 60) + (NR_2 * 40) + (NR_1 * 20)] \right\} / 100 \quad (2)$$

Donde *P* corresponde a la productividad, *NR₅* es el número de respuestas con valoración 5, *NR₄* es el número de respuestas con valoración 4, *NR₃* es el número de respuestas con valoración 3, *NR₂* es el número de respuestas con valoración 2, *NR₁* es el número de respuestas con valoración 1. El valor 100, 80, 60, 40 y 20 que operan con el *NR₅*, *NR₄*, *NR₃*, *NR₂* y *NR₁*, corresponden a la valoración dada al número de observaciones con evaluación en 1, 2, 3, 4 y 5 respectivamente. Para esta métrica se acepta como valores de conductas adecuadas aquellas que están por arriba de 50%.

- ✓ **Calidad:** se define como la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor³⁵. Esta medición permite medir la condición, estado o presentación del trabajo mediante el instrumento de Observación (Registro de Defectos del Artefacto Construido) teniendo en cuenta las cualidades más representativas del producto como tal durante y después de la actividad. Se obtiene a partir de la siguiente formula (3):

$$Cal = \left[\frac{(NR_{SI} + NR_{NO}) - NR_{NO}}{6} \right] \times 100 \quad (3)$$

³³ <http://definicion.de/comportamiento/>

³⁴ <http://tesis.uson.mx/digital/tesis/docs/7268/Capitulo1.pdf>

³⁵ **RAE:** Real Academia de la Lengua

Donde Cal corresponde a Calidad, NR_S es el número de respuestas afirmativas, NR_{NO} es el número de respuestas negativas. El número 6 corresponde al número de características del instrumento de calidad evaluadas a los equipos. Para esta métrica se acepta como valores de conductas adecuadas aquellas que están por arriba de 50%.

- ✓ **Frecuencia de Formas de Organización y Dinámicas (FOD):** se define como el conjunto de formas de organización y las dinámicas expuestas por los equipos. Esta medición permite medir la frecuencia de aceptación de los equipos en cuanto a comportamiento, productividad y calidad. El instrumento empleado es el chequeo de los datos teniendo en cuenta los resultados obtenidos después de la actividad. Se obtiene a partir de la siguiente fórmula (4):

$$FOD = \frac{C+P+Cal}{3} \quad (4)$$

Donde FOD corresponde a las formas de organización y dinámicas, C es el valor promedio de comportamiento obtenido por cada equipo, P es el valor promedio de la productividad obtenido por el equipo, Cal es el valor promedio de la calidad obtenido por el equipo. El número 3 corresponde al total de métricas utilizadas en esta fórmula. Para esta métrica se aceptable se tendrán en cuenta valores superiores a 50%.

4.3.1.6 Ejecución del Estudio de Caso

Este primer estudio de caso se ejecutó en el periodo comprendido entre el 2 de abril y el 15 de junio de 2012. La actividad se realizó durante 5 sesiones, cada sesión con un curso correspondiente a los grados cuarto y quinto de básica primaria. La sesión contó con una duración de una hora y media donde inicialmente se realizó una introducción de cómo se iba a trabajar y en qué consistía la actividad en general. Por la naturaleza del estudio se seleccionaron los equipos de trabajo de forma aleatoria, para dar la oportunidad a los niños de compartir y relacionarse con algunos compañeros con los que quizá no tuvieran mucho contacto.

Una vez los equipos se conformaron se presentó la actividad, la cual se constituyó como un trabajo recreativo donde los niños de forma libre y espontánea en sus equipos de trabajo construyeron una torre de forma poliédrica. Uno de los objetivos de la actividad fue permitirles a los niños tomar confianza con su equipo de trabajo y a la vez adquirieran compromiso con sus compañeros frente a las tareas a realizar y los materiales a utilizar.

La descripción de la actividad realizada en este estudio de caso se puede ver más en detalle en el Anexo G.

4.3.1.7 Resultados Cuantitativos

Mediciones Directas y Análisis de Resultados

Comportamiento:

En esta actividad los sujetos investigados, participaron construyendo la figura poliédrica propuesta, basándose en la muestra entregada y la guía provista para el trabajo.

En la figura 33 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos.

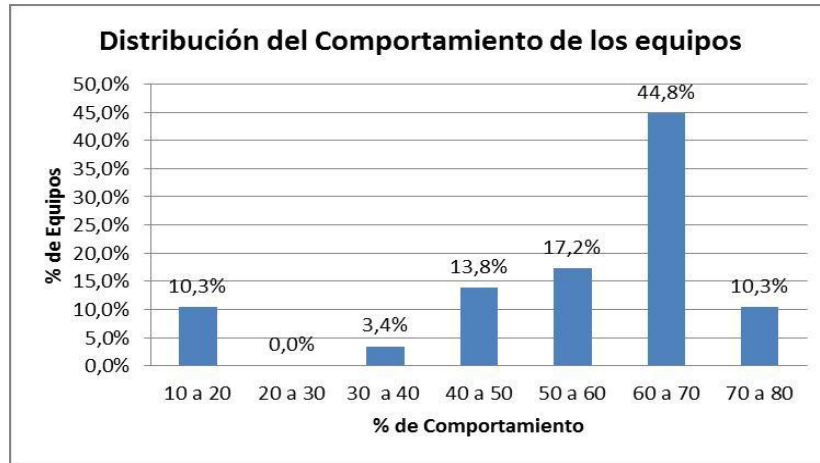


Figura 33. Gráfico de Distribución de Comportamiento por Equipos

El total de equipos observados es de 29 equivalente al 100% de la población y teniendo en cuenta el registro de los datos tomados para la medición de comportamiento Ver Anexo J y cotejando la fórmula (1) definida, se observa en la figura 33 que la mayor parte de los equipos (44,8%) reflejaron con el desarrollo de la actividad actitudes definidas en el instrumento con frecuencias entre 60% y 70%. Según esta grafica nos muestra que la mayor parte de los equipos tiende a tomar comportamientos y actitudes comunes frente a tareas de este tipo, dado que ejecutan acciones que son característicos en ellos en este rango de edad.

Productividad:

En la figura 34 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

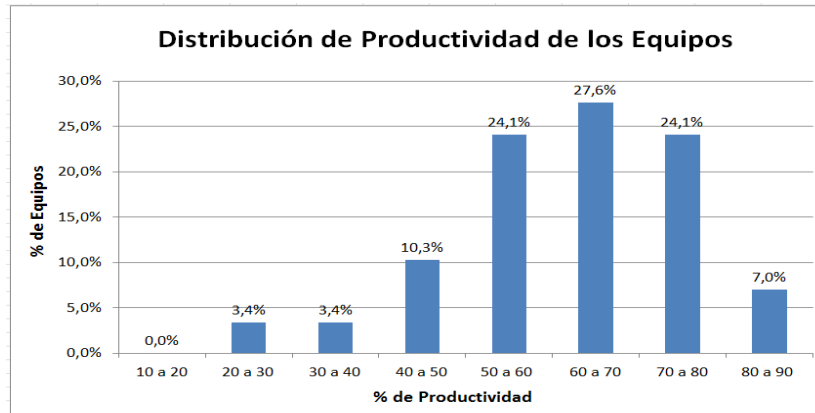


Figura 34. Gráfico de Distribución de Productividad de los Equipos

En la figura 34 se observa que un 71,8% de los equipos logran a través de las entregas del producto (estructura) reflejar características definidas en el instrumento para esta medición con frecuencias entre el 50% y 80%. Hay una mayor tendencia de los equipos a mostrar estos aspectos porque lograron aprovechar los recursos entregados y aplicar una secuencia de pasos básica pero definida para la el desarrollo de la actividad. Los resultados donde se observa pocos equipos con baja frecuencia de características productivas se debe a que hay presencia de equipos que presentaban problemas de comunicación, y de entendimiento entre los miembros que afecto en la evidencia productiva (estructura final). Así mismo, hay pocos equipos con una alta frecuencia (en porcentaje) de productividad, debido a que presentaban cualidades más comunes de trabajo que les ayudo a construir adecuadamente la estructura.

Calidad:

En la figura 35 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

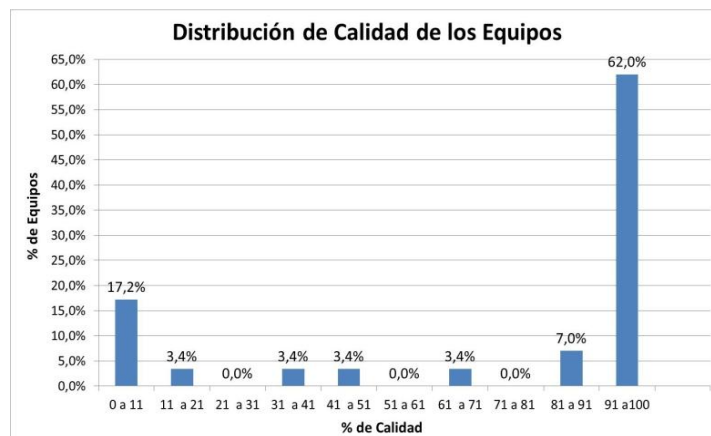


Figura 35. Gráfico de Distribución de Calidad de los Equipos

En la figura 35 se observa que el 62% de los equipos mostraron características iguales o cercanas a las definidas en el instrumento de observación para la medición de la calidad, con una frecuencia de entre el 91% y el 100%. Lo anterior indica que la gran mayoría de estos equipos no solo construyeron la estructura para cumplir con el objetivo, sino que además presentaron características que representaban un producto con calidad en cuanto a su forma, aspecto, modo de trabajo, entre otras. Así como se observa que hay equipos que sobresalen, hay otros que no, por ejemplo, el 17,2% de los equipos entregaron estructuras con características que no se acercaban a las definidas en el instrumento y por ende con poca frecuencia de evidencia en ellas. En general puede observarse que los equipos de trabajo en su mayoría asimilaron las condiciones y lograron cumplir con calidad el requerimiento de la construcción de la estructura.

Frecuencia de Formas de Organización y Dinámicas:

En la figura 36 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

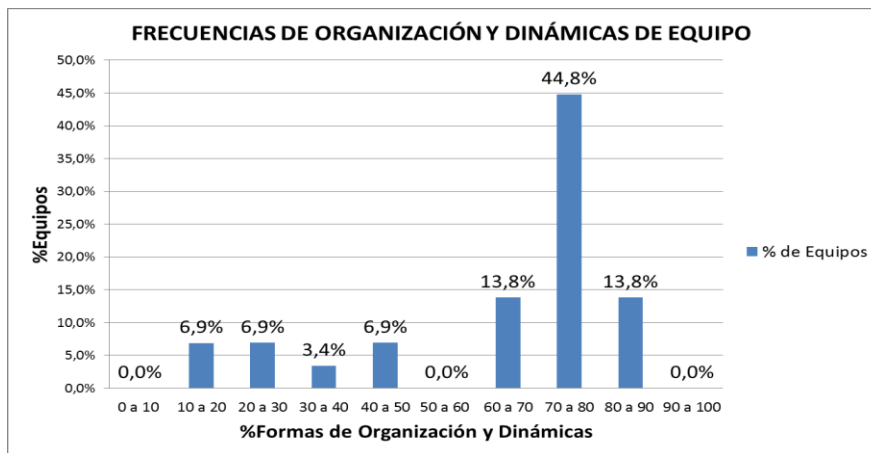


Figura 36. Frecuencias de Organización y Dinámicas de Equipo

En la figura 36 se observa que en aspectos de organización y dinámicas de equipo evidenciadas el 44,8% de ellos tuvieron más frecuencia de características definidas en el instrumento para esta medición entre el 70% y 80% respectivamente, lo cual indica que para estos casos los equipos presentaron características más comunes para organizarse y manejar la actividad en equipo a diferencia de otros pocos que evidenciaron con menor frecuencia los aspectos definidos en el instrumento de observación. Hay más tendencia de los equipos a responder de forma común a este tipo de actividad mostrando una mayor frecuencia debido a que es más cercana a su entorno cotidiano, por tener componentes de juego y creatividad.

Dado que para este estudio se tenían como base un conjunto de prácticas teóricas (P_0), la experiencia después de la observación dio como resultado de este estudio un subconjunto de prácticas frecuentes en los equipos de trabajo, las cuales fueron recopiladas en un nuevo conjunto (P_1) quienes son inicial y potencialmente las “Prácticas Child”.

$$P_1 \subseteq P_0 \quad (5)$$

Donde P_1 corresponde al conjunto de las prácticas recopiladas, P_0 denota el conjunto de prácticas teóricas de la cual se extrae un subconjunto de ellas.

En la tabla 9 se presentan las 10 prácticas se recopilan las prácticas evidenciadas en los equipos con mejor desempeño para este estudio.

No. Práctica	Prácticas Evidenciadas
1	Entienden Conceptos
2	Buscan explicaciones lógicas
3	Siguen Ordenes
4	Aceptan y respetan reglas
5	Establecen canales de comunicación
6	Voluntad para el Trabajo en Equipo
7	Entregas Pequeñas
8	Diseño Simple e Incremental
9	Entregable del Producto
10	Compromiso del Equipo

Tabla 9. Conjunto de “Prácticas Empíricas Child”

Teniendo en cuenta los datos arrojados por comportamiento, productividad y calidad se realizó un conteo a través del instrumento de aceptación de métricas por equipo en el que se tuvieron en cuenta los valores que superaran el 50% aceptado los cuales son considerados buenos para este estudio, ver Anexo J. En la tabla 10 se muestran estos resultados.

Frecuencias con alto grado	Resultados Generales
No. de Equipos con alto grado de Comportamiento	21
No. de Equipos con alto grado Productividad	24
No. de Equipos con alto grado de Calidad	23

Tabla 10. Registro de Frecuencias

4.3.1.8 Resultados Cualitativos

4.3.1.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso

Durante la exploración observamos que los equipos de trabajo encontraron formas diferentes de desarrollar la actividad partiendo de sus experiencias, es así, como en algunos equipos fue evidente ver como los niños relacionaban conceptos de geometría en la construcción de la torre propuesta, comparando figuras y formas, la aritmética básica para hacer conteos sencillos y la literatura y la escritura la cual se reflejó en la comprensión de la actividad y el desarrollo de la misma. La interacción entre los integrantes de los equipos también, reflejo un comportamiento de aceptación y asimilación de las condiciones logrando la socialización entre géneros, edades y personalidades de cada niño. En cuanto a la actividad, se mostró interés y motivación por parte de los niños, dado que tomaron como un juego el desarrollo de la actividad lo cual les permitió asimilar y acoplarse rápido al desarrollo.

A partir de las observaciones realizadas en la actividad, las cuales fueron registradas en videos y fotografías, se extrajeron varios resultados cualitativos que se mencionan a continuación. Ver Anexo Digital.



Figura 37. Imágenes de diferentes Equipos en el Desarrollo de la Actividad 1.

4.3.1.9 Amenazas de Validez

A continuación se resume un cierto número de factores que ponen en peligro la validez del estudio:

- Dadas las condiciones de la institución educativa, se presentaron dificultades en cuanto al lugar de trabajo por la presencia de otras actividades previstas con anterioridad dentro del calendario académico.
- Se presentaron diferencias en este aspecto debido que los integrantes de los equipos tenían una organización definida teniendo en cuenta sus afinidades, sus gustos y complementos lo cual interfirió un poco al inicio de la actividad.

- En este aspecto las diferencias en los resultados se dieron por la falta de fiabilidad de los instrumentos de medida que no fueron diseñados en su totalidad para la unidad de análisis definida, ocasionando algunas malos entendidos con las preguntas, además, se evidenció subjetividad en uno de los instrumentos lo cual no permitió obtener información suficientemente verídica. Estos resultados fueron eliminados del estudio.
- Se evidenció en algunos casos, equipos de trabajo que recurrían a observar el trabajo de otros equipos con el objetivo de copiar su estrategia y ponerla en práctica en su grupo para mejorar sus resultados.
- La explicación inicial de la actividad pudo haber causado efectos de sensibilidad a la población en cuanto al desarrollo de la misma.
- Adicionalmente fue evidente la falta de espacios adecuados, algunos muy reducidos o con mucho ruido lo que impedía un poco el desarrollo normal de la actividad ocasionando algunas dificultades para la observación.

4.3.2 Estudio de CASO 2

4.3.2.1 Pregunta de Investigación

Para conocer cómo los niños en sus equipos de trabajo adoptan y aplican las prácticas ágiles, colaborativas y cognitivas evidenciadas en la realización de una actividad colaborativa, este segundo estudio de caso pretende resolver la siguiente pregunta:

¿Cómo la adopción de prácticas ágiles de desarrollo y de técnicas de aprendizaje colaborativo incrementan la capacidad del trabajo en equipo y su productividad en el área de informática con equipos de niños entre 8 y 10 años de edad de los grados cuarto y quinto de básica primaria de la institución educativa técnico industrial sede san camilo?

4.3.2.2 Objetivo del Estudio

Evidenciar el impacto de las prácticas de colaboración, cognición y de agilidad en equipos de (5±2) niños con edades comprendidas entre los 8 y 10 años y seleccionar las que finalmente van a ser incorporadas al modelo Child Programming.

4.3.2.3 Selección del Estudio

La unidad de análisis para este caso siguen siendo los equipos de trabajo. El estudio de caso 2 es de tipo **Embebido** [64], considerando las 29 unidades de análisis con los mismos 177 sujetos de investigación. Para este caso se dividieron las 29 unidades de análisis en dos grupos, un grupo experimental con 19 unidades y un grupo de control de

10 unidades. Las unidades de análisis seleccionadas continúan siendo de tipo **Aleatoria** en su conformación y de tipo **Revelatorio** en su resultado, lo que lo hace suficientemente completo para evaluar las características colaborativas y ágiles dentro de sus equipos de trabajo.

4.3.2.4 El Estudio de Caso y los Sujetos de Investigación

Este estudio de caso es de tipo embebido [64] al igual que el estudio de caso 1, donde las unidades de análisis siguen siendo los 29 equipos de trabajo conformados por niños con edades comprendidas entre los 8 y 10 años, de los grados cuarto y quinto de básica primaria, quienes para efectos de este caso se constituyen como las fuentes primarias de información, en total 177. Este estudio es exploratorio el cual pretende evidenciar el impacto de las prácticas seleccionadas en los equipos de trabajo.

El diseño de este estudio de caso conto con 5 sesiones de trabajo una sesión por cada curso como se definió en el estudio de caso 1. A partir de las prácticas extraídas del estudio de caso 1, se aplicó una actividad que involucraba los dos grupos seleccionados (experimentales y control).

Es de resaltar que todos los sujetos de investigación en este estudio de caso igual que en el estudio de caso 1, contaron con las mismas condiciones de trabajo en cuanto al espacio, los materiales y el desarrollo de la actividad; con la diferencia de que los grupos seleccionados como de control no contaron con la información correspondiente a las prácticas que deberían aplicarse para el desarrollo de la misma.

4.3.2.5 Indicadores y Mediciones

Para obtener la información necesaria para éste Estudio de Caso y dar respuesta a la pregunta de investigación fue necesario emplear el conjunto de mediciones definidos en el estudio de caso 1 adicionando una nueva medición, "*Nivel de Adopción de Prácticas*". La tabla 11 muestra el resumen de los indicadores y mediciones definidos.

PREGUNTA DE INVESTIGACIÓN	INDICADOR	MEDICIONES	INSTRUMENTO
<i>¿Cómo la adopción de prácticas ágiles de desarrollo y de técnicas de aprendizaje colaborativo incrementan la capacidad del trabajo en equipo y su productividad en el área de informática con equipos de niños entre 8 y 10 años de edad de los grados cuarto y quinto de básica</i>	Presencia e Impacto de Prácticas cognitivas, colaborativas, cognitivas y ágiles en equipos de	Nivel de Comportamiento observado en los equipos de trabajo	Observación, Protocolo de Observación
		Nivel de Productividad Alcanzada por los equipos de trabajo	Observación, Evaluación de Conformidad del Investigador
		Nivel de Calidad con respecto al producto	Observación, Registro de Defectos

primaria de la institución educativa técnico industrial sede san camilo?	trabajo	entregado por los equipos de trabajo	del Artefacto Construido
		Nivel de Adopción de Prácticas expuestas por los equipos de trabajo en el desarrollo de la actividad.	Observación, Adopción de Prácticas Equipos de Trabajo

Tabla 11. Indicadores y Mediciones Estudio de Caso 2

Como las tres primeras mediciones ya están descritas en el Estudio de Caso 1, a continuación se describe la medición “Nivel de Adopción de Prácticas” incluida en este estudio.

- ✓ **Nivel de Adopción de Prácticas:** las prácticas corresponden a los métodos o programas que han demostrado ser exitosos en alcanzar sus metas y que pueden ser usados, o adaptados para su uso, en sus circunstancias propias³⁶. Permite medir la adopción de prácticas en los equipos de trabajo para desarrollar la actividad propuesta. La fórmula (6) que hemos definido para hacer el cálculo de la adopción de prácticas está dada por:

$$Adp = \frac{1}{10}[(NR_S * 100) + (NR_A * 50) + (NR_I * 10)]/100 \quad (6)$$

Donde *Adp* corresponde a Adopción de Prácticas, NR_S es el número de respuestas satisfactorias, NR_A es el número de respuestas aceptables, NR_I es el número de respuestas insatisfactorias. El factor (1/10) se relaciona con el número de características contenidas en el instrumento de observación (Adopción de prácticas de equipos de trabajo). El valor 100, 50 y 10 que operan con el NR_S , NR_A , NR_I corresponden a la valoración dada al número de observaciones con evaluación Satisfactoria, Aceptable e Insatisfactorio respectivamente. Para esta métrica se aceptable se tendrán en cuenta valores superiores a 50%.

4.3.2.6 Ejecución del Estudio

Este segundo estudio de caso se puso en marcha en el periodo comprendido entre el 20 de agosto y el 19 de octubre de 2012. Para este estudio se diseñó una actividad colaborativa descrita más adelante. La actividad se realizó igualmente durante 5 sesiones, cada sesión con un curso, la cual contó con una duración de una hora y media.

Teniendo en cuenta los resultados arrojados por el estudio de caso 1 (antes descrito), se seleccionaron 10 equipos de trabajo como equipos de control para darle a esta investigación un factor de comparación que permita evaluar la incidencia del modelo propuesto. La selección de estos equipos se dio de la siguiente forma:

³⁶ <http://ctb.ku.edu/es/tablecontents/seccionprincipal19.6.aspx>

- Por cada curso se seleccionaron 2 equipos, uno de ellos con bajo desempeño y otro con mejor desempeño, evidenciado en la actividad 1, con el propósito de dar un balance equitativo entre el total de equipos del curso al tomar los extremos.
- Los equipos de control contaron con la información de la actividad, sin embargo no tuvieron el soporte metodológico de prácticas ágiles, colaborativas y cognitivas aplicables a la actividad que fue entregado a los equipos experimentales (19 equipos).

La descripción de la actividad definida para este estudio se puede ver con más detalle en el Anexo G.

4.3.2.7 Resultados Cuantitativos

Mediciones Directas y Análisis de Resultados

Comportamiento:

En esta actividad los sujetos investigados, participaron aplicando la técnica JIGSAW para analizar el tema algorítmico y elaborar un diagrama de flujo según la tarea especificada.

En la figura 38 se muestra el resultado gráfico obtenido de esta valoración, agrupada en intervalos de 10 valores medidos en porcentajes por equipos para los equipos experimentales y de control, ver Anexo J.

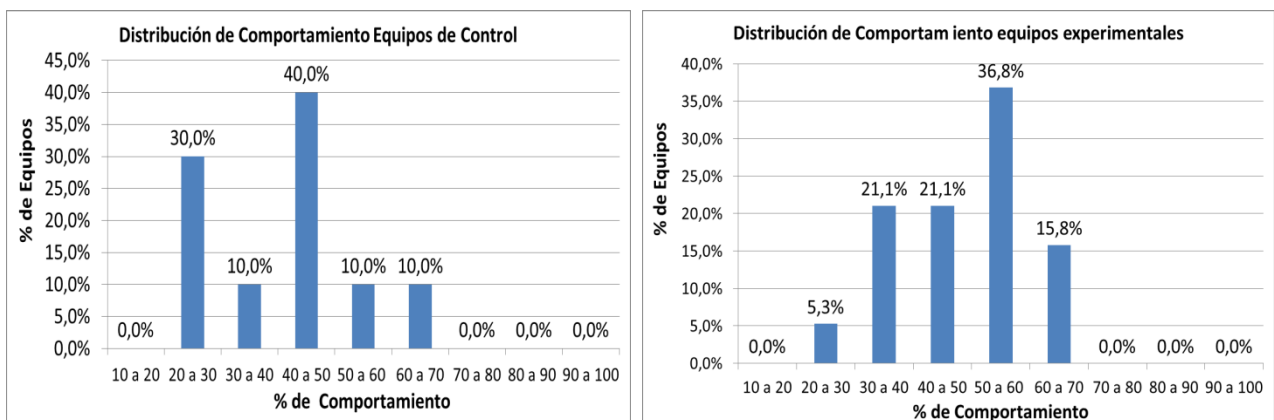


Figura 38. Distribución de Comportamiento Equipos Experimentales y de Control

Para este estudio los 29 equipos que se tenían inicialmente fueron divididos en 2 grupos: Grupos experimentales (19) y Grupos de Control (10). Nos centramos específicamente en

los equipos experimentales que son quienes reciben las mejores prácticas que deben aplicar en el desarrollo de la actividad como apoyo de trabajo. Aquí se observa que la población de equipos experimentales está más desplazado a la derecha con respecto a los de control, lo que indica que evidencian con un poco más de frecuencia estas características definidas en el instrumento acorde para esta medición. Puntualizamos señalando que el 36.8% de los equipos experimentales presentaron una frecuencia entre 50% y 60% en lo evidencia de estos aspectos del instrumento de observación para esta medición, lo cual indica que estos tienen comportamientos y/o actitudes comunes para desarrollar esta actividad que les permitió actuar de forma un poco más acertada para cumplir con la meta. Los equipos de control solo nos permiten contrastar los resultados de inducir las prácticas y su reflejo en este aspecto a través del desarrollo de la actividad

Productividad:

En la figura 39 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

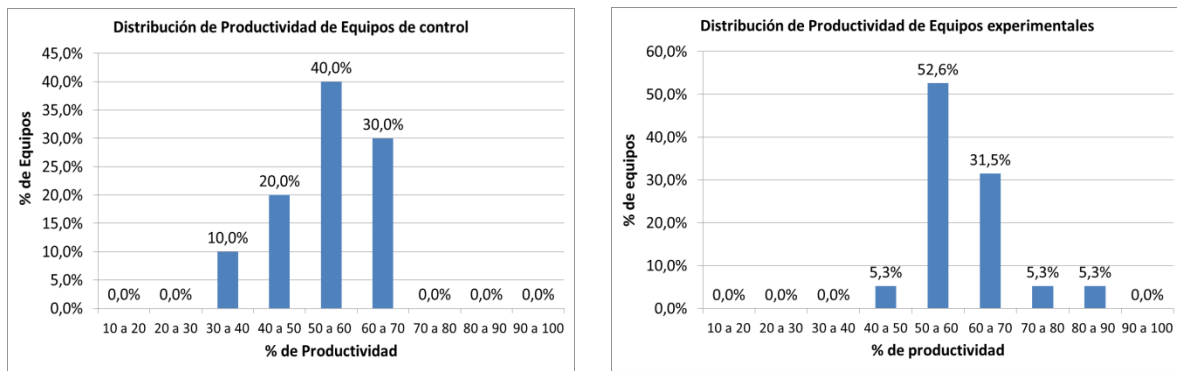


Figura 39. Gráfico de Distribución de Productividad de Equipos Experimentales

En la figura 39 se observa que la población de equipos experimentales tiende más a la derecha con respecto a los de control, es decir hay más concentración de equipos que evidencian más frecuencia (en %) de características definidas en el instrumento de observación para esta medición, lo cual indica que aprovecharon las practicas entregadas para soportar el aspecto productivo manifestado a través de los procedimientos, acciones y características del trabajo presentado al final de la actividad. De esta manera observamos que se concentró el 52.6% de los equipos experimentales que presentaron una frecuencia de características formalizadas entre el 50% y 60%. Por otro lado los equipos de control concentran su población por debajo de los resultados de los experimentales lo cual refleja que hubo una diferencia significativa en cuanto al aprovechamiento de los prácticas para apropiarse de los temas, los recursos y la interacción con los integrantes del equipo para presentar de manera conforme el trabajo donde reflejara el conocimiento adquirido.

Calidad:

En la figura 40 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

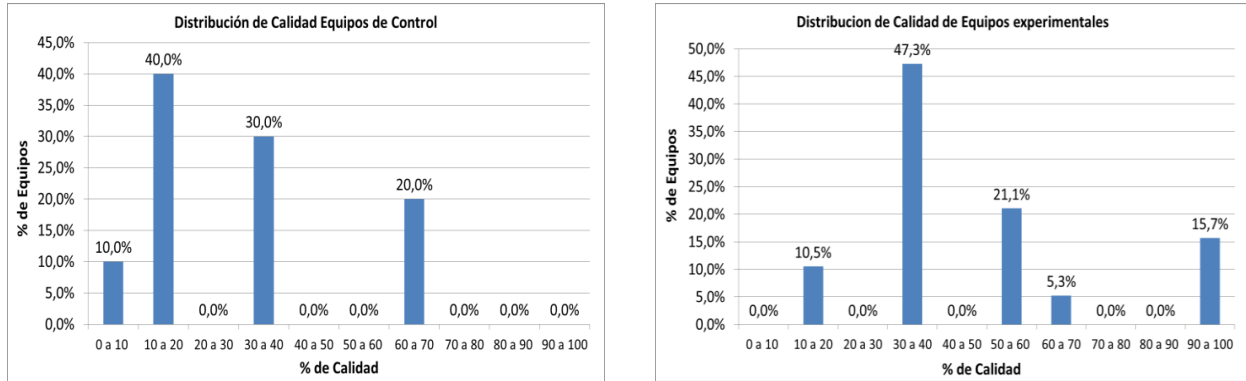


Figura 40. Gráfico de Distribución de Calidad de Equipos Experimentales

En la figura 40 se observa que los equipos experimentales muestran menor frecuencia de características definidas en el instrumento de observación para esta medición, lo que indica que a pesar de que presentaron la tarea propuesta para esta actividad, no evidenciaron un mejor acercamiento a las cualidades esperadas. Así por ejemplo a pesar de que el 15.7% de los equipos experimentales mostraron características en su trabajo con frecuencia entre el 90% y aproximadamente el 100%, la mayor parte de equipos es del 47.3% con poca frecuencia de aspectos que se esperaban se vieran reflejados en las entregas. Sin embargo a diferencia de los de control, los equipos experimentales tienden a tener mejor resultado en cuanto a este aspecto, aprovechando las pocas practica que lograron asimilar para cumplir con la entrega y dado que esta actividad implicaba un tema nuevo para ellos es un factor a tener en cuenta.

Adopción de Prácticas:

En la figura 41 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos para los equipos experimentales y de control.

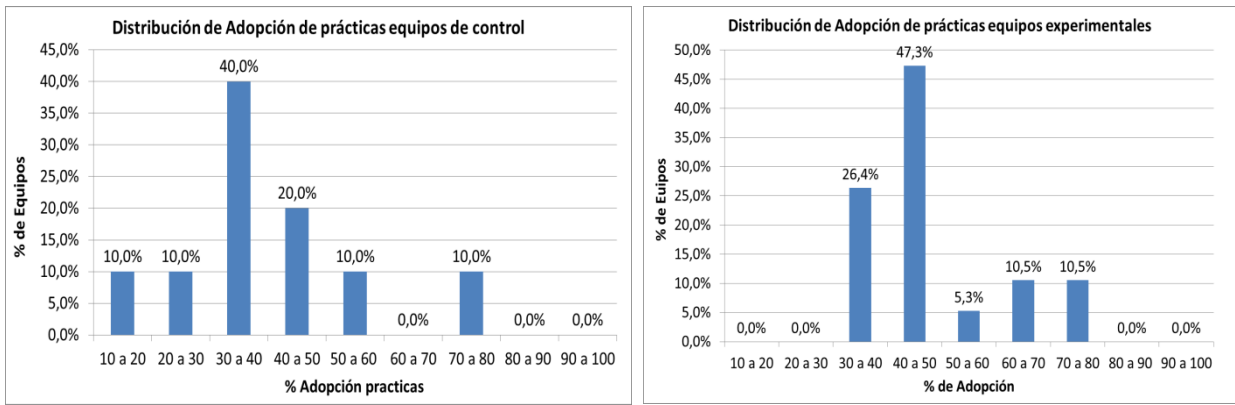


Figura 41. Distribución de Adopción de Prácticas Equipos Experimentales y de Control

En la figura 41 se observa que de los equipos experimentales el 47.3% obtuvieron una frecuencia de adopción de prácticas entre el 40% y el 50%, que representa un resultado bajo para lo que se esperaba, en principio se puede pensar que es debido a que para los niños es una manera nueva de trabajar y se empiezan a acoplar con esta metodología, sin embargo para ser la primera vez que lo hacen estar entre ese porcentaje no es tan negativo como se considera. Una muestra de que hubo equipos que lograron adaptar estas prácticas en su trabajo se refleja en los intervalos 6 y 7, donde el 21% en total de equipos que están entre estos dos intervalos registran porcentajes entre el 60% y 80% de frecuencia en la aplicación de estas prácticas

Para contrastar estos resultados se procedió a realizar un estudio comparativo con respecto a la adopción de prácticas con el fin de evidenciar la presencia de estas en los equipos de trabajo.

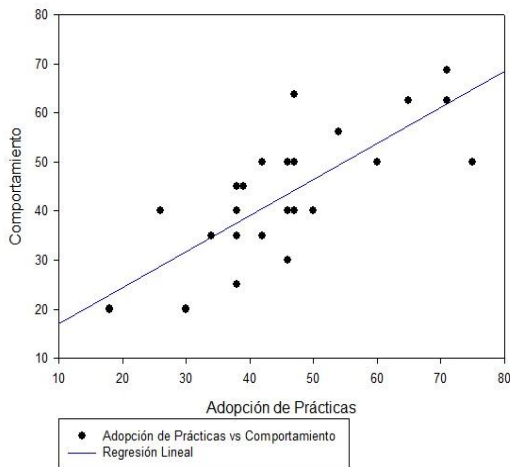


Figura 42. Gráfico de Dispersión Adopción Vs Comportamiento

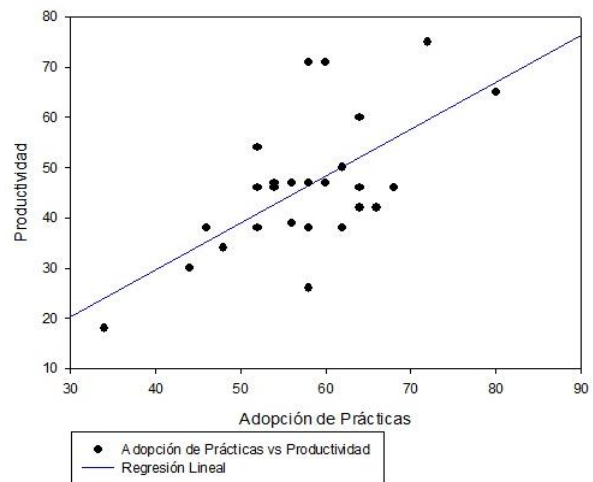


Figura 43. Gráfico de Dispersión Adopción Vs Productividad

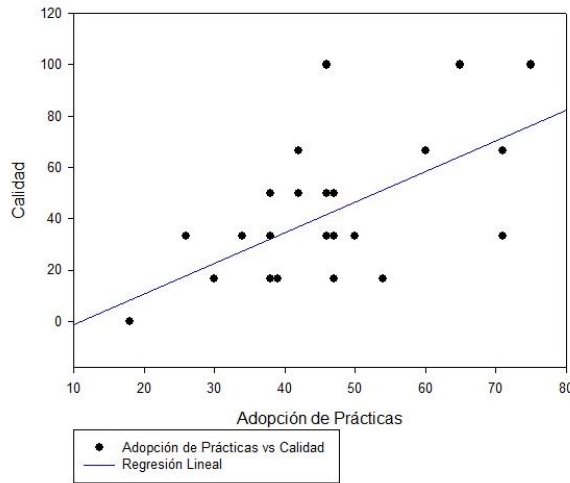


Figura 44. Gráfico de Dispersión Adopción Vs Calidad

En las figuras 42, 43 y 44 se puede observar en general un comportamiento disperso de los equipos de trabajo (control y experimentales), particularmente en la figura 50 se observa que los datos se encuentran mucho más dispersos con respecto a la figura 51 que indica la productividad de los equipos, dado que existe una mejor estimación en los resultados obtenidos por los equipos, que aunque las unidades generadas fueron aceptables, la ausencia de calidad es notoria por la presencia de defectos en la construcción del algoritmo y la falta de comprensión de algunos conceptos.

Este estudio da como resultado un nuevo conjunto de prácticas (P_2), que se derivan de la aplicación del conjunto de prácticas (P_1) descrito en el estudio de caso 1 y la diferencia entre un compendio de nuevas prácticas teóricas (P_0) y las prácticas (P_1) antes mencionadas. La aplicación de la fórmula (7) permite extraer el conjunto final de prácticas que harán parte de *ChildProgramming*.

$$P_2 = P_1 \cup (P_0 - P_1) \quad (7)$$

Donde P_2 hace referencia al nuevo conjunto de prácticas observadas, P_1 corresponde al conjunto de las prácticas recopiladas, P_0 denota el conjunto de prácticas teóricas del cual se le restan las prácticas recopiladas inicialmente dando como resultado un compendio de nuevas prácticas.

En la tabla 12 se observa el conjunto de prácticas y las operaciones realizadas para seleccionar los conjuntos P_1 y P_2 . De allí que salen las 14 prácticas nuevas observadas y analizadas para este estudio.

Prácticas P0 Teóricas	Estudio de Caso 1	P0 - P1	Estudio de Caso 2
	Prácticas P1		P2 = P1 ∪ (P0 - P1)
Trabajo por Pares		Trabajo por Pares	Trabajo por Pares
Diseño Simple e Incremental	Diseño Simple e Incremental		Diseño Simple e Incremental
Voluntad para el Trabajo en Equipo	Voluntad para el Trabajo en Equipo		Voluntad para el Trabajo en Equipo
Entregas Pequeñas	Entregas Pequeñas		Entregas Pequeñas
Propiedad Colectiva de Trabajo	NO OBSERVADAS		
Equipo Auto-Gestionado			

Refactoring o Mejora		Refactoring o Mejora	Refactoring o Mejora
Pruebas		Pruebas	Pruebas
Integración Continua		Integración Continua	Integración Continua
Estandarización		Estandarización	Estandarización
Espacio Informativo		Espacio Informativo	Espacio Informativo
Transparencia	NO OBSERVADA		
Marco de Trabajo		Marco de Trabajo	Marco de Trabajo
Entregable del Producto	Entregable del Producto		Entregable del Producto
Compromiso con el Equipo	Compromiso con el Equipo		Compromiso con el Equipo
Aceptan y Respetan Reglas	Aceptan y Respetan Reglas		Aceptan y Respetan Reglas
Análisis de Rendimiento	NO OBSERVADA		
Trabajo en Equipo		Trabajo en Equipo	Trabajo en Equipo
Cooperación para Cumplir Tareas Establecidas		Cooperación para Cumplir Tareas Establecidas	Cooperación para Cumplir Tareas Establecidas
Establecen Canales de Comunicación	Establecen Canales de Comunicación		Establecen Canales de Comunicación
Desarrollo de Estrategias para Cumplir el Objetivo		Desarrollo de Estrategias para Cumplir el Objetivo	Desarrollo de Estrategias para Cumplir el Objetivo
Análisis de Rendimiento	NO OBSERVADA		
Participación Continua		Participación Continua	Participación Continua
Entienden Conceptos	Entienden Conceptos		Entienden Conceptos
Buscan Explicaciones Lógicas	Buscan Explicaciones Lógicas		Buscan Explicaciones Lógicas
Siguen Ordenes	Siguen Ordenes		Siguen Ordenes
Leer Instrucciones y Explicar los Pasos a Seguir		Leer Instrucciones y Explicar los Pasos a Seguir	Leer Instrucciones y Explicar los Pasos a Seguir
Análisis de Rendimiento	NO OBSERVADA		
Comprender lo que se lee		Comprender lo que se lee	Comprender lo que se lee
Expresar Dudas y Comentarios		Expresar Dudas y Comentarios	Expresar Dudas y Comentarios
Usan Símbolos y guías	NO OBSERVADAS		
Leen Instrucciones y Explican los pasos a Seguir			

Tabla 12. Conjunto de Prácticas Observadas en el Estudio

No. Práctica	Prácticas Evidenciadas
1	Trabajo por Pares
2	Refactoring o Mejora
3	Pruebas
4	Integración Continua
5	Estandarización
6	Espacio Informativo
7	Marco de Trabajo
8	Trabajo en Equipo
9	Cooperación para Cumplir Tareas Establecidas
10	Desarrollo de Estrategias para Cumplir el Objetivo
11	Participación Continua
12	Leen Instrucciones y Explican los Pasos a Seguir
13	Comprenden lo que Leen
14	Expresan Dudas y Comentarios

Tabla 13. Conjunto de Nuevas Prácticas Potenciales "Prácticas Child"

4.3.2.8 Resultados Cualitativos

El análisis cualitativo se realizó teniendo en cuenta las observaciones realizadas a nivel general a las unidades de análisis, en total 29 equipos de trabajo divididos en 5 sesiones.

En la tabla 14 se presenta un análisis cualitativo teniendo en cuenta las observaciones realizadas en este estudio de caso.

Equipos de Control	Equipos Experimentales
Se mostraron un tanto desorientados durante la ejecución de la actividad en cuanto a la manera de proceder, de organización y de comprensión de las tareas	Se mostraron atentos a las indicaciones de la actividad y de las prácticas que se les daba para aplicarlas. Por ellos no se mostró desorientación cuando realizaban las tareas.
Por la no presencia de roles se presentaba desorden, pues todos se involucraban en la misma tarea y no había distribución de trabajo.	Había presencia de algunos roles que les permitió organizarse mejor en cuanto a la asignación de tareas.
Falta de comunicación entre los integrantes, les generó discusiones que no le permitía tomar buenas decisiones para llegar a la solución.	Mantuvieron una buena comunicación entre los integrantes que les ayudó a entender entre todos el problema a resolver.
Había ideas en el grupo pero fallaban para poder ejecutar la mejor de ellas en la actividad por cuestiones de indecisión en escoger la mejor.	Todos aportaban ideas para solucionar el problema, de tal manera que había más opciones para escoger. Eran más seguros para tomar las decisiones y aplicarlas en la actividad.
Falta de estrategias que les permitiera rendir en la actividad, nadie se hacía cargo de apuntarlas para ponerlas en discusión.	Las estrategias planteadas le ayudó a mejorar en el planteamiento de la solución porque
La socialización entre los integrantes de los equipos era menor porque no había tanta voluntad para trabajar en algunos integrantes.	Dado que conocían la práctica de compromiso del equipo, trataron de mejorar las relaciones y socializaban más para armar parejas de trabajo internamente.
Generaban poco o nada en cuanto a la solución, se dedicaban a hacer otras cosas ajenas a la actividad.	Realizaban pequeñas muestras de lo que iban haciendo en la actividad, lo cual les permitía retroalimentarse y mejorar la solución.
Muy poco comprometidos en la actividad porque no había voluntad individual de algunos integrantes de los equipos.	Había más compromiso de trabajo por parte de los equipos porque conocían mejor el objetivo de la actividad.

Tabla 14. Análisis Cualitativo Equipos Experimentales y Equipos de Control

4.3.2.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso

En este estudio los dos grupos seleccionados para observar (experimentales y de control) mostraron características diferentes en cuanto a comportamiento, productividad y la calidad del desarrollo de la actividad. Los equipos de trabajo experimentales lograron apropiarse de la actividad en la medida que ellos iban aplicando el conjunto de prácticas involucradas, esta aplicación brindó a los equipos la confianza necesaria para desempeñarse de la forma más adecuada y con las herramientas más útiles. A diferencia

de los equipos experimentales, los equipos de control presentaron comportamientos en algunos casos inadecuados, generando desorden en el espacio, falta de interés, desmotivación y falta de compromiso; la razón fue atribuida a que los equipos experimentales contaron con el conjunto completo de prácticas que podrían emplear para el desarrollo de esta actividad. Como se trató de una actividad donde se aplicó una de las técnicas de aprendizaje colaborativo (JIGSAW), los equipos de trabajo en la “fase de expertos” se unificaron de tal forma que compartieran el mismo conocimiento, en este momento los equipos de control trataron de acoplarse activamente al grupo de expertos aportando ideas y apropiándose de las ideas de los demás, pero una vez retornaron a sus equipos originales fue evidente el cambio de actitud frente a la explicación del tema a los demás compañeros.

Estas manifestaciones por parte de los equipos se vieron reflejadas en la calidad de los trabajos entregados por los equipos, lo cual fue notable en sus valoraciones.

A partir de las observaciones realizadas en la actividad fue registrado en videos y fotografías el desarrollo de la misma en sus diferentes fases como se observa en las figuras 45 y 46. Ver también Anexo Digital.



Figura 45. Imágenes de Diferentes Equipos en el Desarrollo de la Actividad 2 – Fases 1, 2, 3 de Jigsaw



Figura 46. Imágenes de Diferentes Equipos en el Desarrollo de la Actividad 2 – Fases 4 y 5 de Jigsaw

4.3.2.9 Amenazas de Validez

A continuación se resume un cierto número de factores que ponen en peligro la validez interna del estudio:

- Las condiciones del nuevo espacio asignado para la realización de la actividad (la biblioteca), no era la más adecuada dado que no contaba con las suficientes dimensiones deseadas, además contaba con poca iluminación y ventilación, lo cual incomodaba a los equipos y sus integrantes por el espacio reducido.

- En algunas oportunidades el profesor se acercó a los equipos tratando de dar solución a posibles dificultades en la actividad, en un momento no indicado para hacerlo lo que generó un poco de desorden al interior del espacio.
- Un factor importante como amenaza fue el tiempo disponible para realizar la actividad, dado que el horario asignado no se respetada totalmente por parte de la institución.
- La presencia de ruido, fue notoria en este estudio por la cercanía del espacio asignado a la calle principal. Fue evidente escuchar carros, motos y demás vehículos que generaban distracción a los niños en sus equipos de trabajo
- Otro factor menos frecuente pero no menos importante de mencionar fueron las faltas que algunos niños tenían a la institución los días de aplicación de las actividades, lo que generaba equipos incompletos con más responsabilidad y compromiso en el trabajo.

4.3.3 Estudio de Caso 3

4.3.3.1 Pregunta de Investigación

Para conocer si el modelo propuesto ChildProgramming representa en los niños una guía efectiva de trabajo que involucra prácticas ágiles, colaborativas y cognitivas, este estudio de caso pretende resolver la siguiente pregunta:

¿Es ChildProgramming una guía efectiva de desarrollo de una solución software para un equipo pequeño (5±2) de niños en edades comprendidas entre los 8 y 10 años de los grados cuarto y quinto de básica primaria de la institución educativa técnico industrial sede san camilo?

4.3.3.2 Objetivo del Estudio

Aplicar el modelo ChildProgramming a través de una actividad práctica con equipos pequeños (5±2) niños en edades comprendidas entre los 8 y 10 años de los grados cuarto y quinto de básica primaria, analizando los resultados de su aplicación y comparándolos con los resultados arrojados de este estudio.

4.3.3.3 Selección del Estudio

La unidad de análisis para este caso es el **Modelo ChildProgramming**. El estudio de caso 3 es de tipo **Holístico** [64]. Considerando una única unidad de análisis (proyecto de programación) con 29 sujetos de investigación que corresponden al número de equipos conformados. La unidad de análisis seleccionada es de tipo **Revelatoria** en su resultado, lo que la hace suficientemente para evaluar la aplicabilidad de la propuesta.

4.3.3.4 El Estudio de Caso y los Sujetos de Investigación

Este estudio de caso es de tipo holístico, considerando el modelo ChildProgramming como unidad de análisis. Este estudio de caso mantiene un enfoque confirmatorio dado que pretende evidenciar el resultado positivo de la aplicación del modelo propuesto.

Se diseñó el estudio de caso manteniendo cinco sesiones de trabajo, una sesión por cada curso como se definió en el estudio de caso 1 y 2, con el objeto de observar en cada equipo de trabajo la aplicabilidad del modelo y los resultados generados por el mismo.

4.3.3.5 Indicadores y Mediciones

Para obtener la información necesaria para éste estudio de caso y dar respuesta a la pregunta de investigación fue necesario emplear el conjunto de mediciones definidas en el estudio de caso 1 y 2 adicionando una nueva medición, “*Nivel de Adopción de Nuevas Prácticas*” y un indicador para este estudio. La tabla 15 muestra en resumen el indicador y las mediciones definidas.

PREGUNTA DE INVESTIGACIÓN	INDICADOR	MEDICIONES	INSTRUMENTO
¿Es ChildProgramming una guía efectiva de desarrollo de una solución software para un equipo pequeño (5±2) de niños en edades comprendidas entre los 8 y 10 años de los grados cuarto y quinto de básica primaria de la institución educativa técnico industrial sede san camilo?	Aplicación efectiva de prácticas colaborativas, ágiles y cognitivas extraídas	Nivel de Comportamiento observado en los equipos de trabajo	Observación, Protocolo de Observación
		Nivel de Productividad Alcanzada por los equipos de trabajo	Observación, Evaluación de Conformidad del Investigador
		Nivel de Calidad con respecto al producto entregado por los equipos de trabajo	Observación, Registro de Defectos del Artefacto Construido
		Nivel de Adopción de Prácticas en Equipos de Trabajo	Observación, Adopción de Prácticas
		Nivel de Adopción de Nuevas Prácticas en Equipos de Trabajo	Observación, Adopción de Nuevas Prácticas

Tabla 15. Indicadores y Mediciones Estudio de Caso 3

Como los cuatro de las primeras mediciones ya están descritos en el Estudio de Caso 1 y 2, a continuación se describe el “Nivel de Adopción de Nuevas Prácticas” incluido en este estudio.

- ✓ **Adopción de Nuevas Prácticas:** las prácticas corresponden a los métodos que han demostrado ser exitosos y que pueden ser usados o adaptados para su uso, en sus

circunstancias propias³⁷. La adopción de nuevas prácticas permite adicionar al proceso prácticas adicionales que apoyaran de manera efectiva el desarrollo de la actividad propuesta. Permite medir la adopción de nuevas prácticas en los equipos de trabajo para desarrollar la actividad propuesta. La fórmula que hemos definido para hacer el cálculo de la adopción de prácticas está dada por:

$$AdNp = \left\{ \frac{1}{14} [(NR_S * 100) + (NR_A * 50) + (NR_I * 10)] \right\} * 100 \quad (8)$$

Donde *AdNp* corresponde a Adopción de Prácticas, *NR_S* es el número de respuestas satisfactorias, *NR_A* es el número de respuestas aceptables, *NR_I* es el número de respuestas insatisfactorias. El factor (1/14) se relaciona con el número de características contenidas en el instrumento de observación (Adopción de Nuevas Prácticas Equipos de Trabajo). El valor 100, 50 y 10 que operan con el *NRS*, *NRA*, *NRI* corresponden a la valoración dada al número de observaciones con evaluación Satisfactoria, Aceptable e Insatisfactorio respectivamente. Para esta métrica se aceptable se tendrán en cuenta valores superiores a 50%.

4.3.3.6 Ejecución del Estudio

Previo a la ejecución de este estudio se realizaron durante un mes jornadas de capacitación en Scratch con los cinco cursos, comprendidas entre el 3 de septiembre de 2012 y el 2 de octubre de 2012, es de anotar que estas capacitaciones se realizaron en las aulas clase de cada curso teniendo en cuenta la facilidad de los recursos de la institución. Esta capacitación comprendió aspectos como reconocimiento de las máquinas (equipos portátiles), manejo del entorno de programación Scratch (objetos, escenarios, controles, menús, etc), y ejercicios de clase y en casa. La capacitación fue realizada en presencia del docente del área de tecnología de la institución previo acuerdo con el coordinador. Una vez terminada la capacitación se procedió al diseño del estudio.

Este tercer estudio de caso se ejecutó en el periodo comprendido entre el 3 de septiembre y 19 de noviembre de 2012. Para este estudio se diseñó una actividad colaborativa la cual se describe más adelante. La actividad se realizó igualmente durante 5 sesiones, cada sesión con un curso, la cual contó con una duración de una hora y media.

Los equipos de trabajo se mantuvieron iguales en su conformación, además de la distribución como equipos de control y equipos experimentales para seguir manteniendo en la investigación el factor de comparación entre ellos que permita evaluar la incidencia del modelo propuesto.

La descripción de esta actividad se presenta con más detalle en el Anexo G.

³⁷ <http://ctb.ku.edu/es/tablecontents/seccionprincipal19.6.aspx>

4.3.3.7 Resultados Cuantitativos

Mediciones Directas y Análisis de Resultados

Comportamiento:

En la figura 47 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos para los equipos experimentales y de control ver Anexo J. En esta actividad los sujetos investigados, realizaron un proyecto de desarrollo de software, donde se aplicaron conceptos de algoritmia y programación básica enfocada a los niños.

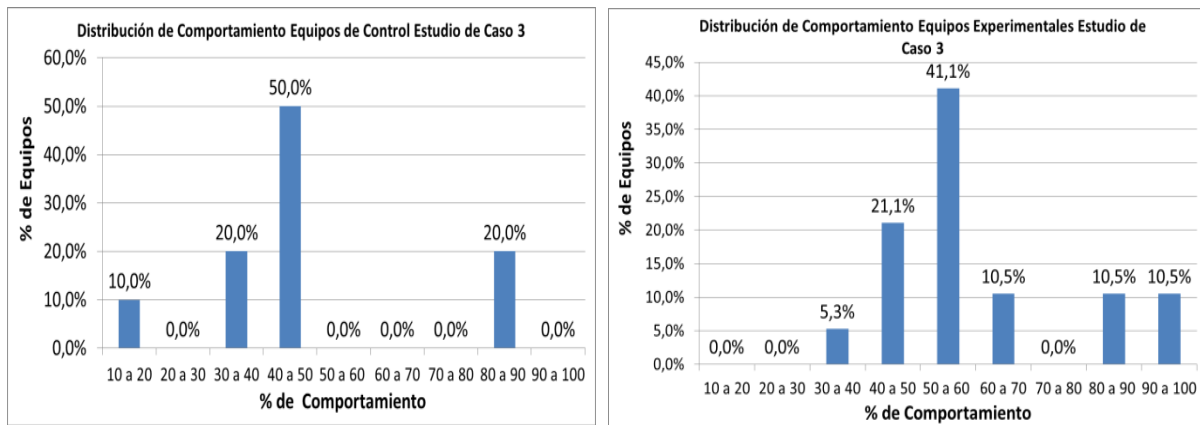


Figura 47. Distribución de Comportamiento Equipos Experimentales y de Control Estudio de Caso 3

En la figura 47 se observa que el 41,1% de equipos experimentales evidenciaron una frecuencia de características definidas en el instrumento de observación de esta medición entre el 50% y 60%. Lo cual indica que a partir de las prácticas entregadas para este estudio rescataron varias para poder aplicarlas en favor de las actitudes, acciones o formas de proceder frente a esta actividad que les permitiera tener mejor desempeño a nivel grupal. Este resultado se debe también a que la actividad les dió más confianza, más seguridad y mejor actitud frente a la toma de decisiones para cumplir con las tareas propuestas para este caso. En comparación con los equipos de control los experimentales tienen mejor tendencia de la población a mejores resultados entendiéndose como mayor frecuencia de características que se rescataron y se aplicaron en la actividad a partir de las entregadas al principio. Hay una diferencia significativa lo que pone en evidencia a los experimentales como la población que afronto la actividad con mejores actitudes de trabajo que los de control.

Productividad:

En la figura 48 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

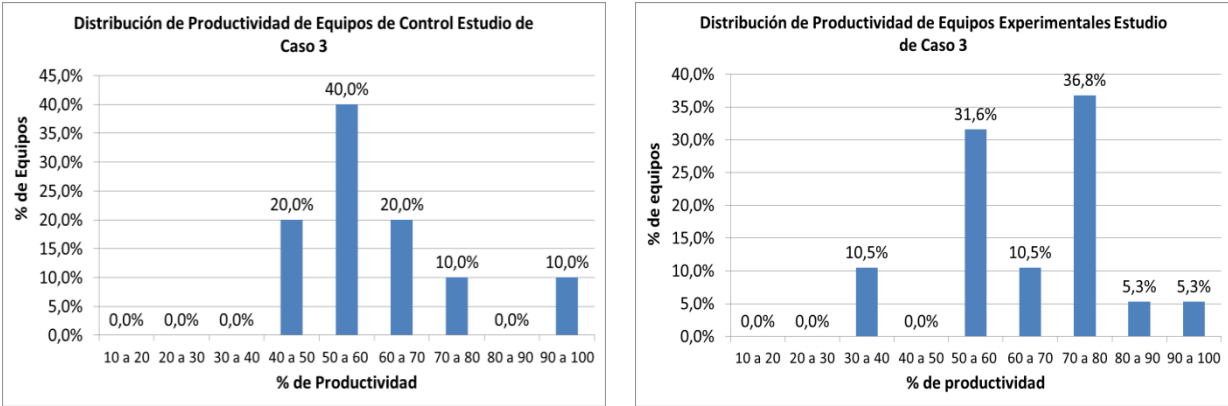


Figura 48. Gráfico de Distribución de Productividad de Equipos Experimentales Estudio de Caso 3

En la figura 48 se observa que los equipos experimentales demostraron con esta actividad que podían producir buenos resultados en las tareas asignadas y concretamente en los productos entregados al finalizar la sesión. Muestra de ello se refleja en la gráfica de distribución de productividad, donde el 36,8% de los equipos lograron evidenciar con su trabajo la aplicación de características definidas en el instrumento de esta medición y la aplicación de la metodología para esa actividad logrando frecuencias entre el 70% y el 80%. Aunque los resultados de los equipos experimentales en el gráfico indiquen variaciones y altibajos, es notable el desempeño aceptable en las entregas del producto final.

Calidad:

En la figura 49 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos, ver Anexo J.

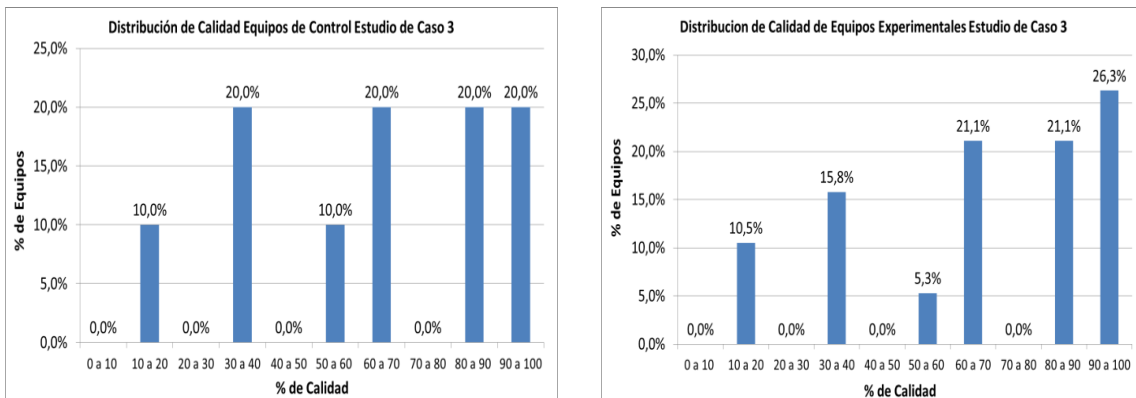


Figura 49. Gráfico de Distribución de Calidad de Equipos Experimentales Estudio de Caso 3

En la figura 49 se observa que los equipos experimentales para esta actividad obtuvieron mejores productos logrando evidenciarse características acordes a la esperadas con una frecuencia entre 80% y 100%, es decir el 47,4% de los equipos experimentales lograron obtener un desarrollo con calidad aceptable. Se presenta una población de experimentales y de control un poco estrecho en cuanto a resultados para esta medición y se atribuye en cierta forma a que los equipos de control de forma natural observaban los desarrollos de los equipos experimentales y los trataban de replicar. Aunque no es un comportamiento adecuado, es aceptable dada la intensión de los equipos de control por realizar la actividad de manera correcta y obtener buenos resultados.

Adopción de Prácticas:

En la figura 50 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos para los equipos experimentales y de control, ver Anexo J.

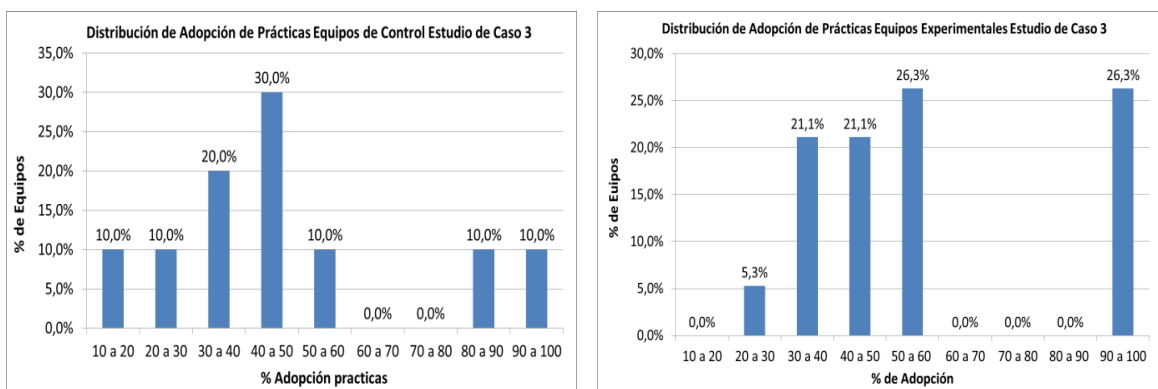


Figura 50. Distribución de Adopción de Prácticas Equipos Experimentales y de Control Estudio de Caso 3

En la figura 50 se observa que en los equipos experimentales hay una buena aceptación y aplicación de las prácticas que fueron entregadas a este grupo. Observemos que el 26.3% de estos equipos se apropiaron de las prácticas y las utilizaron para desarrollar las tareas y en general el producto final, logrando alcanzar muy buenos resultados. Hay una mejor tendencia de la población experimental hacia la obtención de mejores resultados en cuanto a la adopción de las prácticas, sin embargo, no hay que descartar que también se presentaran equipos experimentales con bajo reporte de adopción de estas prácticas, por ejemplo se observa que 21.1% de ellos tuvieron frecuencia de características propias de esta medición entre el 30% y el 50% de aplicación de prácticas en el cumplimiento de esta última actividad. Esto supone que aunque los resultados que se observaron en los

equipos experimentales fueron relativamente aceptables, se debe mejorar para que se logre un resultado más uniforme que y garantice respuestas positivas en este aspecto.

Adopción de Nuevas Prácticas:

En la figura 51 se muestra el resultado gráfico obtenido de esta valoración agrupada en intervalos de 10 valores medidos en porcentajes por equipos para los equipos experimentales y de control. Ver Anexo J.

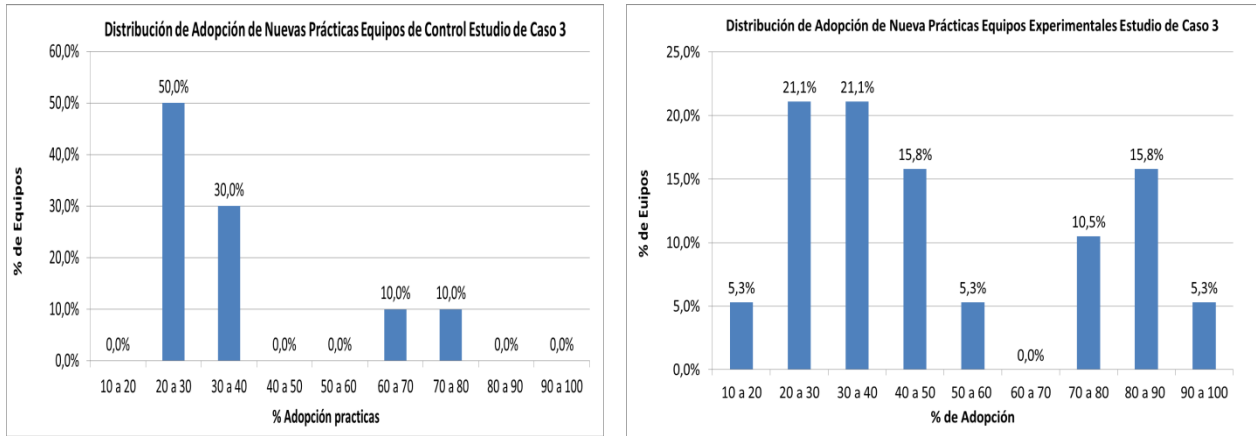


Figura 51. Distribución de Adopción de Nuevas Prácticas Equipos Experimentales y de Control Estudio de Caso 3

En la figura 51 se observa que de los equipos experimentales el 15.8% lograron evidenciar características de adopción de nuevas prácticas para esta medición con una frecuencia entre el 80% y el 90%. La tendencia de los equipos experimentales se concentra hacia la derecha lo cual indica que hay una buena aceptación y apropiación de los equipos en acoger estas características y adaptarlas a la actividad. A pesar de que también hay equipos con baja adopción de prácticas se puede contrastar con las anteriores mediciones las razones por las cuales existen impactos significativos en aspectos como la productividad y la calidad a través de un análisis que me permita observar el conjunto de prácticas que me define las que más impacto tienen cuando se aplica el modelo finalmente en una actividad de desarrollo software.

Para contrastar estos resultados se procedió a realizar un estudio comparativo con respecto a la adopción de nuevas prácticas con el fin de evidenciar la presencia de estas en los equipos de trabajo.

En la figura 53 se puede observar que la adopción de prácticas se ve reflejada de manera positiva en el comportamiento de los equipos, lo cual indica que la inclusión de estas al proceso *ChildProgramming* genera buenas condiciones para el trabajo en equipo, en la figura 52 el análisis de dispersión con respecto a la productividad también es notable, la

incidencia de las prácticas en el desarrollo de la actividad genero confianza y seguridad lo cual permitió que los equipos produjeran resultados satisfactorios.

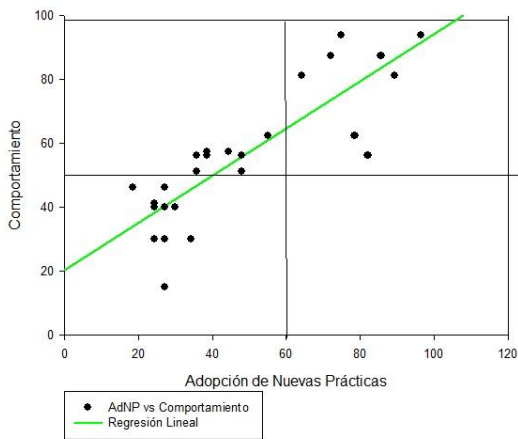


Figura 53. Gráfico de Dispersión Adopción Nuevas Prácticas – Comportamiento

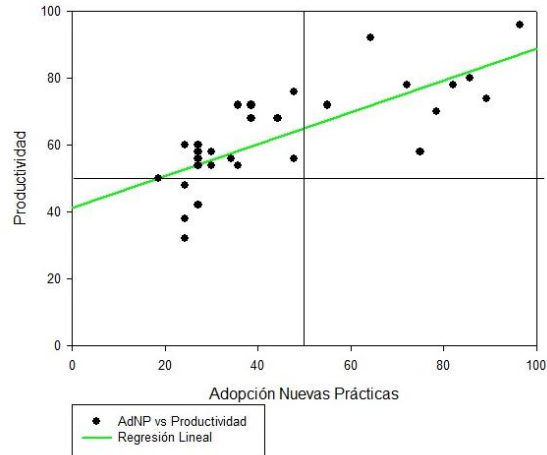


Figura 52. Gráfico de Dispersión Adopción de Nuevas Prácticas – Productividad

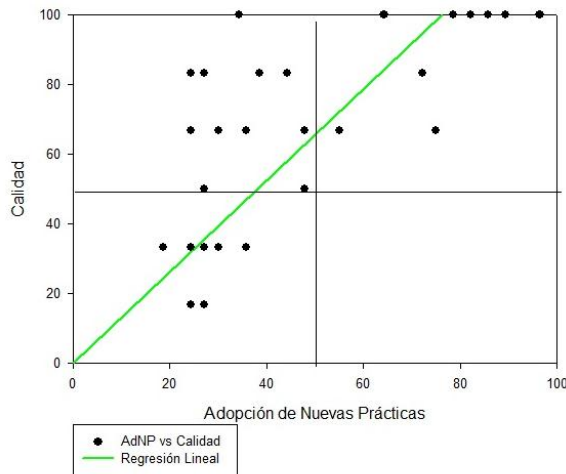


Figura 54. Gráfico de Dispersión Adopción de Nuevas Prácticas – Calidad

En la figura 54 se observa que a nivel general la calidad de las soluciones entregadas fue satisfactoria, la influencia de las prácticas motivo a los equipos a preocuparse por generar mejores resultados y mostrar buenos trabajos. Es de aclarar que en los gráficos de dispersión analizados incluyen a toda la población de estudio (grupos experimentales y de control) haciendo la notación que los grupos de control se encuentran ubicados en las zonas con más baja influencia (de 0 a 45%) de adopción de prácticas.

Para obtener el conjunto final de prácticas “ChildProgramming” después de la recolección de datos y los análisis realizados se procedió a determinar este conjunto haciendo uso de la teoría de conjuntos. Para ello inicialmente se sustrajo de los diagramas de dispersión

la información de los equipos experimentales por cuadrantes (ver figura 55), donde fue más notable la aplicación de las prácticas y determinar las frecuencias de estas para cada equipo.

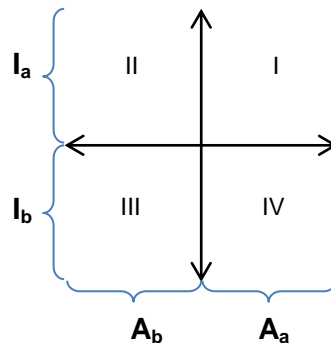


Figura 55. Regiones de la Gráfico de Dispersión

Cada cuadrante cuenta con una descripción como se menciona a continuación:

- ✓ Cuadrante I: Representado por C_{AaIa} donde C hace referencia al Conjunto formado por los elementos del primer cuadrante, AaIa corresponde a “Adopción alta, Indicador alto”.
- ✓ Cuadrante II: Representado por C_{AbIa} donde C hace referencia al Conjunto formado por los elementos del primer cuadrante, AbIa corresponde a “Adopción baja, Indicador alto”.
- ✓ Cuadrante III: Representado por C_{AbIb} donde C hace referencia al Conjunto formado por los elementos del primer cuadrante, AbIb corresponde a “Adopción baja, Indicador bajo”.
- ✓ Cuadrante IV: Representado por C_{AaIb} donde C hace referencia al Conjunto formado por los elementos del primer cuadrante, AaIb corresponde a “Adopción alta, Indicador bajo”.

Así mismo para determinar cuáles son las mejores prácticas dentro de cada cuadrante y realizar el respectivo cruce de los mismos se definieron cinco indicadores así:

- ✓ Prácticas de Menor Impacto al Indicador: $C_{AaIa} - C_{AbIa}$
- ✓ Prácticas de Mayor Impacto al Indicador: $C_{AbIb} - C_{AaIa}$
- ✓ Prácticas de Mayor Impacto al Indicador’: $C_{AbIa} - C_{AbIb}$
- ✓ Prácticas de Menor Impacto al Indicador’: $C_{AaIb} - C_{AaIa}$
- ✓ Prácticas que Entorpecen el Trabajo: $C_{AaIb} - C_{AbIa}$

Con esta información se procede a realizar las respectivas operaciones de conjuntos y así ir seleccionando las prácticas potencialmente adecuadas y válidas para ChildProgramming. Para más detalle dirigirse ver Anexo K.

Finalmente y después del resultado arrojado por cada operación, las prácticas resultantes son:

Adopción de prácticas:

Comportamiento:	P ₁ , P ₃ , P ₄ , P ₅ , P ₆ , P ₈ , P ₉ , P ₁₀	}	P ₁ , P ₃ , P ₄ , P ₆ , P ₈ , P ₉
Productividad:	P ₁ , P ₃ , P ₄ , P ₆ , P ₈ , P ₉ , P ₁₀		
Calidad:	P ₁ , P ₃ , P ₄ , P ₆ , P ₈ , P ₉		

P₁: Entienden Conceptos	P₃: Siguen órdenes
P₄: Aceptar y respetan reglas	P₆: Voluntad para el trabajo en equipo
P₈: Diseño simple e incremental	P₉: Entregable del producto

Adopción de nuevas prácticas:

Comportamiento:	P ₁ , P ₂ , P ₆ , P ₈ , P ₉ , P ₁₁ , P ₁₂ , P ₁₄	}	P ₁ , P ₆ , P ₈ , P ₁₄
Productividad:	P ₁ , P ₆ , P ₈ , P ₉ , P ₁₁ , P ₁₄		
Calidad:	P ₁ , P ₂ , P ₆ , P ₈ , P ₁₄		

P₁: Trabajo en pares	P₆: Espacio informativo
P₈: Trabajo en equipo	P₁₄: Expresión de dudas y comentarios

En la tabla 16 se presenta la selección final resultado del análisis anterior.

No. Práctica	Prácticas ChildProgramming
1	Entienden Conceptos
2	Aceptar y Respetar Reglas
3	Seguir Ordenes
4	Voluntad para el Trabajo en Equipo
5	Diseño Simple e Incremental
6	Entregable del Producto
7	Trabajo en Pares
8	Espacio Informativo
9	Trabajo en Equipo
10	Expresión de Dudas y/o Comentarios

Tabla 16. Prácticas Resultantes ChildProgramming

La evaluación de estas prácticas permitió confirmar cuales se ajustan de manera adecuada al desarrollo de actividades de este tipo y así obtener el conjunto final denominado “*ChildProgramming*”.

4.3.3.8 Resultados Cualitativos

En la tabla 17 se presenta un análisis cualitativo teniendo en cuenta las observaciones realizadas en este estudio de caso.

Equipos de Control	Equipos Experimentales
Presentaron una buena actitud frente a la actividad que realizaron en Scratch, mostrando entusiasmo y motivación. Sin embargo, por la falta de conocimiento de las prácticas ágiles, cognitivas y colaborativas no pudieron mejorar en el aspecto productivo porque les faltó más apoyo como equipo ya que los resultados no fueron tan buenos.	En todo momento de esta actividad tuvieron una buena actitud, desarrollándola con entusiasmo, motivación, manteniendo la voluntad para trabajar en equipo. Desde el primer momento se tomaron la actividad más como un juego.
Hubo poca presencia de roles, tratando de organizarse para cumplir con el objetivo. Se presentaban discusiones porque no tenían quien pusiera orden e hiciera respetar las reglas para dedicarse a las tareas en su totalidad.	Basaron su forma de trabajar mediante el compromiso de equipo, con la presencia de un líder que controlara la ejecución de las tareas, y se agruparon en parejas por tarea para poder distribuirse las funciones y tener mejor desempeño en la actividad.
Había comunicación entre ellos pero faltaba quien les sirviera de mediador para escuchar las opiniones de los demás integrantes.	El éxito que tuvieron los equipos se basó fundamentalmente en la buena comunicación de los integrantes para acoplarse, sobre todo cuando todos tenían diferentes tareas que cumplir y posteriormente unirlos de la mejor manera.
Algunos tenían dudas con respecto a ciertas funciones, sin embargo no las discutieron o las trataron de aclarar en equipo, sino que se quedaron con la duda y por ello también les influyó en el resultado presentando.	Usaban la expresión y solución de dudas entre el guía y los integrantes del equipo para tratar de aclararlas y solucionarlas juntos. Había más apoyo grupal lo que se vio reflejado en varios equipos en su solución.
El producto entregado presentaba diversos detalles erróneos en cuanto a la definición de variables, algunos conceptos ambiguos que se reflejaron en la solución plasmada en el programa que presentaron en Scratch.	Los equipos aunque tenían defectos en la parte de la presentación del programa en Scratch, lograron tener un buen desempeño en cuanto a la parte lógica de la solución del problema

Tabla 17. Análisis Cualitativo Equipos Experimentales y Equipos de Control – Estudio de Caso 3

4.3.3.8.1 Apreciaciones de los investigadores durante el desarrollo del estudio de caso

En este estudio fue evidente el trabajo constante de los niños en cada uno de sus equipos de trabajo, manifestando su interés y entusiasmo para desarrollar la actividad. Los equipos de trabajo apropiaron las prácticas entregadas de forma positiva las cuales fueron

reflejadas en el desarrollo de sus trabajos, fue notable como desempeñaron con agilidad el desarrollo de software en los equipos quienes se veían entregados, comprometidos y colaboradores entre sí para llegar al objetivo propuesto.

Los equipos experimentales mantuvieron dinámicas propias de ellos como mostrar entre sí lo que estaban trabajando, comentarlo con sus compañeros, emocionarse por que les funcionaba alguna estructura o simplemente porque tenían claro a donde querían llegar, los equipos de control mantuvieron su comportamiento desorganizado, algunos estuvieron distantes y poco comprometidos, debido a la falta de orientación pero en general se dieron sus formas para llegar a una solución.

Podemos decir que este estudio fue satisfactorio, los niños evidenciaron claramente la dinámica de trabajo en equipo y fue muy interesante ver como respetaban las reglas y las asumían con tal responsabilidad y compromiso. Es de anotar que a pesar de su falta de experiencia en este tipo de actividades, los niños estuvieron todo el tiempo con la mejor disposición para trabajar, para aprender, conocer las nuevas temáticas y actividades a las que se iban a enfrentar.

Las observaciones realizadas registradas en videos y fotografías del desarrollo de esta actividad se observa en las figuras 56 y 57. Ver también Anexo Digital.



Figura 56. Imágenes Proceso de Desarrollo de los Equipos – Estudio de Caso 3



Figura 57. Imágenes Proceso de Desarrollo de los Equipos - Estudio de Caso 3

4.3.3.9 Amenazas de Validez

A continuación se resume un cierto número de factores que ponen en peligro la validez interna del estudio:

- Debido a que la actividad se desarrolló en un espacio reducido, las mesas de trabajo se encontraban muy juntas se corría el riesgo que los equipos fácilmente

observaban el trabajo de otros y por ende tomaban las ideas y las adaptaban para su propio beneficio

- Presencia de niños en los equipos de trabajo con alto dominio de los temas o mejor manejo de las herramientas, que provoquen distanciamiento a otros en la participación de la actividad.
- Intervención del Docente del área de tecnología en los equipos para apoyar el desarrollo de los programas en Scratch.
- El tiempo dispuesto para desarrollar las tareas propuestas, no es suficiente dado que inicialmente los equipos de trabajo deben dedicar parte de ese tiempo para planear su estrategia de desarrollo y la aplicación de la misma.

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

En este capítulo se expone los aportes concluyentes del trabajo investigativo, los aspectos que se tuvieron en cuenta en la aplicación del modelo, las pautas para realizar trabajos investigativos posteriores encaminados en la misma temática y que sirvan de referencia para mejorar aspectos relevantes del modelo “ChildProgramming”.

5.1 CONCLUSIONES

Este trabajo expone el resultado práctico de una propuesta de aprendizaje en equipo que permite a niños de educación básica primaria de últimos niveles, asimilar conceptos de programación mediante la aplicación de prácticas ágiles, cognitivas y colaborativas adaptados en un ambiente infantil. Se mantuvo un modo de trabajo divertido y motivante que les ayudó a ver el proceso de una forma diferente a las formas de aprendizaje tradicionales. Por esta razón el trabajo que se alcanzó durante todo el proceso fue satisfactorio en la medida que los niños fueron apropiándose de los contenidos tratados a través de las actividades diseñadas y realizadas en cada estudio de caso propuesto.

Basados en el desarrollo de este trabajo de investigación, se expone a continuación las conclusiones a las que se llegó al finalizar este proyecto:

- ✓ El modelo ChildProgramming define una guía de trabajo colaborativo útil para equipos conformados por niños entre los 8 y 10 años, que les permite realizar actividades en torno un tema específico en este caso la programación e inducirlos de manera sencilla en un proceso de desarrollo de software ejecutado por ellos aplicando características del desarrollo ágil.
- ✓ ChildProgramming ha permitido definir de forma práctica un proceso para el desarrollo de software adaptable al contexto de los niños. Por tanto, ChildProgramming logró llamar la atención de los niños, dado que se mantuvo un modo de trabajo divertido y motivante que les ayudó a ver el proceso de una forma diferente a las formas de aprendizaje tradicionales.
- ✓ Utilizando el Modelo propuesto, se lograron resultados muy positivos, en la parte académica, el resultado entregado por los niños fue de muy buena calidad. En la parte cognitiva los niños y los profesores manifestaron su interés en este tipo de actividades lo que puede permitir llevarlo a otros escenarios donde se aplique en nuevos procesos de enseñanza, en la parte social se logró que los niños desarrollaran habilidades de colaboración para definir estrategias de organización y dinámicas de trabajo en equipo propias para el desarrollo de las actividades propuestas y en la parte investigativa la metodología aplicada permitió recopilar

una fuente importante de información empírica que sirve de base para futuras investigaciones.

- ✓ Para evaluar el modelo Child Programming propuesto en esta tesis se utilizaron los estudios de caso, los que fueron de gran utilidad para la extracción de evidencia, así como la corroboración de la aplicabilidad del modelo. En este caso, se buscó confirmar el impacto de las prácticas ágiles de desarrollo, de aprendizaje colaborativo y de cognición en los equipos de trabajo, centrándonos particularmente en los grupos experimentales como muestras de interés para la investigación. Los equipos de control permitieron contrastar los resultados y determinar la validez de las conclusiones.
- ✓ ChildProgramming está en evolución, por lo tanto permite seguir enriqueciéndolo con inclusión de nuevas y mejores prácticas dada la capacidad y la adaptabilidad a cualquier entorno de trabajo, empleando las mismas características definidas en cuanto a la edad de la población y el nivel de escolaridad para el que fue propuesto.
- ✓ La adaptación de ChPScratch vs 1.0 como Ambiente de Programación para los niños, permite realizar trabajo en red, compartir elementos y ejecutar acciones desde su propio código, brindando a los niños una nueva experiencia de trabajo.
- ✓ La adaptación del Ambiente Virtual motiva a los niños a trabajar de manera más cómoda y divertida a través de la interacción en línea entre los integrantes de un equipo de trabajo mediante la utilización de servicios como el chat, el tablero de tareas y el entorno de programación Scratch.
- ✓ El método de estudio de caso es considerado valioso como método de investigación para este proyecto dado su rigurosidad para explorar y obtener un conocimiento sobre el comportamiento, las dinámicas de trabajo y las prácticas empleadas por los equipos de trabajo entorno al desarrollo de software realizado por niños, quienes se perfilan como potenciales ingenieros a futuro. Es importante tener en cuenta los estudios de caso como métodos de investigación porque brindan una cercanía de los investigadores con las unidades de análisis y los sujetos de investigación produciendo una excelente dinámica para la obtención de buenas fuentes de datos y así realizar posteriores análisis de resultados. Además genera a través de experiencias prácticas y reales la interacción y acción en procesos o proyectos de gran envergadura.
- ✓ Como producto de un proceso de investigación rigurosa y documentada se obtuvo una recopilación completa de observaciones audiovisuales, que involucran todo un proceso de trabajo desde la caracterización, capacitación, desarrollo de actividades y reporte de experiencias por parte de los niños. Esta documentación marca un punto de partida importante hacia futuras investigaciones.

5.2 RECOMENDACIONES

Para trabajos posteriores que se realicen basados en este proyecto, se desea en lo posible que se tenga presente durante el proceso de investigación las siguientes recomendaciones:

- ✓ Debido a que en este trabajo de investigación se planteó tomar como población de estudio niños de básica primaria entre los 8 y los 10 años de edad se recomienda planear las actividades de trabajo grupal de manera clara y sencilla, utilizando un lenguaje adecuado que les facilite la comprensión de los conceptos tratados en el proceso o en las actividades, ya que posiblemente pueden llegar a presentarse inconvenientes de este tipo que no aporten en la recolección de la información y entorpezcan el trabajo.
- ✓ Dado que el trabajo de investigación se realiza con niños de básica primaria, se recomienda plantear o diseñar las actividades de los estudio de caso de la manera más lúdica, divertida o entretenedora, con el fin de que los equipos de trabajo mantengan una buena atención o concentración en el proceso, ya que los niños les atrae generalmente actividades que sean agradables o que impliquen “jugar”. Por otro lado si hay un juego siempre debe haber un objetivo o una meta que alcanzar que implique esfuerzo, es por ello que se recomienda igualmente plantear las actividades de trabajo de manera motivante, es decir buscando que los equipos de trabajo realicen las actividades con la mayor calidad, productividad y esfuerzo posible para que logren el premio el cual será la gran motivación que los mantenga con deseos de ser los mejores.
- ✓ Por otro lado el proceso de investigación requiere que las observaciones que se hagan durante las actividades con los niños sean lo más objetivas posibles debido a que pueden presentarse sesgos de información, datos que resulten incoherentes con la realidad evidenciada lo cual puede entorpecer el proceso de aplicación del modelo de aprendizaje propuesto, por ello se recomienda que hayan varios observadores que puedan captar u observar para tener distintos puntos de vista que permitan discutir y analizar con objetividad.

5.3 TRABAJO FUTURO

En este trabajo se han analizado algunos puntos que pueden ser tenidos en cuenta para trabajos futuros a corto y mediano plazo, entre ellos se incluye:

- ✓ Continuar el proceso de investigación en cuanto a la aplicación de nuevos estudios de caso y experimentos controlados en los que se puedan plantear otras

estrategias que puedan resultar más efectivas para la recolección de información, de evidencias de observación y que involucren mayor interacción de los niños con la metodología de aprendizaje de conceptos de desarrollo de la programación manteniendo el mismo objetivo del trabajo.

- ✓ Se espera que a través de la metodología propuesta los docentes del área de la informática de las escuelas o colegios de educación primaria puedan tener una pequeña guía de enseñanza previa del mundo de la programación de software manteniendo un estilo de juego donde ellos vean esta forma de aprender de manera motivante y la pueda reflejar a través de pequeños desarrollos o proyectos de clase que el docente proponga de acuerdo al nivel que van adquiriendo.
- ✓ Se desea que a través de este trabajo se pueda crear un ambiente de trabajo más adecuado o que implique más la interacción de los niños con un sistema bajo un equipo de cómputo y se pueda evidenciar mejores actitudes que las que se presentaron en la investigación presente, ya que permitiría mejorar o refinar el modelo que aquí se expone dándole más soporte y robustez a la misma.

BIBLIOGRAFIA

- [1] A. Druin, C. Solomon. "Designing Multimedia Environments for Children", *Wiley & Sons*, 1996.
- [2] J. Pane. "Designing a Programming System for Children with a Focus on Usability", in *Adjunct Proceedings SIGCHI'98: Conference Summary: Human Factors in Computing Systems*, Los Angeles, CA, pp. 62-63, 1998.
- [3] A. Druin. "The Design of Children's Technology", *Morgan Kaufmann Publishers*, 1998.
- [4] A. Druin. "Cooperative Inquiry: Developing New Technologies for Children with Children Human-Computer Interaction", *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, Lab University of Maryland, pp. 592-599, 1999.
- [5] M. Eisenberg, N. Elumeze, M. MacFerrin, L. Buechley. "Children's Programming, Reconsidered: Settings, Stuff, and Surfaces", *U. of Colorado Dept. of Computer Science. MIT Media Lab*, pp. 1-8, 2009.
- [6] K. Sheingold, "The microcomputer as a symbolic medium". In *R. D. Pea & K. Sheingold (Eds.), Mirrors of minds: Patterns of experience in educational computing*, pp. 198-208, 1987.
- [7] J. Ruiz. "Implicaciones Educativas del Lenguaje LOGO", *CL&E*, pp. 111-118, 1994.
- [8] *Observatorio Tecnológico*. Available: [en línea] <http://recursostic.educacion.es/observatorio/web/ca/software/programacion/593-rafael-alba> [Consulta: 2011 Agosto 28].
- [9] M. Calzadilla. "Aprendizaje Colaborativo y tecnologías de la información y la comunicación", *Revista Iberoamericana de Educación*, 2002.
- [10] A. Aristóteles. "Ética de Nicómaco", Centro de Estudios Constitucionales, Madrid, 1989.
- [11] J. Archer-Kath, D.W. Johnson, R. Johnson. "Individual Versus Group Feedback in Cooperative Groups", *Journal of Social Psychology*, 134 (5), pp. 681-694, 1994.
- [12] E. Turban. "Decision Support and Expert System", *Management Support System*, New Jersey, Prentice Hall, pp. 10-127, 1995.
- [13] J. Bonals. "El Trabajo en Pequeños Grupos en el Aula", Editorial Grao, Vol. 145, 2000.
- [14] Sitio Web: Biblioteca Virtual de las Ciencia en Cuba. Available: [en línea] <http://www.bibliociencias.cu/gsd/collect/libros/index/assoc/HASH0173/c26d62c2.dir/doc.pdf> [Consulta: 12/08/13].
- [15] D. Izuzquiza, M. Gómez. "Tecnología y Aprendizaje Colaborativo en el Diseño de Materiales para Desarrollo del Pensamiento Abstracto en Didáctica de las Matemáticas", *Revista Latinoamericana de Tecnología Educativa*, Universidad Autónoma de Madrid, vol. 3, nº 1, pp. 234, 2004.
- [16] D. E. Leidner y S. L. Jarvenpaa, "The Use of Information Technology to Enhance Management School Education: A Theoretical View", *MIS Quarterly*, vol. 19, nº 3, pp. 265-291, 1995.
- [17] N. Scagnoli. "Estrategias para Motivar el Aprendizaje Colaborativo en Cursos a Distancia", Conferencia Paper University of Illinois at Urban-Champaign USA, 2005.
- [18] M. Chiarani, M. Lucero, y I. Pianucci, "Modelo de Aprendizaje Colaborativo en el Ambiente ACI", *Departamento de Informática, Universidad Nacional de San Luis*, 2003.
- [19] M. Lucero. "Entre el Trabajo Colaborativo y el Aprendizaje Colaborativo", *Revista Iberoamericana de Educación*, pp. 1-21, 2003.

- [20] Sitio Web: Universidad Católica de Temuco. Available: [en línea] <http://www.uctemuco.cl/cedid/archivos/apoyo/Aprendizaje%20Colaborativo.pdf> [Consulta: 12/08/10]
- [21] S. D. Amaro, J. C. Valverde. “Metodologías Ágiles”, *Universidad Nacional de Trujillo, Escuela de Informática*, Trujillo – Perú, 2007
- [22] P. Rodriguez. “Estudio de la Aplicación de Metodologías Ágiles para la Evaluación de Productos Software”, *Universidad Politécnica de Madrid, Facultad de Informática*, 2008.
- [23] B. Boehm. “A View of 20th and 21st Century Software Engineering”, in *28th International Conference on Software Engineering*, pp. 12-29, Shanghai, China 2006.
- [24] B. Boehm. “Software Engineering Economics”, in *Prentice Hall*, 1981.
- [25] J. Hurtado. Proyecto SIMEP-SW “Hacia una Línea de Procesos Ágiles Agile SPsL”, *Universidad del Cauca*, 2005.
- [26] B. Boehm, R. Turner. “Balancing Agility and Discipline A guide for the Perplexed”, Addison-Wesley, 2003.
- [27] Sitio Web: Agile Alliance. Available: [en línea] <http://www.agilealliance.org> [Consulta: 2012 Agosto 18].
- [28] J. Canós, P. Letelier, M. Penadés. “Metodologías Ágiles en el Desarrollo de Software”, *DSIC, Universidad Politécnica de Valencia*, 2003.
- [29] A. Orjuela, M. Rojas. “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo”, *Revista Avances en Sistemas e Informática*, pp. 159-171, 2008
- [30] F. P. Brooks, Jr. “No Silver Bullet” *Essence and Accidents of Software Engineering Computer*, pp. 10-19, 1987.
- [31] J. Hurtado, M. Bastarrica, A. Quispe, S. Ochoa. “An MDE approach to software Process Tailoring”, in *Proceeding of the 2011 International Conference on Software and Systems Process (ICSSP '11)*, ACM, New York, NY, USA, pp. 43-52, 2011.
- [32] Beck, K. “Extreme Programming Explained”. Embrace Change, Addison-Wesley, 1999.
- [33] M. Paulk. “Extreme Programming from CMM Perspective”, *IEEE Software*, 2001.
- [34] A. Orjuela, M. Rojas. “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo”, *Revista Avances en Sistemas e Informática*, pp. 159-171, 2008.
- [35] K. Schwaber. “The Scrum Development Process”, *OOPSLA 95 Workshop on Business Object Design and Implementation*, Austin, 1995.
- [36] A. Cockbun. “Agile Software Development”, Addison-Wesley, 2001.
- [37] K. Schwaber, M. Beedle, R. Martin. “Agile Software Development with SCRUM”, Prentice Hall, 2001.
- [38] P. Coad P, E. Lefebvre, J. De Luca. “Java Modeling In Color With UML: Enterprise Components and Process”, Prentice Hall, 1999.
- [39] T. Gilb. “Evolutionary Development”, *ACM SIGSOFT Software Engineering Notes*. Vol. 6, Issue 2, pp. 17, 1981
- [40] A. Boltman, A. Druin. “Children’s Storytelling Technologies: Differences in Elaboration and Recall”, 2001.
- [41] L. Morgado, M. Cruz, K.Kahn. “Using ToonTalk in Kindergartens”, *Proceedings of the IADIS International Conference e Society*, Vol 2, pp. 988 – 994, 2003.
- [42] E. Howard, D. Evans, J. Courte, C. Bishop-Clark. “A qualitative look at Alice and pair-programming”, *Proceedings of ISECON 2006*, 2006.

- [43] C. Rader, C. Brand, C. Lewis. "Degrees of Comprehension: Children's Understanding of a Visual Programming Environment", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 351 – 358, 1997.
- [44] A. Cypher, D. Smith. "KidSim: End User Programming of Simulations", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 27 – 34, 1995.
- [45] S. Crook. "Embedding Scratch in the Classroom", 2009.
- [46] J.M.C. Lin, L.Y Yen, M.C. Yang, C.F. Chen. "Teaching Computer Programming in Elementary Schools: A Pilot Study", National Educational Computing Conference, 2005.
- [47] A. Sánchez-Ruíz, L. Jamba. "FunFonts: introducing 4th and 5th graders to programming using Squeak", in Proceedings of the 46th Annual Southeast Regional Conference on XX (ACM-SE 46). ACM, New York, NY, USA, pp. 24-29, 2008.
- [48] Sitio Web: Wikilibros. Available: [en línea] http://es.wilibooks.org/wiki/Aprendizaje_colaborativo [Consulta: 2012 Agosto 31].
- [49] R. Briggs, and G.J Vreede. *Collaboration Engineering: designing repeatable processes for high value*. Delft University of Technology, University of Arizona: Proc. 38th Hawaii International Conference on System Sciences, 2005.
- [50] *Defining key concepts for collaboration engineering*. Briggs, R.O., y otros. Atlanta: N.C. Romano Jr.v (ed), Proceedings of the Twelfth Americas Conference on Information Systems, 2006, págs. 1-9.
- [51] *Thinklet Design Support Booklet. September*. Kolfshoten, G.L. and Vreede, G.J. 2006.
- [52] *A collaborative software code inspection: the design and evaluation of a repeatable collaboration*. D.L. Dean., A. Fruhling, P.G. Konert, G.J. Vreede, and P. Wolcott. s.l. : Proc. International Journal of Cooperative Information Systems, 2006.
- [53] *Collaboration Engineering with Thinklets to pursue sustained success with group support systems*. R.O. Briggs, J.F. Nunamaker, and G.J. Vreede G.J. s.l.: Proc. Journal of Management Information Systems, 2003.
- [54] Kolfshoten, G.L., y otros. *Thinklets as Building Blocks for Collaboration Processes: A Further Concept*. Lecture notes in computer science, Berlin: Springer Verlag, 2004.
- [55] Briggs, R.O., and Vreede, G.J. *Thinklets: Building Blocks for Concerted Collaboration*. Center for Collaboration Science, University of Nebraska at Omaha, USA. 2009.
- [56] Sitio Web: Dirección Nacional de Servicios Académicos Virtuales Universidad Nacional de Colombia Sede Manizales. Available: [en línea] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060010/lecciones/Capitulo1/modelo.htm> [Consulta: 2013 Enero 18].
- [57] J. Hurtado, C. Collazos, S. Cruz, O. Rojas. "Child Programming: Una Estrategia de Aprendizaje y Construcción de Software basada en la Lúdica, la Colaboración y la Agilidad". *Revista Universitaria RUTIC*, 1(1). 2012.
- [58] Sitio Web: Eduteka. Available: [en línea] <http://www.eduteka.org/EntrevistaLilianaVictor.php> [Consulta: 2013 Enero 12].
- [59] S. Cruz, O. Rojas, J. Hurtado, C. Collazos. "Auto-Organización y Dinámicas de Trabajo en Equipo: Un Caso de Estudio con Escolares de 8 a 10 años". Universidad del Cauca, 2013.
- [60] S. Cruz, O. Rojas, J. Hurtado, C. Collazos. "Proceso Child Programming: Un Modelo de Desarrollo de Software Orientado a los Niños". Universidad del Cauca, 2013.
- [61] Arancibia, V., Herrera, P., Strasser, K. (1999). Teorías cognitivas del aprendizaje. En: Psicología de la educación. México, Alfaomega. pp. 75-96 y 99-101.

- [62] Ferrer-Mico, T., Fernández-Prats, M. À., & Rehacer Sánchez, A. (2012). Impacto de programación Scratch en los estudiantes la comprensión de su proceso de aprendizaje. *Procedia-Social y Ciencias de la Conducta*, 46 , 1219-1223.
- [63] Software & Systems Process Engineering Meta-Model Specification. Group), OMG (Object Management. 2008.
- [64] Yin, R.K. Case Study Research. Desing and Methods. Beverly Hills, Sage. 1984.
- [65] Chetty, S. The Case Study Method for Research in Small and Medium sized firms.
- [66] Runeson, P. and M. Höst, Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg.*, 2009. 14(2): p. 131-164.
- [67] Dart, S. A., Ellison, R. J., Feiler, P. H., Habermann, A. N., & Fritzon, E. P. Overview of software development environments. (1992).
- [68] Sitio Web: Universidad Manuela Beltrán. Available: [en línea] <http://www.umb.edu.co/pdf/modelos%20pedagogicos.pdf> [Consulta: 2013 Enero 12].
- [69] Silva, C., Campos, R., Ramo, O., Método María Montessori, Psicología Educacional Valdebenito, Instituto Profesional Luis Galdanes, 2003.
- [70] Sitio Web: Espacio Logopedico. Available: [en línea] http://www.espaciologopedico.com/articulos/articulos2.php?palabra=montessori&Id_articulo=197 [Consulta: 2013 Enero 12].
- [71] Mounoud, P. El desarrollo cognitivo del niño. *Contextos educativos*, (4), 53. 2001.
- [72] Sitio Web: Educando Portal de Educación Dominicana. Available: [en línea] http://www.educando.edu.do/Userfiles/P0001/File/Que_es_el_constructivismo.pdf [Consulta: 2013 Enero 12].
- [73] Sitio Web: Feliciano Villar Profesor titular Universidad de Barcelona. Available: [en línea] http://personales.ya.com/fvillar/principal/pdf/proyecto/cap_05_piaget.pdf [Consulta: 2013 Febrero 26].
- [74] Sitio Web: Universidad Estatal de Bolivar Ecuador. Available: [en línea] <http://www.biblioteca.ueb.edu.ec/bitstream/15001/50/1/0014.pdf> [Consulta: 2013 Febrero 26].
- [75] Woolfolk, A. E. *Psicología Educativa*. México: Editorial Mexicana, séptima edición. 662 pág. 1999.
- [76] Sitio Web: Unicef Chile. Available: [en línea] <http://www.unicef.cl> [Consulta: 2013 Abril 18].
- [77] Gomez, L. A., Sánchez, A. J. Ambiente Colaborativos en Ludomática. *Revista Informática Educativa*, 13, 1. 2000.
- [78] Hansen, K. M. Agile Environments–Some Patterns for Agile Software Development Facilitation. In *The First Scandinavian Conference on Pattern Languages of Programs (VikingPLoP 2002)*, Højstrupgård, Denmark. 2002.
- [79] Sitio Web: Colombia Aprende. Available: [en línea] <http://www.colombiaaprende.edu.co/html/productos/1685/w3-article-288989.html> [Consulta: 2013 Febrero 26].