

# **Integrando el Desarrollo del Pensamiento Computacional y los Conceptos T&E STEM en el Enfoque Metodológico de ChildProgramming**



**Pedro del Socorro Gómez Álvarez  
Ángel Alveiro Gómez Noguera**

Director: PhD. Julio Ariel Hurtado Alegría

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Sistemas**

**Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)**

**Línea de Investigación en Ingeniería de Software**

Popayán, febrero de 2023

# **Integrando el Desarrollo del Pensamiento Computacional y los Conceptos T&E STEM en el Enfoque Metodológico de ChildProgramming**



Monografía de Trabajo de Grado para optar al título de Ingeniero en  
Electrónica y Telecomunicaciones

**Pedro del Socorro Gómez Álvarez**  
**Ángel Alveiro Gómez Noguera**

Director: PhD. Julio Ariel Hurtado Alegría

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Sistemas**

**Grupo de Investigación y Desarrollo en Ingeniería de Software (IDIS)**

**Línea de Investigación en Ingeniería de Software**

Popayán, febrero de 2023

# NOTA DE ACEPTACIÓN

---

---

---

---

PRESIDENTE DEL JURADO

---

JURADO

# AGRADECIMIENTOS

*A Dios, por el cumplimiento de sus promesas y por permanecer siempre a mi lado. A mi Esposa Bibiana Galvis y mis hijos Santiago, Samuel y Juan, por su amor y su apoyo incondicional en todas las decisiones de mi vida. A mi Madre y mis hermanos quienes siempre me han acompañado en mis triunfos y fracasos a lo largo de la vida.*

*A mi familia, amigos y demás personas que de una u otra forma marcan y cambian mi existencia.*

*A la Universidad del Cauca y docentes por las enseñanzas y momentos vividos, determinantes en mi vida profesional.*

**Pedro del Socorro Gómez Álvarez**

*Al Señor todopoderoso que me ha dado las fuerzas para seguir adelante a mis hijos, Deiby y Alejandra Gómez mi esposa Sandra Fernández por toda la paciencia en mis horas de trabajo. También a las almas de mis difuntos padres José Rafael Gómez y Alejandrina Noguera quienes en vida se esforzaron y me aconsejaron para que siguiera adelante a todos mis hermanos quienes comparten este logro. ¡Gracias!*

**Ángel Alveiro Gómez**

## **Agradecimientos Especiales**

*Gracias al Ingeniero Julio Ariel Hurtado por la dirección de este proyecto, su disposición, apoyo y motivación en todos los momentos.*

*A la ingeniera Gineth Magali Cerón Ríos por su apoyo en las etapas del desarrollo de esta investigación.*

*Y muchas gracias a todas aquellas personas que colaboraron o participaron, de una u otra forma, en la realización de este trabajo.*

# Tabla de contenido

|   |    |
|---|----|
| 1. Capítulo 1 Introducción .....  | 12 |
| 1.1 Planteamiento del problema .....  | 13 |
| 1.2 Objetivos .....   | 15 |
| 1.2.1 Objetivo general. ....  | 15 |
| 1.2.2 Objetivos específicos.....  | 15 |
| 1.3 Metodología.....  | 16 |
| 1.3.1 Fase de exploración. ....   | 16 |
| 1.3.2 Fase de construcción del modelo.....                                  | 16 |
| 1.3.3 Fase de evaluación. ....  | 17 |
| 1.4 Aportes.....  | 17 |
| 1.4.1 En el ámbito académico. ....  | 17 |
| 1.4.2 En la investigación.....  | 17 |
| 1.5 Resultados obtenidos .....  | 18 |
| 1.6 Organización del documento .....  | 18 |
| 2. Capítulo 2 Marco teórico y revisión de la literatura.....                | 19 |
| 2.1 Marco teórico.....  | 19 |
| 2.1.1 Pensamiento computacional (CT).....                                   | 19 |
| 2.1.2 Educación STEM.....   | 24 |
| 2.1.3 Enseñanza de tecnología e ingeniería (T&E).....                       | 25 |
| 2.1.4 ChildProgramming.....   | 26 |
| 2.1.5 Scratch. ....   | 28 |
| 2.2 Revisión de la literatura .....   | 30 |
| 2.2.1 Selección de los documentos. ....                                     | 30 |
| 2.2.2 Distribución de los documentos seleccionados.....                     | 32 |
| 2.2.3 El Pensamiento computacional y la educación STEM. ....                | 32 |
| 2.2.4 Integración del pensamiento computacional en Ingeniería.....          | 34 |
| 2.2.5 Evaluación de las habilidades de pensamiento computacional.....       | 38 |
| 2.2.6 ChildProgramming en el desarrollo del pensamiento computacional. .... | 39 |
| 2.2.7 Competencias STEAM y habilidades de CT. ....                          | 39 |

|   |     |
|---|-----|
| 3. Capítulo 3 Modelo CT4E: Pensamiento computacional para la formación de ingenieros en el contexto STEM..... | 43  |
| 3.1 Modelo conceptual de CT4E.....  | 44  |
| 3.1.1 Estructura del modelo ChildProgramming.....   | 45  |
| 3.1.2 El modelo CT4E en ChildProgramming. ....  | 46  |
| 3.2 Arquitectura conceptual de CT4E .....   | 46  |
| 3.2.1 Contexto del modelo CT4E.....   | 48  |
| 3.2.2 Actores del modelo CT4E.....  | 48  |
| 3.2.3 Dimensiones de ChildProgramming-CT4E. ....  | 49  |
| 3.2.4 Roles del modelo CT4E.....  | 50  |
| 3.3 Paquetes metodológicos CT4E.....  | 51  |
| 3.3.1 Áreas conceptuales CT4E. ....   | 52  |
| 3.3.2 Guía de actividades CT4E.....   | 56  |
| 3.3.3 Recomendaciones para la aplicación de CT4E.....   | 74  |
| 4. Capítulo 4 Planeación, diseño, ejecución y resultados del estudio .....                                    | 76  |
| 4.1 Planeación del estudio.....   | 76  |
| 4.1.1 Metodología.....  | 76  |
| 4.2 Diseño del estudio de caso descriptivo .....  | 78  |
| 4.2.1 Selección de los participantes del estudio. ....  | 78  |
| 4.2.2 Contexto de la investigación. ....  | 79  |
| 4.2.3 Variables observadas. ....  | 80  |
| 4.2.4 Preguntas de Investigación.....   | 81  |
| 4.2.5 Objetivo del estudio. ....  | 81  |
| 4.2.6 Selección del estudio.....  | 81  |
| 4.2.7 Los sujetos de investigación. ....  | 81  |
| 4.2.8 Instrumentos y técnicas para recolección de información estudio de caso descriptivo. ....               | 82  |
| 4.2.9 Ejecución, resultados y análisis de la fase exploratoria estudio de caso descriptivo. ....              | 86  |
| 4.2.10 Resultados cuantitativos.....  | 92  |
| 4.2.11 Ejecución, resultados y análisis de la fase comparativa de estudio de caso descriptivo. ....           | 99  |
| 4.3 Discusión.....  | 111 |

|   |     |
|---|-----|
| 5. Capítulo 5 Conclusiones y trabajos futuros ..... | 113 |
| 5.1 Conclusiones .....                              | 113 |
| 5.2 Trabajos futuros.....                           | 114 |
| 6. Bibliografía.....                                | 115 |

# Índice de figuras

|   |     |
|---|-----|
| Figura 1. Descripción del marco conceptual del Pensamiento Computacional.....   | 21  |
| Figura 2. Modelo inicial planteado de ChildProgramming .....  | 26  |
| Figura 3. Dimensiones de ChildProgramming [15].....   | 27  |
| Figura 4. Ciclo de vida ChildProgramming .....  | 27  |
| Figura 5. Proceso de selección de las publicaciones .....   | 31  |
| Figura 6. Distribución a) por cantidad y b) por porcentaje de los documentos seleccionados para revisión .....  | 32  |
| Figura 7. Competencias y dimensiones STEAM [4] .....  | 40  |
| Figura 8. Participantes Programando en Scratch durante la Fase Exploratoria del Estudio de Caso Descriptivo.....  | 44  |
| Figura 9. Actividad Desconectada, Red de clasificación de seis entradas .....   | 45  |
| Figura 10. Arquitectura ChildProgramming-CT4E .....   | 47  |
| Figura 11. Arquitectura del Modelo CT4E .....   | 48  |
| Figura 12. Modelo CT4E y sus paquetes metodológicos .....   | 51  |
| Figura 13. Áreas Conceptuales y sus sub-paquetes .....  | 53  |
| Figura 14. Relación Transversal de las Competencias STEM y los Conceptos de CT con ChildProgramming .....   | 56  |
| Figura 15. Guía de actividades modelo CT4E.....   | 57  |
| Figura 16. Plantilla práctica cognitiva.....  | 61  |
| Figura 17. Plantilla práctica ágil .....  | 62  |
| Figura 18. Plantilla práctica colaborativa.....   | 63  |
| Figura 19. Ciclo de vida Modelo CT4E.....   | 72  |
| Figura 20. Ruta de Aplicación del Modelo CT4E.....  | 75  |
| Figura 21. Edificio de Ingenierías, Universidad del Cauca .....   | 79  |
| Figura 22. Participantes del Estudio de Caso Descriptivo .....  | 80  |
| Figura 23. Evidencia fotográfica – Sesión 1 .....   | 87  |
| Figura 24. Plegable Construye tu Primer Programa en Scratch.....  | 88  |
| Figura 25. Evidencia fotográfica - Sesión 2.....  | 89  |
| Figura 26. Plegable “Red de clasificación de seis entradas”.....  | 89  |
| Figura 27. Evidencia fotográfica - Sesión 3.....  | 91  |
| Figura 28. Evidencia fotográfica - Sesión 4.....  | 92  |
| Figura 29. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Post-Test de Fase exploratoria del estudio de caso descriptivo ..... | 92  |
| Figura 30. Distribución de las respuestas a la pregunta N.6 a) en el Pre-test y b) en el Post-test de Fase exploratoria del estudio de caso descriptivo ..... | 94  |
| Figura 31. Comparativo a) Pre-test y b) Pos-test sobre la apropiación de los conceptos de CT.....   | 97  |
| Figura 32. Resultados del análisis con Dr. Scratch del código de a) la sesión 2. “Construyendo tu primer programa en Scratch” .....                           | 98  |
| Figura 33 Cantidad de aciertos Pre Test.....  | 106 |

|   |     |
|---|-----|
| Figura 34 Cantidad de aciertos Pos-Test.....                              | 106 |
| Figura 35. Comparativo Pre Test y Post Test - Concepto Computacional..... | 108 |
| Figura 36. Comparativo Pre Test y Post Test - Tarea Requerida.....        | 109 |
| Figura 37. Comparativo Pre Test y Post Test - Capacidad Evaluada.....     | 110 |

# Índice de tablas

|  |    |
|--|----|
| Tabla 1. Conceptos computacionales .....   | 22 |
| Tabla 2. Prácticas computacionales [34].....   | 22 |
| Tabla 3. Perspectivas computacionales .....  | 23 |
| Tabla 4. Competencias y Dimensiones STEM [4] .....   | 25 |
| Tabla 5. Modelo para el desarrollo de habilidades de CT propuesto por Atmatzidou y Demetriadis, condensado de [54] y [55].....                                 | 37 |
| Tabla 6. Competencias y dimensiones STEM con su definición[4], [48].....   | 40 |
| Tabla 7. Roles adaptados por CT4E como una extensión de ChildProgramming .....   | 50 |
| Tabla 8. Competencias STEM con su definición[4] .....  | 55 |
| Tabla 9. Plantilla No. 1 de Actividades de resolución de problemas.....  | 58 |
| Tabla 10. Plantilla No. 2 de Actividades de entendimiento compartido .....   | 58 |
| Tabla 11. Plantilla No. 3 de Actividades evaluación CT (Pre-Test, Pos-Test) .....  | 59 |
| Tabla 12. Plantilla No. 4 de Actividades evaluación CT (Observación) .....   | 60 |
| Tabla 13. Tipo de prácticas ChildProgramming.....  | 60 |
| Tabla 14. Descripción de las Actividades y Prácticas de ChildProgramming-CT4E por sus Dimensiones .....  | 64 |
| Tabla 15. Identificación y formulación del problema.....   | 65 |
| Tabla 16. Actividad de formulación de soluciones.....  | 66 |
| Tabla 17. Actividad de evaluación por prototipos .....   | 66 |
| Tabla 18. Actividad de entendimiento compartido.....   | 67 |
| Tabla 19. Actividad evaluación CT (Pre-Test) .....   | 69 |
| Tabla 20. Actividad evaluación CT (Pos-Test) .....   | 70 |
| Tabla 21. Actividad de observación .....   | 70 |
| Tabla 22. Distribución de estudiantes FIET .....   | 79 |
| Tabla 23. Distribución de las preguntas según las variables a medir en el Pre-Test .....   | 82 |
| Tabla 24. Distribución de las preguntas según las variables a medir en el Pos-test.....  | 83 |
| Tabla 25. Nivel de Competencia para cada aspecto del pensamiento computacional.....  | 84 |
| Tabla 26. Capacidades evaluadas por las preguntas del Test[65] .....   | 85 |
| Tabla 27. Sesión 1 Presentación, aplicación de protocolo de observación y entrevista al docente en clase de Introducción a la Ingeniería .....                 | 86 |
| Tabla 28. Sesión 2 Pre-Test y Actividad “Construyendo tu primer programa en Scratch” .....   | 87 |
| Tabla 29. Sesión 3 Actividad Desconectada “Red de clasificación de seis entradas”.....   | 89 |
| Tabla 30. Sesión 4 Pos-Test y Actividad de entendimiento compartido .....  | 91 |
| Tabla 31. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación. Grupo A Ingeniería Electrónica y Telecomunicaciones ..... | 93 |
| Tabla 32. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación. Grupo B Ingeniería Electrónica y Telecomunicaciones ..... | 93 |
| Tabla 33. Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación. Grupo de Ingeniería Automática Industrial .....           | 94 |

|  |     |
|--|-----|
| Tabla 34 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo A Ingeniería Electrónica y Telecomunicaciones ..... | 95  |
| Tabla 35 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo B Ingeniería Electrónica y Telecomunicaciones ..... | 95  |
| Tabla 36 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo de Ingeniería Automática Industrial .....           | 95  |
| Tabla 37 Nivel de Competencia para cada aspecto del pensamiento computacional.....   | 96  |
| Tabla 38 Habilidades de Pensamiento Computacional, según son descritas por la herramienta Dr. Scratch y su aplicación al fase exploratoria .....                 | 98  |
| Tabla 39. Actividad Evaluación CT (Pre-Test).....  | 99  |
| Tabla 40. Actividad de Scratch Unidad 1: Direcciones .....   | 100 |
| Tabla 41. Actividad de Scratch Unidad 2: Bucles.....   | 101 |
| Tabla 42. Identificación y Formulación del Problema .....  | 101 |
| Tabla 43. Actividad de Scratch Unidad 3: Condicionales .....   | 102 |
| Tabla 44. Actividad de Formulación de Soluciones.....  | 103 |
| Tabla 45. Actividad de Scratch Unidad 4: Funciones .....   | 104 |
| Tabla 46. Actividad de Evaluación por Prototipos .....   | 104 |
| Tabla 47. Actividad Evaluación CT (Pos-Test) .....   | 105 |
| Tabla 48. Actividad Evaluación CT (Pos-Test).....  | 107 |
| Tabla 49. Preguntas relacionadas con la tarea requerida .....  | 108 |
| Tabla 50. Preguntas relacionadas con la capacidad evaluada.....  | 110 |

# Capítulo 1

## Introducción

El Pensamiento Computacional (Computational Thinking en Inglés), en adelante CT<sup>1</sup>, es un concepto que se viene utilizando en el contexto educativo desde hace varias décadas. Pero el concepto se ha popularizado en los últimos años con el trabajo de Jeannette M. Wing (2006) quien aborda su definición como "*una forma de resolver problemas de ingeniería modernos y de diseñar sistemas complejos que se basa en conceptos fundamentales de la informática*" [1], y lo promueve como "*una habilidad fundamental utilizada por todos en el mundo a mediados del siglo XXI*" [1].

El CT se ha integrado eficazmente en los cursos de historia, matemáticas, arte y ciencias. Sin embargo, han habido muy pocos esfuerzos para integrar el CT en la educación de Tecnología e Ingeniería (T&E<sup>2</sup>), lo cual contrasta con las grandes oportunidades que ofrece para involucrar a los estudiantes en las prácticas de CT en el contexto de auténticos desafíos tecnológicos [2]. Aunque el CT se ha discutido y debatido durante mucho tiempo, la literatura sobre el CT está todavía en una etapa temprana de madurez, especialmente sobre el cómo desarrollar el CT en el contexto de la ingeniería [2]. Gran parte de la literatura se centra en aplicar el CT a las disciplinas, especialmente en los campos de la educación STEM (Acrónimo de Science, Technology, Engineering and Mathematics). El CT es indispensable en la ingeniería moderna, y el CT podría obviamente desarrollar y mejorar la competitividad académica de los estudiantes, las habilidades de resolución de problemas complejos y el diseño de sistemas mediante la exploración y la aplicación de enfoques computacionales. La integración del CT en los programas de ingeniería ofrece un rico entorno para la investigación académica [3].

Así mismo, las competencias STEM y su integración en el contexto educativo, son un marco interesante para adquirir o desarrollar habilidades de CT y su aplicación pretende generar niveles de aprehensión de conocimiento, realmente significativos en los estudiantes, logrando un alto nivel de desarrollo cognitivo [4].

En el área del desarrollo del CT, el grupo IDIS<sup>3</sup> ha venido investigando en la formulación de enfoques que integran aspectos cognitivos, el aprendizaje colaborativo, la lúdica, entre otros. Esto ha dado origen a la metodología ChildProgramming. Aunque el concepto de CT no aparecía de forma explícita en el modelo original de Hurtado et al. (2012) [5], sí ha hecho parte de trabajos posteriores.

---

<sup>1</sup> CT: Computational Thinking.

<sup>2</sup> T&E: Technology and Engineering education.

<sup>3</sup> IDIS: Investigación y Desarrollo en Ingeniería de Software. <http://www.unicauca.edu.co/idis/>

En este sentido, el presente trabajo propone un conjunto de mejoras al actual modelo ChildProgramming, relacionadas con la integración de las competencias STEM y el desarrollo de las habilidades de CT, dicha investigación está orientada mediante las fases del ciclo de investigación científica definida por Hurtado [6], donde se integra lineamientos generales de Bunge [7] y particulares de Shaw [8]. Destacando tres fases principales las cuales son: exploración, formulación y evaluación. Así mismo, un estudio de caso descriptivo para lo cual se siguió la guía de Runeson *et al.* [9], para establecer las necesidades desde las interacciones de los equipos, así como la evaluación de la efectividad del proceso resultante. Para el desarrollo del estudio de caso se utilizó la herramienta SCRATCH y participaron los estudiantes de primer semestre de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial de la Universidad del Cauca.

## 1.1 Planteamiento del problema

Para Jeannette Wing (2006) [1] el CT será una habilidad fundamental utilizada por todos en el mundo a mediados del siglo XXI. Este tipo de pensamiento representa una actitud y un conjunto de habilidades de aplicación universal que todas las personas, no sólo los informáticos, estarían en capacidad de aprender y utilizar. De igual manera refiere que, “CT es el proceso de pensamiento involucrado en la formulación de un problema y la expresión de su(s) solución(es), de tal manera que un agente - humano o máquina - pueda ejecutarla efectivamente” [10]. Más concretamente, incluye la extracción de información clave de los detalles de un problema (abstracción), reformular un problema mayor en un conjunto de problemas más pequeños (descomposición), la identificación de elementos recurrentes entre las soluciones a problemas y en los datos (reconocimiento de patrones) y, el desarrollo y la aplicación de algoritmos (automatización) [11].

En los últimos años, está surgiendo una tendencia de integrar la informática (por tanto, el CT) en la educación disciplinaria, especialmente en los campos de la Ciencia, Tecnología, Ingeniería y Matemáticas (STEM<sup>4</sup>) [11]. La investigación ha demostrado que los contenidos y contextos STEM pueden beneficiar el aprendizaje del CT. Asimismo, la incorporación del CT en la educación STEM también mejora el aprendizaje de los contenidos de STEM por parte de los estudiantes [12]. Aunque el CT se ha discutido y debatido durante mucho tiempo, la literatura sobre el CT está todavía en una etapa temprana de madurez [13]. Especialmente sobre el cómo desarrollar el CT en el contexto de la ingeniería (T&E) sigue siendo un reto [2]. La integración del CT en los programas de ingeniería (T&E) ofrece un rico entorno para la investigación académica [3].

---

<sup>4</sup> STEM: Science, Technology, Engineering and Mathematics.

Hay evidencias de que el CT es indispensable en la ingeniería (T&E) moderna, este tipo de pensamiento podría desarrollar y mejorar la competitividad académica de los estudiantes, las habilidades de resolución de problemas, la comprensión de los sistemas complejos y el diseño de sistemas mediante la exploración y la aplicación de enfoques computacionales [3]. Sin embargo, han habido muy pocos esfuerzos para integrar el CT en la educación en Ingeniería (T&E), a pesar de las grandes oportunidades que ofrece para involucrar a los estudiantes en las prácticas de CT en el contexto de auténticos desafíos tecnológicos [2]. Por otra parte, Mohtadi et al. (2013) [14] establecieron que los estudiantes que ingresan en los cursos de ingeniería (T&E) tienen menos competencias en las materias de ciencia, tecnología, ingeniería y matemáticas (STEM), a pesar de que han crecido en un entorno con todo tipo de dispositivos informáticos y electrónicos. Esto se presenta debido a los cambios recientes en la enseñanza secundaria que han provocado un descenso en las habilidades y conocimientos subyacentes de los estudiantes en las áreas STEM. Lo que hace necesario impulsar la investigación sobre la integración del CT en la Ingeniería (T&E) para desarrollar habilidades específicas en los estudiantes, tales como la solución de problemas y el CT en general.

En el área del desarrollo del CT, el grupo IDIS ha venido investigando en la formulación de enfoques que integran aspectos cognitivos, el aprendizaje colaborativo, la lúdica, entre otros. Esto ha dado origen a la metodología ChildProgramming. El concepto de CT ha hecho parte de trabajos tales como el de Chimunja et al. (2017) [15], llamado ChildProgramming-C, y el estudio de Zúñiga et al.(2019) [16]. El modelo se ha robustecido a través de algunas extensiones tales como: ChildProgramming-C, dirigida a satisfacer la dimensión colaborativa del modelo, ChildProgramming-G [17], enfocada en la gamificación como forma de incorporar el aspecto lúdico al mismo. Child Gender, encaminada a extender ChildProgramming con un aspecto de diversidad de género [18]. ChildProgramming-MT, dirigida a estudiar el modelo para encontrar elementos que permitan desarrollar un sistema de Memoria Transactiva [19], ChildProgramming Evolution, se hace una revisión general de la evolución de la metodología ChildProgramming [20]. ChildProgramming-RE, enfocada a incentivar la participación y motivación hacia el desarrollo del CT en niños entre 10 y 12 años, a partir de los conceptos de STEM relacionados con la Robótica Educativa [21]. ChildDebugging, Se extraen las principales estrategias y recursos utilizados por los equipos de niños para depurar programas basados en el paradigma de bloques [22]. Block-Based Design, propuesta basada en la aplicación de un método para el diseño de programas basados en bloques, utilizando mecanismos de abstracción, propios del pensamiento computacional, para facilitar a los niños la construcción, descomposición e integración de nuevos bloques [23]. Los citados resaltan, como punto en común, la importancia y necesidad de fomentar la motivación en el proceso de aprendizaje, y se evidencia la carencia de material educativo que facilite la comprensión de conceptos computacionales.

Sobre la base de la problemática expuesta sobre la necesidad de integración práctica de los conceptos CT y STEM en ingeniería (T&E), y considerando los esfuerzos previos del grupo

IDIS en el contexto del modelo ChildProgramming, se ha formulado la siguiente pregunta de investigación: ¿cómo Integrar el Desarrollo del Pensamiento Computacional en las actividades formativas en Ingeniería (T&E) en el primer semestre de la carrera, junto a los conceptos de STEM, en el contexto del enfoque metodológico planteado por ChildProgramming?

## **1.2 Objetivos**

### **1.2.1 Objetivo general.**

Construir un modelo que permita integrar el Desarrollo del Pensamiento Computacional en las actividades formativas en Ingeniería (T&E) en los primeros semestres de la carrera, junto a los conceptos de STEM, en el contexto del enfoque metodológico de ChildProgramming, para producir un incremento significativo en la apropiación de los conceptos del CT en los estudiantes de ingeniería.

### **1.2.2 Objetivos específicos.**

- Caracterizar a partir de estudios previos, las relaciones entre los aspectos de STEM, particularmente de Ingeniería y los aspectos involucrados en el desarrollo del pensamiento computacional.
- Diseñar e instrumentalizar un modelo metodológico, a través de la incorporación de los conceptos de STEM, en el enfoque metodológico de ChildProgramming con el fin de integrar el desarrollo del Pensamiento computacional (CT) en las actividades formativas de los estudiantes de primer semestre de carreras de Ingeniería en Telecomunicaciones.
- Evaluar el efecto del modelo en la asimilación de los conceptos propios del área disciplinar de Ingeniería (T&E), así como los propios del desarrollo del Pensamiento Computacional (CT) (abstracción, descomposición, reconocimiento de patrones y diseño de algoritmos) por parte de estudiantes de primer semestre de carreras de ingeniería en telecomunicaciones, a través de un estudio de caso comparativo.

## 1.3 Metodología

El desarrollo de este proyecto se efectuará en 3 fases fundamentadas en los lineamientos del ciclo de investigación científica definida por Hurtado [6], donde se integra lineamientos generales de Bunge [7] y particulares de Shaw [8]. Destacando tres fases principales las cuales son: exploración, formulación y evaluación. Siguiendo este orden inicialmente se realiza una revisión de la literatura existente, en la cual se formula el problema y se plantean las hipótesis. A continuación, se diseña e instrumentaliza un modelo metodológico, a través de la incorporación de los conceptos de STEM, en el enfoque metodológico de ChildProgramming con el fin de integrar el desarrollo del CT en las actividades formativas de los estudiantes de primer semestre de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial. Por último, se evalúa el efecto del modelo en la asimilación de los conceptos propios del área disciplinar de Ingeniería (T&E), así como los propios del desarrollo del Pensamiento Computacional (CT) por parte de los estudiantes de primer semestre de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial, a través de un estudio de caso descriptivo. Esta metodología propone exponer la organización y el cumplimiento de los objetivos expuestos anteriormente. Siendo apoyados en metodologías de investigación determinadas descritas en cada fase.

### 1.3.1 Fase de exploración.

La finalidad de esta fase es buscar la información bibliográfica existente sobre el contexto de STEM, pensamiento computacional (CT), ChildProgramming, las herramientas, posibles modelos de integración y sistemas de evaluación del CT. Esto incluye consultar en fuentes relevantes, la integración del desarrollo del pensamiento computacional en las actividades formativas en Ingeniería (T&E), incluyendo una exploración de las herramientas software y hardware elegidas para el proyecto (equipo de cómputo, lenguajes de programación, etc.). Para esto se conducirá una revisión de la literatura utilizando el diagrama de flujo de cuatro fases de PRISMA<sup>5</sup>[24], y adaptando la guía diseñada por Turner *et al.* [25]. Además, se diseñará y ejecutará un estudio de caso descriptivo para lo cual se seguirá la guía de Runeson *et al.* [9].

### 1.3.2 Fase de construcción del modelo.

En esta fase se crea el modelo para integrar los conceptos de STEM, en el contexto del modelo metodológico de ChildProgramming con el fin de integrar el desarrollo del CT, siguiendo el enfoque de ingeniería de método situacional propuesto por Mirbel y Ralyté [26] lo cual incluye

---

<sup>5</sup> PRISMA: Preferred Reporting Items for Systematic reviews and Meta-Analyses: <http://www.prisma-statement.org/>

capturar los requerimientos del método, seleccionar componentes de método y ensamblar los componentes del método.

### **1.3.3 Fase de evaluación.**

El objetivo de esta fase es evaluar el efecto del modelo en la asimilación de los conceptos propios del área disciplinar de Ingeniería (T&E), así como los propios del desarrollo del Pensamiento Computacional (CT) a través del diseño y ejecución de un estudio de caso descriptivo, siguiendo la guía de Runeson *et al.* [9].

## **1.4 Aportes**

### **1.4.1 En el ámbito académico.**

La presente monografía ofrece una experiencia de referencia sobre un modelo de integración del desarrollo del CT en las actividades formativas en Ingeniería (T&E) junto a los conceptos de STEM, en el enfoque metodológico de ChildProgramming (Pensamiento computacional para la formación de ingenieros en el contexto STEM (“COMPUTATIONAL THINKING FOR ENGINEERING”, en adelante CT4E)), con el fin de incrementar las habilidades en los estudiantes, en cuanto al desarrollo de sus capacidades de CT, basada en estudios de caso, la cual puede ser utilizada como contribución académica en la Facultad de Ingeniería Electrónica y Telecomunicaciones. El modelo puede ser usado para fortalecer el curso de introducción a la ingeniería en programas de ingeniería y afines a través de procesos, métodos y prácticas con el fin de incrementar su aplicabilidad.

### **1.4.2 En la investigación.**

Existen variadas metodologías y modelos para el desarrollo de habilidades de pensamiento computacional, pero estas no están asociadas a los conceptos STEM en el enfoque metodológico de ChildProgramming, esta metodología surge como un modelo de construcción de software en niños de edad escolar con el que se busca desarrollarles habilidades colaborativas, comunicativas y cognitivas. El modelo CT4E busca integrar el desarrollo del CT en las actividades formativas en Ingeniería (T&E) en los primeros semestres de la carrera, junto a los conceptos de STEM, en el contexto del enfoque metodológico de ChildProgramming. El aporte a la comunidad científica es un estudio investigativo acerca de la evidencia empírica sobre el valor que tiene la integración, para mejorar el desarrollo de aquellas habilidades, tanto desde el punto de vista teórico (estado del arte), como empírico (estudio de caso descriptivo) y, particularmente, enriquecerá la base de conocimiento alrededor del CT desarrollada por el grupo IDIS desde el año 2011.

Dicho modelo se evaluará a través de un estudio de caso descriptivo, con un alcance de aplicación establecido, contribuyendo con un estudio replicable que sirva de base experimental para que la comunidad del CT plantee nuevas hipótesis y contraste otras propuestas y resultados. Se espera, en consecuencia, que los resultados del estudio de caso descriptivo sean uno de los principales aportes de esta investigación.

## **1.5 Resultados obtenidos**

Monografía de grado: este documento incluye una base conceptual, la descripción del modelo construido, así como los resultados de la evaluación del estudio de caso descriptivo, aportes, conclusiones y recomendaciones para el desarrollo de futuras investigaciones.

Anexos: este documento contiene aportes necesarios y complementarios del proyecto, no incluidos en la monografía.

Artículo: el artículo “Integrando el Desarrollo del Pensamiento Computacional y las Competencias STEM en la Educación de Tecnología e Ingeniería(T&E). Una revisión de la literatura” se hizo con base en el Capítulo 2 de esta monografía.

## **1.6 Organización del documento**

El presente documento se encuentra organizado por capítulos de la siguiente manera:

Capítulo 1. Introducción.

Capítulo 2. Marco teórico y revisión de la literatura.

Capítulo 3. Modelo CT4E: Pensamiento computacional para la formación de ingenieros en el contexto STEM.

Capítulo 4. Planeación, diseño, ejecución y resultados del estudio.

Capítulo 5. Conclusiones y trabajos futuros.

# Capítulo 2

## Marco teórico y revisión de la literatura

### 2.1 Marco teórico

#### 2.1.1 Pensamiento computacional (CT).

El término de Pensamiento Computacional (CT) se mencionó inicialmente en las década de 1980 en el trabajo de Seymour Papert con el lenguaje de programación LOGO y el desarrollo del pensamiento procedimental en niños a través de la programación [27], [28]. Esto puso de manifiesto la gran influencia que tendrían las computadoras en la vida de las personas, siempre y cuando se le dé una utilización más allá de la diversión, por ello puntualiza: “los niños deben programar la computadora en lugar de ser programados por ella” [29]. Después, en el 2006, Jeannette Wing plantea que el CT es “el proceso de pensamiento envuelto en formular un problema y sus soluciones de manera que esas son representadas de una forma en que pueden ser ejecutadas por un agente de procesamiento de información”. Para referirse a una habilidad fundamental que, junto con la lectura, la escritura y la aritmética, debería hacer parte de la capacidad analítica a desarrollar por todos las personas [1]. En este planteamiento inicial, ella sostiene que el CT involucra la solución de problemas, el diseño de sistemas y la comprensión del comportamiento humano con base en los conceptos fundamentales de la informática, y comprende - pero no se limita a – la reformulación de problemas, el pensamiento recursivo, el procesamiento paralelo, la interpretación de código como información y viceversa, la abstracción, la descomposición, la modularización, y el razonamiento heurístico [1]. Este planteamiento dio origen a múltiples intentos para alcanzar una definición concreta y certera de lo que es el CT, Sin embargo, todavía hay poco consenso sobre una definición formal del CT [30], [31]. Por otro lado, Selby y Woollard (2013) [30], plantean que para algunos autores la importancia radica en cómo se enseña el CT y cómo éste se evidencia en los aprendices, para lo cual solo es necesario una definición amplia de CT; en contraparte, para otros, es necesaria una definición concreta y certera , sin la cual se hace difícil desarrollar herramientas evaluativas que midan la capacidad de pensar computacionalmente, sin olvidar que, como sostienen Haseski et al. (2018) [32], “la definición de conceptos, fenómenos o eventos es crucial para su comprensión y en la determinación de los límites del tema”.

Posteriormente, en el año 2010 Wing actualiza el concepto dándole mayor claridad, para visualizarlo como un marco para el pensamiento y, más precisamente, redefiniéndolo como “Pensamiento Computacional son los procesos de pensamiento implicados en la formulación de problemas y sus soluciones para que estas últimas estén representadas de forma que

puedan llevarse a cabo de manera efectiva por un procesador de información” [10]. Destacando la abstracción como su esencia y resaltando, al igual que otros investigadores, que no se trata solamente de solucionar un problema, sino también de formularlo [10].

Después de que, Selby y Woollard (2013) [30] realizaran una revisión de la literatura existente descubrieron que, en relación al CT, existen tres términos que aparecen con mucha frecuencia: Proceso de pensamiento, abstracción y descomposición. Sobre ellos, pareciera haber un acuerdo para que hagan parte de la definición. Para Selby y Woollard, el Pensamiento Computacional es una actividad enfocada en, más no restringida a, la solución de problemas, que incorpora procesos de pensamiento que involucran abstracción, descomposición, diseño algorítmico, evaluación (de tiempo y espacio, de potencia y almacenamiento) y generalización [30].

Por otra parte, en 2011 La Sociedad Internacional para la Tecnología en la Educación (ISTE<sup>6</sup>) y la Sociedad de Maestros de Ciencias de la Computación (CSTA<sup>7</sup>) de los Estados Unidos, desarrollaron una definición operativa del CT que sirve de marco de trabajo y vocabulario compartido para los profesores de informática en las etapas de educación secundaria y preuniversitaria estadounidense [14]. Una definición muy similar a la que aparece también en Google for Education [33], según la cual, el CT es un proceso de solución de problemas que incluye un conjunto de características y actitudes. Entre las habilidades involucradas en el CT se cuentan: formular problemas, de forma que permita emplear una computadora u otra herramienta para ayudar a resolverlos; organizar y analizar datos de manera lógica; representar datos mediante abstracciones, tales como modelos y simulaciones; automatizar soluciones a través de pensamiento algorítmico (una serie de pasos ordenados); identificar, analizar e implementar posibles soluciones con el fin de alcanzar la más eficiente y efectiva combinación de pasos y recursos; generalizar y transferir este proceso de solución a una amplia variedad de problemas.

Las anteriores destrezas se soportan y mejoran por actitudes como: confianza para enfrentar la complejidad; persistencia al trabajar con problemas difíciles; tolerancia a la ambigüedad; habilidad para lidiar con problemas de final abierto, y habilidad para comunicarse y trabajar con otros para alcanzar una solución o meta común [33].

Siguiendo a la CSTA & ISTE (2011), el CT es un proceso de solución de problemas que incluye (aunque no está limitado a) las siguientes habilidades: [21].

- Abstracción: identificar y extraer información relevante para definir la(s) idea(s) principal(es).

---

<sup>6</sup> ISTE: International Society for Technology in Education: <https://www.iste.org/>

<sup>7</sup> CSTA: Computer Science Teachers Association: <https://www.csteachers.org/>

- Diseño de algoritmos: crear una serie ordenada de instrucciones para resolver problemas similares o para realizar una tarea.
- Automatización: hacer que computadoras u otras máquinas realicen tareas repetitivas.
- Análisis de datos: lograr que los datos tengan sentido al encontrar patrones o indicios.
- Recolección de datos: recolectar información.
- Representación de datos: representar y organizar información en gráficas, tablas, palabras o imágenes adecuadas.
- Descomposición: separar los datos, procesos o problemas en partes más pequeñas y manejables.
- Paralelismo: procesamiento simultáneo de tareas más pequeñas a partir de una más grande para alcanzar una meta común de forma más eficiente.
- Generalización: creación de modelos, reglas, principios o teorías a partir de patrones observados para evaluar los resultados previstos.
- Simulación: desarrollo de un modelo para imitar procesos del mundo real [21].

Así mismo, Brennan & Resnick (2012) [34] proponen un marco conceptual que analiza el desarrollo del pensamiento computacional desde tres dimensiones clave: conceptos, prácticas y perspectivas computacionales. Ver **Figura 1**. Que a su vez las relacionan con el lenguaje de programación visual Scratch<sup>8</sup>. Estas dimensiones están estrechamente ligada a la resolución de problemas de diferentes tipologías [34].

**Figura 1. Descripción del marco conceptual del Pensamiento Computacional**



**Fuente. Propia de la investigación**

- **Conceptos computacionales:** son aquellos conceptos que emplean los diseñadores de programas al elaborar sus aplicaciones. Su aprendizaje les permite a los estudiantes

<sup>8</sup> Scratch: <https://scratch.mit.edu/>

conocer bloques operativos, que se emplean al momento de programar, los cuales, a su vez, son comunes en varios lenguajes de programación, y así mismo, pueden aplicarse en diversos ámbitos, en este caso la resolución de problemas [34]. Ver **Tabla 1**.

**Tabla 1. Conceptos computacionales**

| Conceptos       | Definición  |
|-----------------|---|
| Secuencias      | Serie de pasos que debe realizar un objeto para generar un comportamiento específico  |
| Ciclos (Bucles) | Comportamiento que se repite un número determinado, o indeterminado de veces.   |
| Paralelismo     | Configuración de los diferentes objetos que deben cumplir diferentes funciones o realizar diferentes comportamientos al mismo tiempo.                 |
| Eventos         | Determinación del momento en que una acción o comportamiento debe ser realizado, por ejemplo, la aparición de un objeto, un cambio de escenario, etc. |
| Condicionales   | Procesos que dependen de decisiones basadas en el cumplimiento, o no cumplimiento, de una condición.  |
| Operadores      | Procesos matemáticos en el tratamiento de datos.  |
| Datos           | Almacenamiento y actualización de valores y variables.  |

**Fuente. Propia de la investigación**

- **Prácticas computacionales:** son aquellas prácticas o procedimientos que suelen emplear los diseñadores y creadores de programas al elaborar sus aplicaciones. Se relacionan con los procesos y las prácticas que los estudiantes utilizan en la medida en que van programando, es decir, son aquellos procesos que emplean para construir los proyectos, estos procesos se enfocan en el pensar y el aprender, y cómo está aprendiendo [34]. Ver **Tabla 2**.

**Tabla 2. Prácticas computacionales [34]**

| Prácticas                   | Definición   |
|-----------------------------|--|
| Ser incremental e iterativo | Hace referencia a las consideraciones iniciales en el diseño de proyectos, y como estas se ven alteradas en el transcurso de este.<br>Diseñar un proyecto no es un proceso limpio y secuencial en el que primero se identifica un concepto para un proyecto, luego se desarrolla un plan para el diseño y después se implementa el diseño en código. Es un proceso adaptativo, en el que el plan puede cambiar en respuesta a la búsqueda de una solución en pequeños pasos. |
| Ensayar y depurar           | Consiste en las técnicas adquiridas por los estudiantes a través de la experiencia y el ensayo error, y como las mismas se transforman en estrategias a usar en diferentes problemas.<br>Las cosas rara vez (o nunca) funcionan tal y como se imaginan; es fundamental que los diseñadores desarrollen estrategias para hacer frente a los problemas y anticiparse a ellos.  |
| Reutilizar y remezclar      | Se encuentra enfocada a la utilización de secuencias e ideas diseñadas por pares, integrándolas en proyectos en desarrollo, tanto como para complejizarlos en la búsqueda de un objetivo, como para fomentar la interpretación en la lectura de códigos.   |

|                        |  |
|------------------------|--|
|                        | Basarse en el trabajo de otros es una práctica que viene de lejos en la programación y que se ha visto ampliada por las tecnologías de red, que proporcionan acceso a una amplia gama de trabajos ajenos para reutilizar y recombinar. Esto les permite a los niños, crear cosas mucho más complejas de las que podrían haber creado por sí solos. La reutilización y la recombinación favorecen el desarrollo de la capacidad crítica de lectura del código |
| Abstraer y modularizar | Señala la categorización de las secuencias realizadas en relación con su función, evidenciando como los estudiantes clasifican diferentes módulos, al igual que reconociéndolos en diferentes objetos.<br>Abstraer y modularizar, lo que caracterizamos como construir algo grande juntando colecciones de partes más pequeñas, es una práctica importante para todo diseño y resolución de problemas.   |

Fuente. Propia de la investigación

- **Perspectivas computacionales:** son aquellas perspectivas que generalmente los diseñadores y creadores de programas construyen sobre el mundo que les rodea. Permiten mejorar la percepción que se tiene sobre determinados temas, también la forma en que vemos o consideramos las cosas, ya sea a nivel personal o dirigido a la comunidad [34]. Ver **Tabla 3**.

Tabla 3. Perspectivas computacionales

| Perspectivas | Definición  |
|--------------|---|
| Expresar     | Expone como la computación pasa de ser algo consumible a un medio para mostrar ideas propias por medio del diseño.<br>Un pensador computacional ve la computación como algo más que un producto para consumir; la computación se puede utilizar para el diseño y la autoexpresión.<br>Un pensador computacional ve la computación como un medio y piensa: "Puedo crear" y "Puedo expresar mis ideas a través de este nuevo medio".  |
| Conectar     | Se encuentra relacionada con la comunicación y el acceso al modo creativo de otros diseñadores, de manera indirecta observando los algoritmos realizados, o pidiendo opiniones y colaboraciones de forma particular, reconociendo la importancia de la creación con y para los demás.<br>La creatividad y el aprendizaje son prácticas profundamente sociales, por lo que el diseño de medios informáticos se ve enriquecido por la interacción con los demás.  |
| Preguntar    | Señala el contraste entre la regulación que las tecnologías realizan en la sociedad y la influencia que tienen la personas para influir en ellas, por lo que desarrollar una perspectiva interrogativa hacia cómo y qué se puede cambiar, por medio de los conceptos aprendidos por medio de la elaboración de proyectos, fomentaría el fortalecimiento de un pensamiento computacional.<br>Con la perspectiva computacional del cuestionamiento, buscamos indicadores de que los jóvenes no sienten esta desconexión entre las tecnologías que les rodean y sus capacidades para negociar las realidades del mundo tecnológico. Los jóvenes deberían sentirse capacitados para hacer preguntas sobre y con la tecnología: "Puedo (utilizar la computación para) hacer preguntas para dar sentido a (las cosas computacionales en) el mundo". |

Fuente. Propia de la investigación

### 2.1.2 Educación STEM.

El término STEM fue acuñado por Judith Ramaley cuando era directora de la NSF<sup>9</sup> (National Science Foundation) en 2001 para sustituir a su predecesor SMET. El acrónimo SMET fue utilizado por primera vez en 1993 [35]. Este se dio a conocer al público en una serie de audiencias en el año 1997 [36]. La educación STEM es una integración de cuatro disciplinas descritas a continuación: "S" de Ciencia (Science) se preocupa mucho por lo que es el mundo natural y por preparar a los estudiantes para que piensen y actúen como verdaderos científicos, hagan preguntas, formulen hipótesis y lleven a cabo investigaciones utilizando prácticas científicas estándar [37]. Estos cursos tratan sobre física, biología, química, astronomía, etc. Por su parte, la Tecnología (Technology) "T" es la modificación del mundo natural para satisfacer los deseos y necesidades humanas. Se ocupa mucho de lo que puede y debe ser (diseñado, hecho y desarrollado) a partir de materiales y sustancias del mundo natural para satisfacer las necesidades y deseos humanos. Además, la Ingeniería (Engineering) "E" se define como una profesión en la que el conocimiento de las matemáticas y la naturaleza y el criterio de mejorar las formas de utilizar los materiales y las fuerzas de la naturaleza de forma económica en beneficio de la humanidad. Por su parte, las matemáticas (Mathematics) "M" se definen como la ciencia de los patrones y las relaciones que proporciona el lenguaje exacto para la tecnología, la ciencia y la ingeniería [38]. La comprensión de estas cuatro disciplinas se tiene en cuenta para el desarrollo de las naciones del siglo XXI. Por ello, muchos países intentan mejorar la calidad de la educación con determinadas materias STEM [39].

Kelley y Knowles (2016) [40] define la integración de STEM en educación, como “un esfuerzo que combina alguna o las cuatro disciplinas de ciencia, tecnología, ingeniería y matemáticas en una clase, o lección basada en las diferentes conexiones entre las diferentes materias y los problemas del mundo real” [40].

En general, la educación STEM se refiere a proporcionar oportunidades para que los estudiantes sean capaces de resolver problemas, ser innovadores, inventores, seguros de sí mismos, pensadores lógicos y tecnológicamente alfabetizados [41]. Así mismo la educación STEM se ha contemplado desde la década de 1990 y puede tener éxito en la educación, en la preparación de estudiantes competentes en todos los niveles con las habilidades en la sociedad científica en rápida expansión, especialmente para el éxito futuro de los estudiantes [37]. La educación STEM también tiene el potencial para aumentar la alfabetización de todas las personas en toda la población en áreas tecnológicas y científicas [42].

---

<sup>9</sup> NFS: National Science Foundation: [https:// www.nsf.gov/](https://www.nsf.gov/)

Sánchez Ludeña (2019) [4] afirma que a través de la Educación STEM, los estudiantes conseguirán desarrollar competencias claves como se muestra en la **Tabla 4**.

**Tabla 4. Competencias y Dimensiones STEM [4]**

| Competencias STEM   | Dimensiones  |
|---|--|
| <b>Autonomía y emprendimiento</b><br>Acometer y llevar adelante un proyecto o propósito por propia iniciativa   | Aprender a aprender<br>Autonomía y desarrollo personal<br>Emprendimiento                                     |
| <b>Colaboración y comunicación</b><br>Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento  | Expresión y comunicación<br>Trabajo colaborativo   |
| <b>Conocimiento y uso de la tecnología</b><br>Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso | Cultura tecnológica<br>Uso de productos tecnológicos   |
| <b>Creatividad e innovación</b><br>Resolver de forma original e imaginativa situaciones o problemas en un contexto dado   | Creatividad e innovación   |
| <b>Diseño y fabricación de productos</b><br>Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados  | Diseño<br>Fabricación<br>Planificación y gestión   |
| <b>Pensamiento crítico</b><br>Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos   | Pensamiento lógico<br>Pensamiento sistémico  |
| <b>Resolución de problemas</b><br>Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia  | Obtención y tratamiento de la información<br>Pensamiento computacional<br>Proceso de resolución de problemas |

Fuente Sánchez Ludeña (2019) [4]

Con estas competencias en STEM, se pretende desarrollar el emprendimiento, la innovación y la competitividad, básicos dentro de la economía actual, con el fin de mejorar la inserción en el mundo laboral de esta sociedad cada vez más tecnológica [43].

### 2.1.3 Enseñanza de tecnología e ingeniería (T&E).

La Asociación Internacional de Enseñanza de Tecnología e Ingeniería (ITEEA<sup>10</sup>) define la enseñanza de tecnología e ingeniería como "*la forma en que los seres humanos modifican el mundo que les rodea para satisfacer sus necesidades y deseos o para resolver problemas prácticos.*" Por lo tanto, la enseñanza de tecnología e ingeniería es el aprendizaje basado en problemas por parte de los estudiantes, que utilizan los principios de las matemáticas, las ciencias, la ingeniería y la tecnología. Según Dugger y Satchwell (2022) [44]. Estos estudios implican:

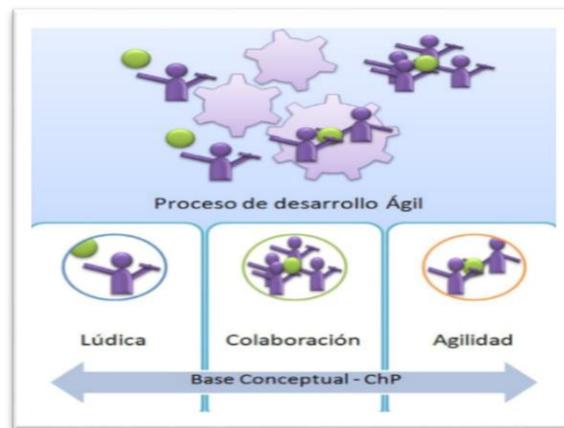
<sup>10</sup> ITEEA: International Technology and Engineering Education Association: <https://www.iteea.org/>

- Diseñar, desarrollar y utilizar sistemas tecnológicos;
- Actividades de diseño abiertas y basadas en problemas;
- Estrategias de aprendizaje cognitivas, manipulativas y eficaces
- Aplicar los conocimientos y procesos tecnológicos a experiencias del mundo real utilizando recursos actualizados;
- Trabajar individualmente y en equipo para resolver problemas [44].

#### 2.1.4 ChildProgramming.

En el área del desarrollo del CT, el grupo IDIS<sup>11</sup> ha venido investigando en la formulación de enfoques que integran aspectos cognitivos, el aprendizaje colaborativo, la lúdica, entre otros. Esto ha dado origen a la metodología ChildProgramming. En el modelo original de Hurtado et al. (2012) [5]. Se recrea un ambiente integral de desarrollo para niños como estrategia de aprendizaje y construcción de software desde una perspectiva basada en la lúdica, la colaboración y la agilidad. Según los autores, con este modelo se ofrece un espacio a los niños para desarrollar sus habilidades lógico-matemáticas y sociales, además se les ofrece a los alumnos la libertad, para facilitar que emerjan en este ambiente de desarrollo aportando a la misión de las instituciones educativas. Esta propuesta metodológica utiliza estrategias de trabajo colaborativo y aprovecha los enfoques ágiles para el desarrollo de software con fines de facilidad y de descubrimiento. La **Figura 2** presenta los principales componentes.

**Figura 2. Modelo inicial planteado de ChildProgramming**



**Fuente. Hurtado et al. (2012) [5]**

ChildProgramming está compuesto por tres dimensiones [15]: la dimensión cognitiva, considerada como el esfuerzo que realizará un niño por comprender, analizar y apropiarse situaciones presentes en las tareas definidas, aportando al proceso los conceptos principales para el desarrollo de las mismas. La dimensión ágil, basada en la promulgación de los valores

<sup>11</sup> IDIS: Investigación y Desarrollo en Ingeniería de Software. <http://www.unicauca.edu.co/idis/>

del manifiesto de las metodologías ágiles de desarrollo de software, esta dimensión aporta a ChildProgramming una forma acorde de trabajo para alcanzar los objetivos donde se evidencie un trabajo de equipo que permita a los integrantes del mismo permanecer juntos a lo largo de la actividad. El modelo ChildProgramming también propone la dimensión colaborativa, que intenta incrementar la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos mediante el desarrollo de software [15]. Ver **Figura 3**.

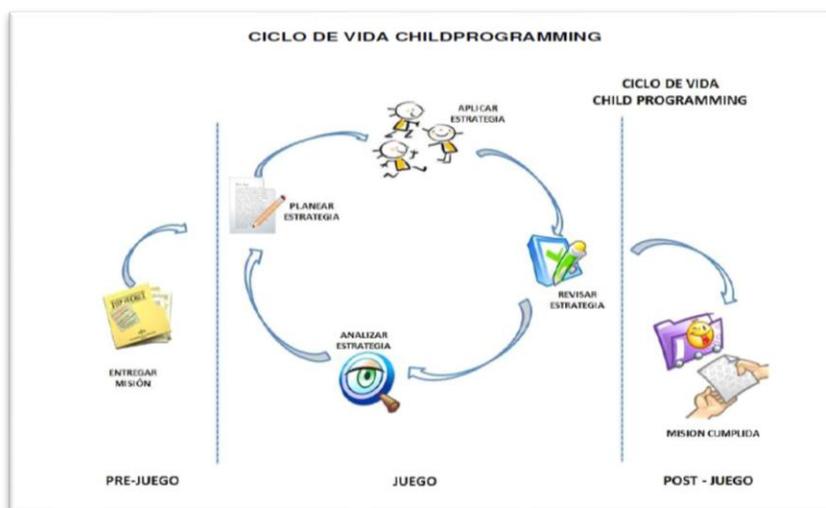
**Figura 3. Dimensiones de ChildProgramming [15]**



Fuente. Chimunja et al. (2017) [15]

ChildProgramming define tres grandes fases entre las cuales se tiene: el pre-juego, el juego y el pos-juego, así mismo propone como roles el profesor, el guía del equipo, el equipo de trabajo y un investigador. Este modelo se ha venido mejorando desde el año 2012 con el desarrollo de diferentes tesis de pregrado y posgrado, que ha abordado el componente cognitivo y el componente lúdico obteniendo un modelo gamificado con mecánicas y dinámicas de juego [17]. En la **Figura 4** se muestra el ciclo de vida del modelo ChildProgramming [22].

**Figura 4. Ciclo de vida ChildProgramming**



Fuente. <http://www.unicauca.edu.co/childprogramming/bitacora/Proceso>

También se extrajo un catálogo de Elementos de ChildProgramming, donde se evidenciaron roles (internos y externos), prácticas y conceptos ágiles y colaborativos, ambiente de trabajo, aspectos humanos y ambientales, para finalizar con un estudio de caso tres en el que se evaluó el modelo, para el desarrollo de esta actividad se trabajó con un grupo experimental y otro grupo de control (que trabajaba de manera intuitiva), en esta actividad computacional, se enfrentaban al lenguaje de programación ya con un esquema de trabajo, en donde tendrían que elaborar piezas funcionales de software, en esta actividad se pidió calcular índice de masa corporal de una persona [15].

ChildProgramming está centrada en el marco de trabajo propuesto por Scrum y basado en prácticas ágiles de desarrollo propuestas por la metodología Extreme Programming - XP. Esta forma de trabajo plantea, además, que estas prácticas las puedan desarrollar de forma divertida, como un juego para que los niños logren asimilar mejor los conceptos que se den dentro del proceso [5].

### **2.1.5 Scratch.**

Fue desarrollado por el “Lifelong Kindergarten group” en el Media Lab del Instituto Tecnológico de Massachusetts y la Universidad de California, los Ángeles. Este fue creado con fines educativos, donde se les permite explorar y experimentar conceptos de programación de computadoras mediante el uso de su sencilla interfaz gráfica, aprendiendo a programar de forma intuitiva por bloques. Desde que se liberó en 2007, el sitio web de Scratch tiene, a la fecha, alrededor de 99.935.948 usuarios y en él se pueden hallar más de 117.638.190 millones de diversos proyectos, la mayoría relacionados con juegos y animaciones sencillas. Es una herramienta que fomenta la creatividad de los niños, permitiendo explorar y experimentar con diferentes creaciones [45].

Es la comunidad de programación para niños y niñas más grande del mundo, y un lenguaje de programación con una interfaz sencilla que permite a los jóvenes crear historias digitales, juegos y animaciones. Está diseñado, desarrollado y moderado por la Fundación Scratch, una organización sin ánimo de lucro [45].

Scratch promueve el pensamiento computacional y las habilidades en resolución de problemas; enseñanza y aprendizaje creativos, auto expresión y colaboración; e igualdad en informática [45].

Está basado en el lenguaje de programación LOGO y sus características más importantes son:

- Está basado en bloques gráficos y la interfaz que tiene es muy sencilla e intuitiva.

- Tiene un entorno colaborativo mediante el cual se pueden compartir proyectos, scripts y personajes en la web.
- El trabajo se realiza mediante la unión de bloques que pueden ser eventos, movimientos de gráficos y sonidos.
- Los programas pueden ser ejecutados directamente sobre el navegador de Internet.

Las razones por las cuales se escogió esta herramienta para el desarrollo de las actividades y los casos de estudio son las siguientes [46]:

- Ya se había hecho un estudio previo de varias herramientas en el trabajo de ChildProgramming, del cual se había escogido Scratch para el desarrollo de las prácticas de ese trabajo.
- Está disponible para varios sistemas operativos, Windows, Mac, Linux y es multiplataforma y multilingüe (se puede utilizar con dispositivos digitales móviles: smartphones y tabletas).
- Es un programa gratuito, de software libre.
- Es perfecto para enseñar y aprender a programar a niños o adolescentes o a cualquier persona sin conocimientos informáticos de programación.
- Por su usabilidad y facilidad de uso, las estructuras de programación son mostradas al usuario como piezas de un rompecabezas y el niño programador puede arrastrarlas hacia la pantalla principal del programa de manera intuitiva simulando la formación de un rompecabezas y creando programas animados de manera sencilla.
- Su interfaz es sencilla para el usuario, y tiene disponible el lenguaje español para mayor comodidad de los niños.
- La cantidad de configuraciones que el niño puede realizar con las piezas esenciales es infinita y, de manera intuitiva, el niño va aprendiendo los conceptos de objetos y reusabilidad del software.
- Permite compartir los proyectos a través de Internet, pudiendo ser descargados y utilizados por otras personas.

Conviene enfatizar que Scratch no es una herramienta para enseñar a los estudiantes a programar con lenguajes de programación, sin embargo, les va a brindar conocimientos y conceptos de la manera como se trabaja en un entorno de programación real, siguiendo determinadas secuencias de órdenes [46].

## 2.2 Revisión de la literatura

### 2.2.1 Selección de los documentos.

A continuación, se expone como se realizó la revisión de la literatura utilizando el diagrama de flujo de cuatro fases de PRISMA [24], y adaptando la guía diseñada por Turner *et al.* [25]. Esta revisión se concentró en artículos y tesis que utilizan o proponen utilizar la integración de las competencias STEM como medio para el desarrollo de habilidades de Pensamiento Computacional, y se tuvo como insumos publicaciones que incluyeran los términos “STEM”, “STEM competences”, “Computational thinking”, “Engineering education”, “integrating” y “Skill”, o sus equivalentes en español, encontrados en las bases de datos ACM<sup>12</sup>, ASEE PEER<sup>13</sup>, ERIC<sup>14</sup>, IEEE Xplore<sup>15</sup>, ResearchGate<sup>16</sup>, Scopus<sup>17</sup>, SpringerLink<sup>18</sup> y SpringerOpen<sup>19</sup>. La búsqueda se realizó entre el 22 de septiembre y el 31 de octubre de 2022, empleando la cadena de búsqueda (“ENGINEERING EDUCATION” OR “EDUCACIÓN DE INGENIERÍA”) AND (“COMPUTATIONAL THINKING” OR “PENSAMIENTO COMPUTACIONAL”) AND (“STEM COMPETENCES” OR “STEM SKILLS”) AND (“INTEGRATING” OR “INTEGRACIÓN”)), se obtuvo como resultado 410 publicaciones.

En estas 410 publicaciones encontradas, se aplicaron los cuatro criterios de exclusión que se muestran a continuación:

- Criterio de exclusión 1 (C1): documentos que no fueron revisados. Los artículos encontrados no pudieron revisarse, sea porque el documento se encontraba en un idioma diferente de inglés o español, estaba incompleto, no se encontraba disponible en el momento de la consulta, la referencia no era válida u otra razón.
- Criterio de exclusión 2 (C2): documentos duplicados. Los artículos ya aparecen en alguna de las otras bases de datos consultadas y, por tanto, se trata de documentos duplicados.
- Criterio de exclusión 3 (C3): documentos no relacionados con el objeto de estudio. Aunque las expresiones referentes a las competencias STEM y Pensamiento Computacional aparecen en el documento, el artículo no se trata de STEM o no se trata

---

<sup>12</sup> ACM Digital Library - Association for Computing Machinery. <https://dl.acm.org/>

<sup>13</sup> ASEE PEER - American Society for Engineering Education. <https://peer.asee.org/>

<sup>14</sup> ERIC - Education Resources Information Center. <https://eric.ed.gov/>

<sup>15</sup> IEEEExplore - <https://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>16</sup> ResearchGate - <https://www.researchgate.net/>

<sup>17</sup> Scopus - <https://www.scopus-com/>

<sup>18</sup> SpringerLink - <https://link.springer.com/>

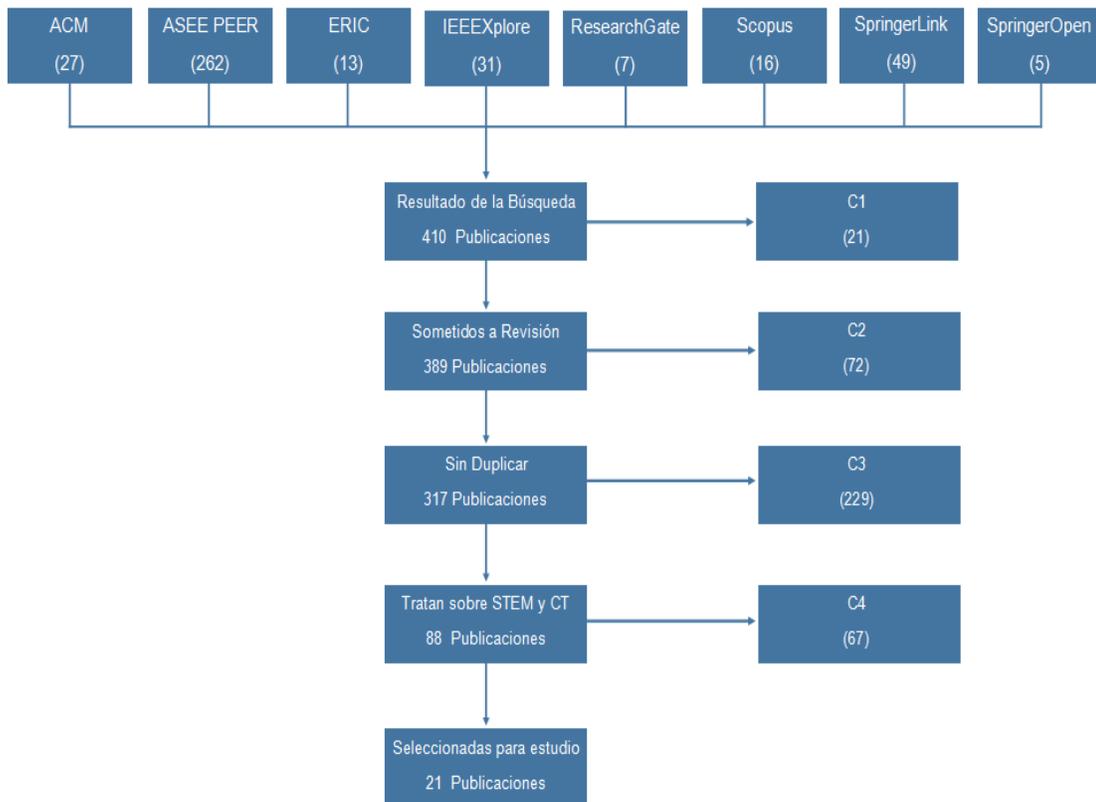
<sup>19</sup> SpringerOpen - <https://www.springeropen.com/>

de CT; es decir, al menos uno de ellos no es parte principal del mismo. En ocasiones, estas expresiones aparecen tan solo en la bibliografía.

- Criterio de exclusión 4 (C4): documentos fuera del alcance de este proyecto. La relación entre STEM y CT, es decir, la forma en que se integran las competencias STEM y como estas influyen en el desarrollo de las habilidades de CT, no es el tema central, no se establece o no se explica. Bajo este criterio también se agrupan artículos en los que los conceptos de CT divergen de los considerados para el presente proyecto, en algunos casos porque no se derivan de alguna definición conocida y concreta de él.

La **Figura 5** muestra el proceso de selección de las publicaciones, partiendo de los resultados de la búsqueda en las bases de datos y siguiendo con la aplicación de los criterios de exclusión.

**Figura 5. Proceso de selección de las publicaciones**



**Fuente. Propia de la investigación**

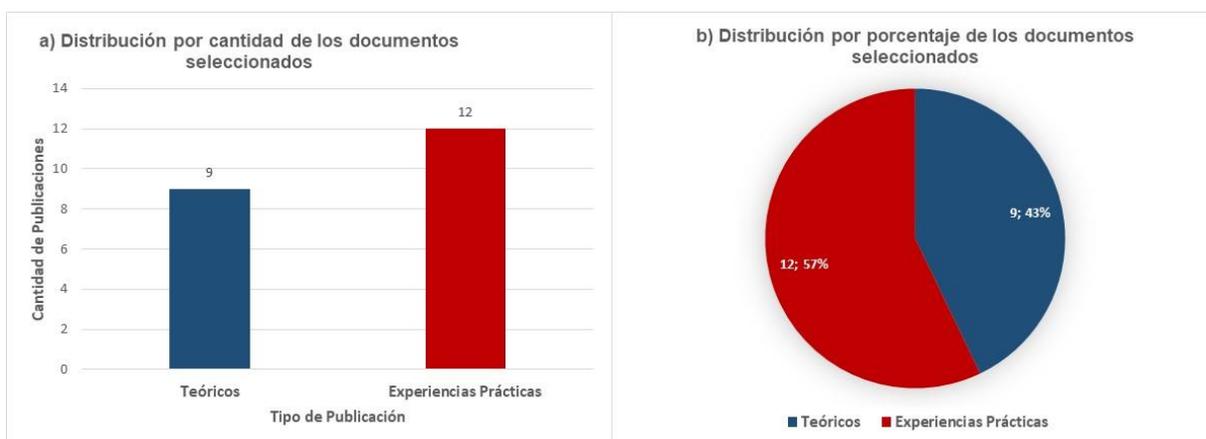
Después de ejecutar la búsqueda con la cadena indicada se obtuvieron 410 publicaciones. De ellas, 21 salieron con en el primer criterio de exclusión (C1), quedando 389 publicaciones sometidas a revisión. El segundo criterio de exclusión (C2) cubrió a 72 publicaciones,

obteniéndose 317 publicaciones únicas. El tercer criterio de exclusión (C3) abarcó 229 publicaciones, por lo que 88 tratan sobre STEM y CT. De estas últimas, 67 fueron eliminadas por el cuarto criterio de exclusión (C4), por lo que, finalmente, 21 fueron seleccionadas para este estudio. En el **Anexo A** se presenta una lista completa de éstas.

## 2.2.2 Distribución de los documentos seleccionados.

En general, y tras aplicar los criterios de inclusión y exclusión anteriormente expuestos, las publicaciones consideradas pueden clasificarse en dos grandes grupos: aquellas que tratan asuntos netamente teóricos, como las dedicadas a la revisión de literatura o la presentación de modelos, y aquellas que, solamente o en adición a lo anterior, reportan el resultado de experiencias prácticas. En el primer grupo se encontraron 9 publicaciones, correspondientes a un 43% del total, mientras que, en el segundo, las 12 publicaciones restantes, para un 57%. La distribución de los documentos seleccionados se observa en la **Figura 6**. En el **Anexo B** se presenta la clasificación y tipo de publicación.

**Figura 6. Distribución a) por cantidad y b) por porcentaje de los documentos seleccionados para revisión**



Fuente. Propia de la investigación.

## 2.2.3 El Pensamiento computacional y la educación STEM.

El trabajo de Wang *et al.* (2021) [11] estudian la integración del pensamiento computacional en la Ciencia, Tecnología, Ingeniería y Matemáticas (STEM) a través de una revisión bibliográfica semi-sistemática que incluye 55 estudios empíricos sobre este tema. Los resultados muestran que se adoptaron definiciones generales de dominio del pensamiento computacional y unos pocos propusieron definiciones específicas. Se resalta el modelo de instrucción basado en problemas y el diseño de juegos en el cual un tercio de las evaluaciones estaban orientadas al pensamiento computacional y la Ciencia, Tecnología, Ingeniería y Matemáticas (STEM) integradas. Dentro de ese marco se intenta crear definiciones de

pensamiento computacional para cada dominio STEM. Por otro lado, las evaluaciones se dirigieron a diferentes objetivos y formatos, observando el rendimiento y los procesos de aprendizaje entre diferentes grupos.

Así mismo, Yang *et al.* (2021) [47] también describen el diseño y la implementación de un plan de estudios integrado de STEM con el pensamiento computacional, guiado por el aprendizaje basado en proyectos. Proponen una integración a través de los programas extraescolares donde se examinan las reacciones de los profesores y los estudiantes al plan de estudios y sus desafíos de implementación, dando como resultado una reacción positiva hacia la ejecución del programa. El trabajo contribuye a la integración del pensamiento computacional y al desarrollo del mismo en los estudiantes y profesores.

Para Mohtadi *et al.* (2013) [14] integrar el pensamiento computacional en el plan de estudios de ingeniería en el siglo XXI es importante ya que los recién ingresados en los cursos de ingeniería tienen menos habilidades en materias como ciencia, tecnología, ingeniería y matemáticas (STEM) a pesar del constante manejo de dispositivos informáticos y electrónicos que poseen. Haciendo que los profesores de ingeniería se enfrenten al reto constante de preparar a los estudiantes en habilidades y conocimientos de diseño y trabajo en equipo que logre resolver problemas complejos, lo que hace necesario la integración del pensamiento computacional en el plan de estudio de ingeniería. Por otro lado, según la experiencia del autor, varias instituciones académicas han demostrado que al incluir el pensamiento computacional de forma constante incrementa un aprendizaje rápido de conceptos matemáticos mediante el uso de herramientas de software donde se adquiere una comprensión más profunda de la programación y la ingeniería, los estudiantes también aprenden a pensar de manera más autónoma, a investigar e indagar aplicando diversas herramientas que satisfagan múltiples objetivos de aprendizaje al mismo tiempo. Su capacidad de usarlas no es solo un requisito para los ingenieros de hoy en día, sino un recurso que les permite poner en práctica los conocimientos adquiridos pues se ha comprobado que utilizar el mismo conjunto de herramientas de software desde el primer año hasta los estudios de posgrado mejora la experiencia de aprendizaje, especialmente cuando esas mismas herramientas son utilizadas por ingenieros que trabajan en problemas del mundo real en la industria.

Por su parte, Sánchez Ludeña (2019) [4] expresa que a los estudiantes del siglo XXI les conviene formarse en habilidades para vivir en una sociedad cada vez más instruida en el uso de las tecnologías y así sobresalir en un mercado laboral cada vez más competitivo. Pero, de esta misma forma se demanda que docentes y estudiantes desarrollen una serie de habilidades y competencias debido a que la forma en que se ha venido entendiendo la educación no puede seguir siendo la misma, sino que esta debe innovar en su adaptación basada en las nuevas condiciones que manifiesta la sociedad.

Del mismo modo, expone que las ciencias deben estar en capacidad de brindar un método para comprender e interpretar el medio natural, ya que de esta manera la tecnología e ingeniería brindan técnicas y herramientas para afrontar la construcción de problemas y las matemáticas aportan un modo de expresión y representación, además de una forma de resolver problemas y fortalecer el pensamiento lógico. Dentro de este contexto afirma que, las metodologías que hacen parte del constructivismo son las más adecuadas para el desarrollo de las competencias STEM, destacando el trabajo por proyectos, en particular el “thinkering” y el movimiento “maker”, este último estrechamente vinculado al desarrollo de habilidades y competencias STEAM [4].

Por último Yepes Miranda (2020) [48], presenta su estudio STEM y sus oportunidades en el ámbito educativo, con el objetivo de describir el enfoque STEM y como este fortalece los procesos de enseñanza-aprendizaje en las instituciones educativas, a través de la identificación de experiencias significativas desarrolladas en diferentes contextos. Esto se realizó a partir de la revisión sistemática de proyectos y experiencias relacionadas con la educación bajo el enfoque STEM. Donde se describe el concepto, las características y las tipologías del enfoque de educación STEM, junto con las competencias que desarrolla en el ser humano, de acuerdo con las definiciones presentadas por los distintos autores.

Los artículos fueron analizados a partir de los objetivos educativos planteados y de sus concepciones de STEM, aplicada a los propósitos investigativos. La valoración de los artículos fue cualitativa, sin olvidar que deben estar presentes los criterios de selección expuestos anteriormente. El periodo de publicación de los artículos encontrados data de 2014 a 2020 [48].

#### **2.2.4 Integración del pensamiento computacional en Ingeniería.**

Se vienen realizando diferentes investigaciones a nivel mundial con el propósito de incentivar el pensamiento computacional en estudiantes universitarios de tecnología e ingeniería. Wu *et al.* (2019) [3] realizaron un análisis sistemático de artículos, informes y publicaciones académicas para identificar los principales elementos del pensamiento computacional en la ingeniería. Asimismo, efectuaron entrevistas estructuradas al personal académico de la Universidad de Zhejiang (ZJU), con el fin de organizar un marco considerando aspectos de alfabetización digital, habilidades de modelado y simulación, resolución de problemas complejos y liderazgo digital. De acuerdo con esto, se recomienda impulsar la educación informática con la intención de estimular en los estudiantes el pensamiento computacional dentro de un contexto de sistemas complejos de ingeniería haciendo énfasis en pensar de forma lógica, sistemática e innovadora.

Hacker (2018) [2] propone integrar este concepto en la enseñanza de la tecnología y la ingeniería donde se revisa y analiza diversas bibliografías. El autor desarrolla una encuesta a

educadores en la Universidad de Hofstra en diciembre de 2016, sobre la inclusión del pensamiento computacional en las materias de ingeniería (T&E). Hacker deduce un modelo de instrucción, que se basa en el diseño informado, mediante el cual los estudiantes completan tareas denominadas constructores de conocimientos y habilidades. A partir de sus hallazgos en esta investigación el autor realiza algunas recomendaciones al plan de estudios del programa para integrar los principios de pensamiento computacional.

Shoaib y Brophy (2020) [49] examinan cómo el pensamiento computacional mejora el aprendizaje mediante el tipo de formación indicada para el desarrollo del mismo. Se elabora una revisión bibliográfica sistemática para averiguar la respuesta a: ¿Qué nos informa la literatura sobre el aprendizaje o la transferencia a través del Pensamiento Computacional en las disciplinas científicas y de ingeniería?, ¿Cómo se puede facilitar el aprendizaje basado en el Pensamiento Computacional a través de la pedagogía? Lo que da como resultado diversas definiciones del pensamiento computacional y su aprendizaje; la incorporación de los conceptos, para lo cual se requiere el desarrollo y apoyo profesional continuo. También se establece cómo se diseña un entorno de aprendizaje activo basado en el construccionismo y la colaboración que despierte el interés de los estudiantes.

Por su parte Gross *et al.* (2014) [50] afirma que promover el pensamiento computacional en la enseñanza de la ingeniería es primordial en casi todos sus espacios de desempeño, incluyendo los que no necesitan un aparato electrónico, al realizar una revisión bibliográfica donde se destacan y extraen las mejores prácticas de los ejemplos que aplican pensamiento computacional se recalcan varios modelos de clases y proyectos que permiten a los estudiantes desarrollar habilidades computacionales a través de diversas herramientas. Se presentan tres estudios de caso de clases de ingeniería en universidades de renombre internacional, con sus respectivos planes de estudios que sirven como cursos de introducción a la ingeniería o a las asignaturas de ingeniería. Se generaliza el uso de herramientas computacionales para ayudar a los estudiantes a adquirir conocimiento y desarrollar habilidades para la resolución de problemas. Obteniendo una visión general de la aplicación de los principios de pensamiento computacional a los diseños de cursos y planes de estudio.

Es así como Miller *et. al.* (2014) [51] dice que Integrar el pensamiento computacional y creativo para mejorar el aprendizaje y el rendimiento en ciencias de la computación es una investigación diversa que abarca desde las especificaciones de los cursos hasta su desarrollo, creación de comunidades, establecimiento de políticas, enseñanza ,aprendizaje y evaluación. Para lo que se diseñó y puso en práctica cuatro ejercicios de pensamiento computacional en estudiantes de cuatro cursos introductorios de Ciencias de la computación de la Universidad de Nebraska, Lincoln. Para estos cursos diseñaron y aplicaron un conjunto de ejercicios de pensamiento creativo con vínculos a conceptos de ciencias de la computación y pensamiento computacional, que pretende mejorar el aprendizaje mediante su composición. Los resultados apoyan el argumento principal de que la asociación de ejercicios de pensamiento creativo

ayuda a optimizar el aprendizaje del pensamiento computacional. De esta manera, los ejercicios tienen un impacto en el rendimiento y el aprendizaje de los estudiantes donde la codificación mejora la capacidad para conectar el conocimiento del pensamiento computacional con múltiples prácticas. Como resultado, los estudiantes de ciencias de la computación obtuvieron un aumento constante del CT con cada ejercicio.

Entorno a esto Vergara *et al.* (2009) [52] afirman que la alineación de la enseñanza de la informática con la ingeniería y mejorar el pensamiento computacional de los graduados son muy importantes al diseñar e implementar procesos para transformar la educación informática dentro de los campos de la ingeniería y la tecnología, cuyo diseño debe involucrar a las partes interesadas, identificar las herramientas informáticas necesarias y las habilidades de resolución de problemas integrando los planes de estudio de las distintas disciplinas con las necesidades de la industria. Con esa finalidad y mediante instrumentos de encuesta y entrevista se determinó las competencias informáticas necesarias en una empresa y el uso que estos hacen de la tecnología. Involucrando a los profesores de ciencias de la computación e ingeniería de una universidad comunitaria y de una universidad de investigación. En el trabajo realizaron un rediseño de los planes de estudio de ingeniería herramientas computacionales, haciendo que la competencia en la resolución de problemas esté en consonancia con las necesidades de la industria. El objetivo es permitir que los graduados accedan a un determinado trabajo con un pensamiento computacional mejorado y preparado para la práctica, el cual incrementa sus habilidades de diseño y resolución de problemas; llegando a la conclusión del gran valor dado por los empleadores en cuanto a las habilidades interpersonales, como la comunicación, la capacidad de organizar, presentar datos y trabajar en equipo, el pensamiento crítico e innovador y la resolución de problemas que son atributos importantes en la actualidad, pues la tendencia a la globalización informática implica que los ingenieros comprendan las prácticas empresariales y la integración de los datos de ingeniería en sistemas más amplios que les permitan utilizar herramientas informáticas para resolver problemas de ingeniería entre los sistemas físicos y las abstracciones del software. La computación para la ingeniería no puede abordarse simplemente con uno o dos cursos de computación, sino que debe integrarse como parte de la formación de un ingeniero para convertirse en un "ingeniero holístico".

Así mismo Crawley *et al.* (2018) [53] hablan del rediseño de las enseñanzas universitarias de ingeniería en el Instituto Tecnológico de Massachusetts y de la iniciativa New Engineering Education Transformation (NEET), el cual está re-imaginando y rediseñando la enseñanza de ingeniería a través de este nuevo programa académico interdepartamental, que se centra en proyectos con colaboración formalizada entre especialidades cuyo objetivo es hacer que la formación en ingeniería sea más relevante, basado en los siguientes cuatro principios que están en consonancia con las necesidades de los estudiantes y de la sociedad: preparar a los estudiantes para desarrollar las nuevas máquinas y sistemas, preparar a los alumnos para ser hacedores y descubridores, formular propuestas que faciliten el aprendizaje de los estudiantes

y enseñar a los alumnos las formas de pensar de NEET: cómo pensar y cómo aprender más eficazmente por sí mismos. Para esto se realizó una encuesta a nivel mundial sobre la enseñanza de la ingeniería donde se comparte sus resultados y se explica las tres invenciones de NEET que surgen del análisis de las pruebas: la forma de pensar de NEET, el concepto curricular centrado en proyecto, y el concepto de hilos. Dando como resultado un marco de diferentes formas de pensar de NEET; que es base de ilimitadas aportaciones y pruebas. En 2017 se lanzó dos hilos piloto, Máquinas Autónomas (que abarcan la aeronáutica y la astronáutica, la ingeniería eléctrica y ciencias de la computación) y Máquinas vivas (que abarcan ingeniería biológica, ingeniería mecánica e ingeniería química, pero también abierto a cualquier estudiante del MIT que esté cursando un programa de grado técnico) y aunque este diseño es relativamente nuevo se han dado avances significativos con una opinión favorable sobre NEET y lo que pretende hacer.

Por otra parte, Atmatzidou y Demetriadis (2014) [27] sostienen que los estudiantes se familiarizan con los conceptos de CT y los integran satisfactoriamente al solucionar problemas en actividades de robótica educativa. En su investigación, los estudiantes desarrollaron con bastante éxito las habilidades de CT, siendo algoritmos, modularidad y descomposición aquellas con que se familiarizaron más y en menor tiempo, y abstracción y generalización con las que tuvieron mayor dificultad.

Posteriormente, Atmatzidou y Demetriadis (2016) [54] retoman su trabajo, sugiriendo con su investigación que ni la edad ni el género son impedimento para que los estudiantes alcancen, eventualmente, el mismo nivel de desarrollo de habilidades del CT y que éstas, en la mayoría de los casos, requieren tiempo para desarrollarse completamente. Es importante traer a colación aquí, como aporte para este proyecto, los conceptos que definen su modelo, como un proceso de solución de problemas, para el desarrollo de habilidades de CT y que se indica en la **Tabla 5**.

**Tabla 5. Modelo para el desarrollo de habilidades de CT propuesto por Atmatzidou y Demetriadis, condensado de [54] y [55]**

| Habilidad de CT                             | Definición  | Guía para su desarrollo   |
|---|---|---|
| Abstracción                                 | Es el proceso de crear algo simple a partir de algo complicado, omitiendo los detalles irrelevantes, encontrando los patrones relevantes y separando ideas de los detalles tangibles. | <ol style="list-style-type: none"> <li>1. Separar la información importante de la redundante.</li> <li>2. Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes.</li> <li>3. Identificación de abstracciones entre ambientes de programación diferentes.</li> </ol> |
| Generalización / Reconocimiento de Patrones | Es transferir un proceso de solución de un problema a una amplia variedad de problemas.   | <ol style="list-style-type: none"> <li>1. Expandir una solución existente en un problema dado para cubrir más posibilidades / casos.</li> </ol>   |

|                |  |  |
|----------------|--|--|
| Algoritmos     | Es la práctica de escribir, paso a paso, instrucciones específicas y explícitas, para llevar a cabo un proceso.  | <ol style="list-style-type: none"> <li>1. Describir explícitamente los pasos del algoritmo.</li> <li>2. Posibilidad de diferentes algoritmos para el mismo problema.</li> <li>3. Esfuerzo para encontrar el algoritmo más efectivo.</li> </ol> |
| Modularidad    | Es el desarrollo de procesos autónomos que encapsulan un conjunto de órdenes usadas con frecuencia que ejecutan una función específica y que podrían utilizarse en el mismo o en diferentes problemas. | Desarrollar secciones de código autónomas para ser utilizadas para el mismo o para diferentes problemas.   |
| Descomposición | Es el proceso de separar los problemas en partes más pequeñas que pueden resolverse con mayor facilidad.   | Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.  |

Fuente. Atmatzidou y Demetriadis, condensado de [54] y [55]

### 2.2.5 Evaluación de las habilidades de pensamiento computacional.

Djambong y Freiman (2016) [28] plantean una evaluación basada en tareas y habilidades de pensamiento computacional desarrolladas mediante la programación visual o entornos de codificación, con el fin de comprender la articulación entre las tareas de pensamiento computacional y las habilidades de los estudiantes y tipos de problemas que pretenden resolver, para lo que se realiza un estudio de caso piloto con dos grupos de estudiantes en tres etapas de la siguiente manera: revisión de la literatura sobre el pensamiento computacional, definición, desarrollo y evaluación, construcción de un cuestionario al tiempo que se realizaba la primera selección de tareas y su validación. Por último, se realizaron pruebas previas y posteriores utilizando las tareas seleccionadas con observaciones de la clase y entrevistas con el profesor y los estudiantes. Los resultados arrojados muestran una diferencia entre los tipos de habilidades evaluadas, la facilidad de las tareas y los grupos de edad. También se evidenció una proporción entre las puntuaciones medias obtenidas por tarea, con el nivel de dificultad previsto y una potencial relación entre la capacidad de los estudiantes para resolver las tareas propuestas, el tipo de habilidades objetivo relacionadas con el pensamiento computacional y el grado de dificultad o complejidad de las tareas presentadas.

A este respecto Dolgopolovas *et al.* (2015) [56] expone que si los estudiantes de primer año de ingeniería cuentan con un curso de programación estructurada (SP) bien desarrollado y centrado en la resolución de problemas se fortalecen sus habilidades de pensamiento computacional. Para esto se construyó y validó una prueba consistente en tareas preseleccionadas del concurso de Bebras el cual consiste en un conjunto de desafíos en forma de preguntas cortas o quiz. Estas tareas pueden ser resueltas sin conocimientos previos de

computación, pero están claramente relacionadas con sus conceptos fundamentales. Asimismo, se midió y comparó las habilidades latentes con los resultados del curso de programación estructurada. mostrando el potencial del test como instrumento para evaluar el pensamiento computacional, al mismo tiempo con su aplicación se evidenció que sólo el 54,2% de respuestas fueron contestadas correctamente por los estudiantes de primer año de ingeniería, lo que se puede deber a que los estudiantes no están suficientemente preparados a nivel escolar, y que no se abordan las habilidades de resolución de problemas de forma adecuada.

### **2.2.6 ChildProgramming en el desarrollo del pensamiento computacional.**

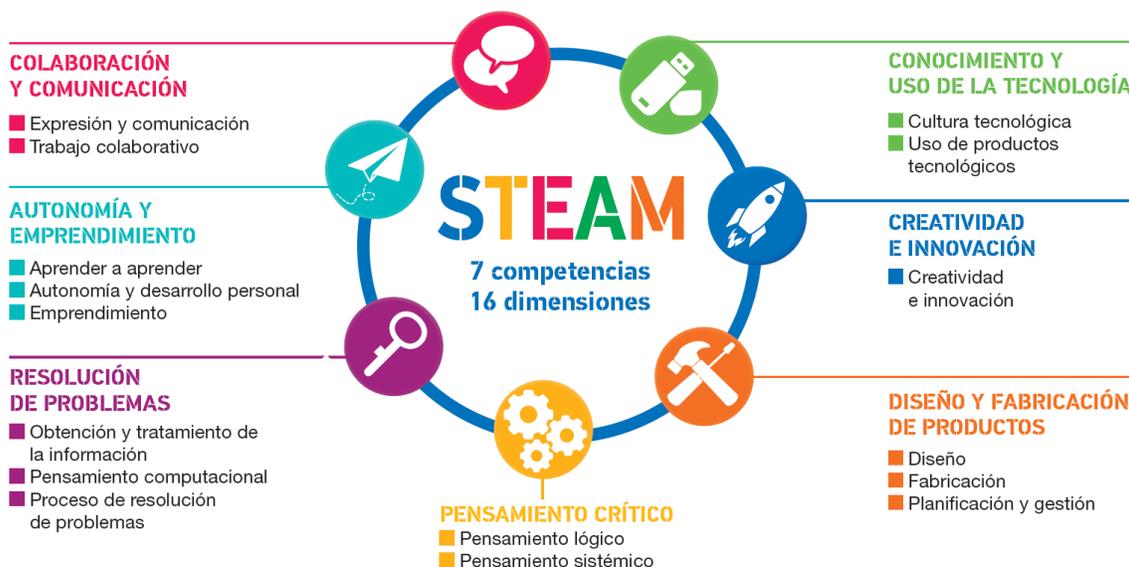
Conjuntamente con los artículos anteriores y como componente esencial para el desarrollo de este proyecto, es inevitable hablar más del modelo ChildProgramming que, más que como un trabajo único, debe ser visto como un proceso de desarrollo evolutivo y continuo, aún en curso. Así, por ejemplo, aunque el concepto de CT no aparecía de forma explícita en el modelo original de Hurtado et al. [5], sí ha hecho parte de trabajos posteriores como el de Chimunja et al. [15], llamado ChildProgramming-C, y el estudio de Zúñiga et al. [16]. El modelo se ha expandido a través de algunas extensiones tales como: ChildProgramming-C, dirigida a satisfacer la dimensión colaborativa del modelo, ChildProgramming-G [17], enfocada en la gamificación como forma de incorporar el aspecto lúdico al mismo. Child Gender, encaminada a extender ChildProgramming con un aspecto de diversidad de género [18]. ChildProgramming-MT, dirigida a estudiar el modelo para encontrar elementos que permitan desarrollar un sistema de Memoria Transactiva [19], ChildProgramming Evolution, se hace una revisión general de la evolución de la metodología ChildProgramming, [20]. ChildProgramming-RE, enfocada a incentivar la participación y motivación hacia el desarrollo del Pensamiento Computacional (CT) en niños entre 10 y 12 años, a partir de los conceptos de STEM relacionados con la Robótica Educativa [21]. ChildDebugging, Se extraen las principales estrategias y recursos utilizados por los equipos de niños para depurar programas basados en el paradigma de bloques [22]. Block-Based Design, propuesta basada en la aplicación de un método para el diseño de programas basados en bloques, utilizando mecanismos de abstracción, propios del pensamiento computacional, para facilitar a los niños la construcción, descomposición e integración de nuevos bloques [23].

### **2.2.7 Competencias STEAM y habilidades de CT.**

Yakman (2008) presenta el acrónimo STEAM introduciendo la A inicial de "Arts" dentro STEM definiéndolo como un enfoque educativo que persigue la integración y el desarrollo de las materias científico-técnicas y artísticas en un único marco interdisciplinar [57].

Dentro de este marco y teniendo en cuenta lo descrito por Sánchez Ludeña (2019) [4], se presenta a continuación a modo de resumen la **Figura 7**, en la cual se explica una serie de competencias y dimensiones que el enfoque STEM desarrolla en los actores que hacen parte del proceso de enseñanza-aprendizaje.

**Figura 7. Competencias y dimensiones STEAM [4]**



Fuente Sánchez Ludeña (2019) [4].

Por consiguiente, el enfoque STEM está constituido sobre una serie de elementos característicos que deben tenerse en cuenta. Para ello podemos agrupar las competencias y dimensiones STEAM como lo definen Sánchez Ludeña (2019) [4] y Yepes Miranda (2020) [48]. Estas permiten identificar cuando los actores han alcanzado el desarrollo de estas, evidenciándose de la siguiente manera en la **Tabla 6**.

**Tabla 6. Competencias y dimensiones STEAM con su definición[4], [48]**

| Competencias STEM[4]  | Dimensiones[4]                  | Guía para su desarrollo [48]  |
|---|---------------------------------|---|
| <b>Autonomía y emprendimiento</b><br>Acometer y llevar adelante un proyecto o propósito por propia iniciativa | Aprender a aprender             | Se debe aprender a organizar el tiempo, recolectar y organizar materiales y juegos, mostrar esfuerzo individual, trabajar con otros, recoger y ordenar información y presentar trabajos de forma adecuada |
|   | Autonomía y desarrollo personal | Se debe adquirir responsabilidades, aceptar cambios, planificar y decidir asumir consecuencias  |

|   |                               |   |
|---|-------------------------------|---|
|   | Emprendimiento                | Es necesario programar el aprendizaje y adquisición de las buenas relaciones sociales por parte del estudiante, partiendo de su valoración personal para llegar a su proyección social a través de la empatía, asertividad y destreza social.   |
| <b>Colaboración y comunicación</b><br>Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento  | Expresión y comunicación      | Está ligada al uso de códigos de comunicación, principalmente con el uso del gesto y el movimiento mediante el lenguaje corporal y al uso de la imagen y la representación con el lenguaje icónico. Se debe conseguir que los alumnos sean competentes para expresarse de forma clara y coherente con un vocabulario adecuado a su edad.  |
|   | Trabajo colaborativo          | Se refiere a una construcción colectiva del aprendizaje, en esta se tiene en cuenta aspectos como el ambiente del aula, los resultados, la dinámica de la clase, el conocimiento construido, la motivación y el producto terminado  |
| <b>Conocimiento y uso de la tecnología</b><br>Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso | Cultura tecnológica           | Definida como la información representacional, práctica o valorativa que comparten los miembros de un grupo social y que son potencialmente relevantes para la creación, producción, posesión o utilización de tecnologías o sistemas tecnológicos  |
|   | Uso de productos tecnológicos | Se trata de potenciar el desarrollo de destrezas necesarias para el uso de recursos tecnológicos y diferentes programas informáticos.   |
| <b>Creatividad e innovación</b><br>Resolver de forma original e imaginativa situaciones o problemas en un contexto dado   | Creatividad e innovación      | Se alcanza cuando se resuelven de forma original e imaginativa situaciones o problemas en un contexto dado. Se considera que dicha competencia se podría entender como el procedimiento pertinente en el que se conduce la formación profesional que demanda toda la sociedad actual, y que tiene como fin encontrar soluciones a problemáticas o limitaciones que hagan parte del contexto circundante |
| <b>Diseño y fabricación de productos</b><br>Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados  | Diseño                        | Se tiene en cuenta la identificación del problema y las necesidades para elaborar una serie de planos, gráficos, características del producto a fabricar, todo esto con el fin de resolver dicha necesidad.   |
|   | Fabricación                   | Se pone a disposición el diseño para elaborar un prototipo funcional del producto. También se le conoce como la etapa de materialización de la idea ya que en esta se lleva a cabo el proceso de construcción haciendo uso de los materiales y recursos respectivos.  |
|   | Planificación y Gestión       | Se evidencia a partir de la toma de decisiones y el manejo y liderazgo que se lleve a cabo a lo largo del proceso. Esta dimensión se caracteriza porque se debe tener en cuenta los roles asumidos a partir del trabajo en equipo, las responsabilidades a nivel colectivo y el cumplimiento y evaluación del mismo.  |

|  |  |   |
|--|--|---|
| <p><b>Pensamiento crítico</b><br/>Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos</p>                          | <p>Pensamiento lógico</p>                        | <p>Se define como aquel pensamiento que nace al relacionar objetos que además son elaborados por el propio individuo. El pensamiento lógico se desarrolla mediante la coordinación de relaciones entre objetos o hechos observados previamente, creado por el sujeto mediante la observación.</p>   |
|  | <p>Pensamiento sistémico</p>                     | <p>Es integrador, tanto al analizar las situaciones con en las conclusiones y resultados que nazcan de este. Aquí se ubica el pensamiento que busca soluciones a partir de la consideración de elementos generales del mundo real, estableciendo un conjunto de relaciones entre ellos y toda su estructura</p>   |
| <p><b>Resolución de problemas</b><br/>Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia</p> | <p>Obtención y tratamiento de la información</p> | <p>Se debe planear búsquedas en las que se identifica la confiabilidad de las fuentes consultadas, se discrimina la información requerida para posteriormente clasificarla y organizarla, respetar los derechos de autor, sintetizarla para ser usada y comunicar el conocimiento adquirido</p>   |
|  | <p>Pensamiento computacional</p>                 | <p>Se basa en aspectos como:</p> <ul style="list-style-type: none"> <li>• La confianza en el manejo de la complejidad, pues se enfrenta al estudiante a estos escenarios permanentemente.</li> <li>• La persistencia en trabajar con problemas difíciles, pues se aprende a través de desafíos La tolerancia para la ambigüedad</li> <li>• La capacidad de lidiar con problemas abiertos, que se generan en diferentes ámbitos de la vida, en el mundo real.</li> <li>• La capacidad de comunicarse y trabajar con otros para lograr un objetivo común o solución.</li> </ul> |
|  | <p>Proceso de resolución de problemas</p>        | <p>Se debe desarrollar un proceso investigativo que parte del reconocimiento del problema, su descripción, organización de ideas, los elementos que hacen parte de este, estrategias de aplicación y finalmente una solución que sea contrastada y acorde con el problema identificado</p>  |

**Fuente. Propia de la investigación**

Si bien las dimensiones de las competencias expuestas hasta aquí, fueron desarrolladas originalmente para STEAM, las mismas pueden aplicarse a cualquier tipo de STEM, puesto que estas competencias son un conjunto de capacidades, destrezas, habilidades y dimensiones que permiten fortalecer de forma significativa el proceso de enseñanza aprendizaje y prepararse para ser competentes en un mundo en constante evolución [48].

# Capítulo 3

## Modelo CT4E: Pensamiento computacional para la formación de ingenieros en el contexto STEM

El capítulo 3 presenta el diseño e instrumentalización de un modelo metodológico, a través de la incorporación de los conceptos de STEM, en el enfoque metodológico de ChildProgramming con el fin de integrar el desarrollo del pensamiento computacional (CT) en las actividades formativas de los estudiantes de primer semestre de las carreras de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial.

En el contexto de la metodología ChildProgramming, se requiere la búsqueda continua de nuevas destrezas que proporcionen una efectiva asimilación de los conceptos y desarrollo de las habilidades de pensamiento computacional; con esta finalidad se diseñó e instrumentalizó el modelo metodológico CT4E que hace uso de las competencias STEM como herramienta para el desarrollo de las habilidades específicas del CT.

Desde esta perspectiva y con el fin de conseguir su objetivo, esta extensión propone una serie de pautas claras para la aplicación de aspectos de STEM por medio de actividades orientadas hacia el desarrollo del pensamiento computacional y el uso de las competencias STEM en los estudiantes de primer semestre de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial siguiendo la dirección sugerida por Hacker (2018) [2] y Sánchez Ludeña (2019) [4], quienes proponen como línea de investigación, la Integración de los concepto de CT en la enseñanza de la tecnología y la ingeniería y el uso de las competencias de STEM como una de las herramienta para el desarrollo de habilidades de CT; asimismo se consideran los elementos encontrados en el estado del arte presentado en el capítulo 2 y los hallazgos de la fase exploratoria del estudio de caso descriptivo que se presenta en el capítulo 4.

A continuación, se describe el modelo CT4E el cual es un complemento del modelo ChildProgramming, incluyendo un conjunto de conceptos, procesos, métodos y prácticas. En la primera parte se contextualizará ChildProgramming. Después se relaciona e identifica los elementos del proceso de ChildProgramming que son afectados por el modelo CT4E. y finalmente se presentaran los conceptos, procesos, métodos y prácticas que complementan el modelo.

### 3.1 Modelo conceptual de CT4E

A través de una revisión de la literatura y la fase exploratoria del estudio de caso descriptivo realizado a los estudiantes de primer semestre de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial de la Universidad del Cauca, se identificaron las metodologías, estrategias y recursos más usados por los docentes para enseñar la materia de Introducción a la Ingeniería. Observándose que los docentes utilizaron implícitamente, los conceptos antes mencionados, pero usando diferentes nombres.

Aunque, en términos generales, la extensión propuesta no depende de un lenguaje de programación particular, fue evaluada en el contexto del lenguaje de programación visual Scratch (**Figura 8**), y a través de actividades desconectadas (**Figura 9**).

**Figura 8. Participantes Programando en Scratch durante la Fase Exploratoria del Estudio de Caso Descriptivo**



Fuente. Propia de la investigación

**Figura 9. Actividad Desconectada, Red de clasificación de seis entradas**



**Fuente. Propia de la investigación**

### **3.1.1 Estructura del modelo ChildProgramming.**

ChildProgramming es un proceso que guía el trabajo de un equipo de desarrollo de software conformado por niños, resultado de varios estudios que han permitido abstraer conceptos, actividades colaborativas y prácticas, en el contexto de equipos de niños de 8 a 12 años desarrollando software. Este modelo muestra como los niños, a través del trabajo colaborativo, prácticas ágiles y elementos cognitivos, logran obtener un conocimiento propio y particular en el área del desarrollo de software [5].

El proceso de ChildProgramming está compuesto por dimensiones, actores y elementos. Entre las dimensiones se encuentran la dimensión cognitiva, considerada como el esfuerzo que realizará un niño por comprender, analizar y apropiarse situaciones presentes en las tareas definidas, aportando al proceso los conceptos principales para el desarrollo de las mismas. La dimensión ágil, basada en la promulgación de los valores del manifiesto de las metodologías ágiles de desarrollo de software, esta dimensión aporta a ChildProgramming una forma acorde de trabajo para alcanzar los objetivos donde se evidencie un trabajo de equipo que permita a los integrantes del mismo permanecer juntos a lo largo de la actividad. El modelo ChildProgramming también propone la dimensión colaborativa, que intenta incrementar la calidad del aprendizaje y favorece la adquisición de conocimientos de los alumnos a través de la interacción entre ellos mediante el desarrollo de software [15]. Entre los actores se encuentran el profesor, estudiante y observador. Y entre los elementos están las actividades,

prácticas y roles. Los roles están divididos en los internos al proceso y los externos. Dentro de los roles internos del proceso están el profesor, guía del equipo y el equipo de trabajo. Dentro de los externos al proceso está el investigador u observador externo [5].

El eje de todo el proceso de ChildProgramming son las prácticas, donde los equipos de trabajo las aplican de forma adecuada para cumplir con la misión (el requerimiento en ChildProgramming). Dentro de ellas se encuentran las prácticas cognitivas, ágiles y colaborativas, donde cada una de ellas cuenta con su respectiva plantilla [15].

### **3.1.2 El modelo CT4E en ChildProgramming.**

La integración de las competencias STEM y el desarrollo de las habilidades de CT en ChildProgramming se ha empaquetado como una extensión denominada CT4E. Esta extensión se ubica principalmente en la dimensión cognitiva dentro de ChildProgramming y contribuye a través de una extensión del ciclo de desarrollo ChildProgramming original al Ciclo CT4E.

En la **Figura 10** se muestra la arquitectura conceptual del nuevo modelo completo CT4E, donde se muestran los paquetes que ya incluía ChildProgramming y la nueva extensión en la dimensión cognitiva que incluye los conceptos, procesos, métodos y prácticas que complementan el modelo.

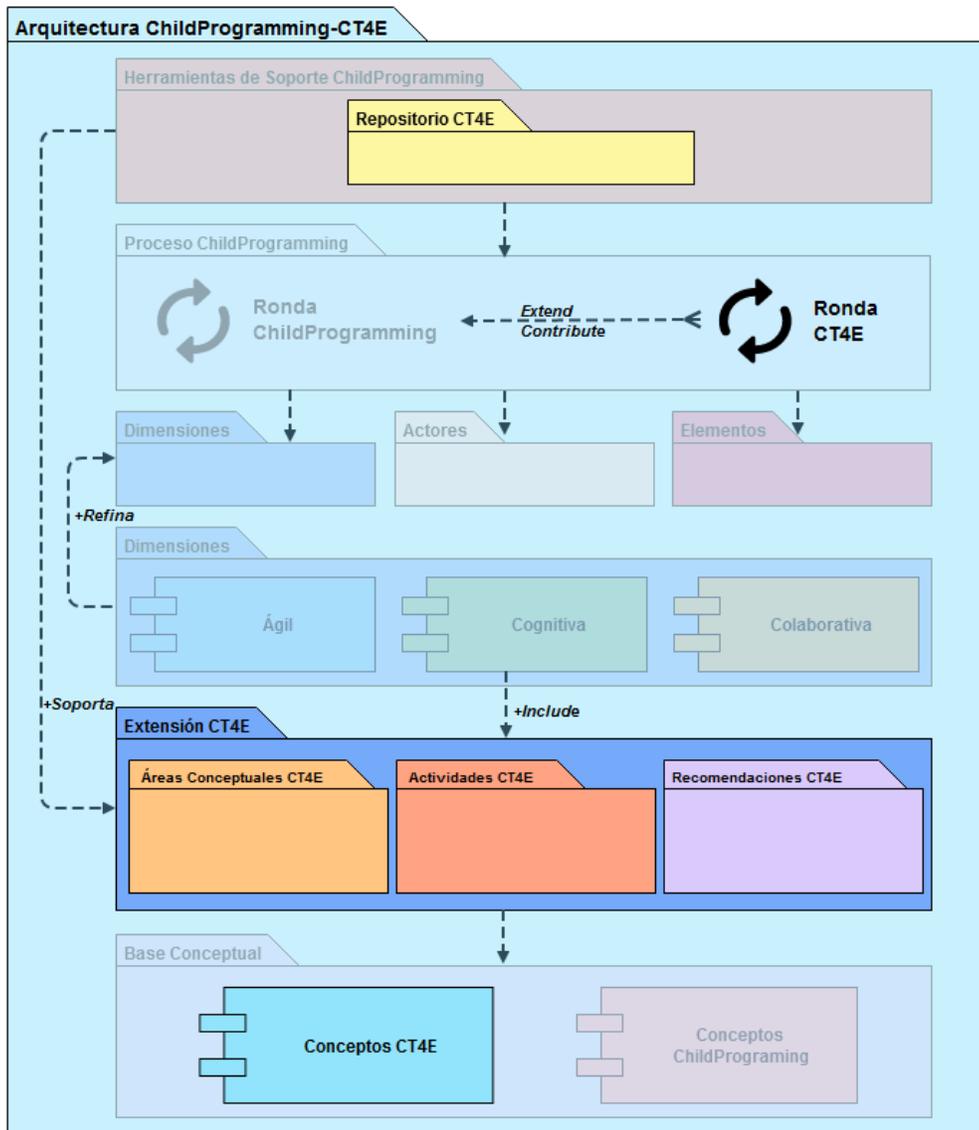
## **3.2 Arquitectura conceptual de CT4E**

El modelo del proceso CT4E es el principal producto de esta investigación, resultado de una revisión de la literatura y de un estudio empírico (estudio de casos) sobre unas prácticas de resolución de problemas a los estudiantes de primer semestre de Ingeniería electrónica y telecomunicaciones e Ingeniería Automática Industrial. El modelo del proceso CT4E refleja como los estudiantes involucrados en este estudio, logran obtener un incremento significativo en la apropiación de los conceptos del CT.

La arquitectura propuesta para el modelo de procesos CT4E está basada en la definida previamente en Cruz y Rojas [58]. Si bien el producto principal de este trabajo es incorporar un conjunto de conceptos, procesos, métodos y prácticas orientadas a la resolución de problemas, fue necesario desarrollar un marco conceptual para soportarlo. De esta manera, la arquitectura original del modelo ChildProgramming ha sido adaptada a esta propuesta, a partir de la experiencia en la fase exploratoria del estudio de caso descriptivo, siguiendo el enfoque de ingeniería de método situacional propuesto por Mirbel y Ralyté [26] la cual incluyó captura de los requerimientos del método, selección componentes de método y ensamblado de los

componentes del método: estructura, ciclo de vida, actores, dimensiones, paquetes metodológicos, actividades y plantillas.

Figura 10. Arquitectura ChildProgramming-CT4E



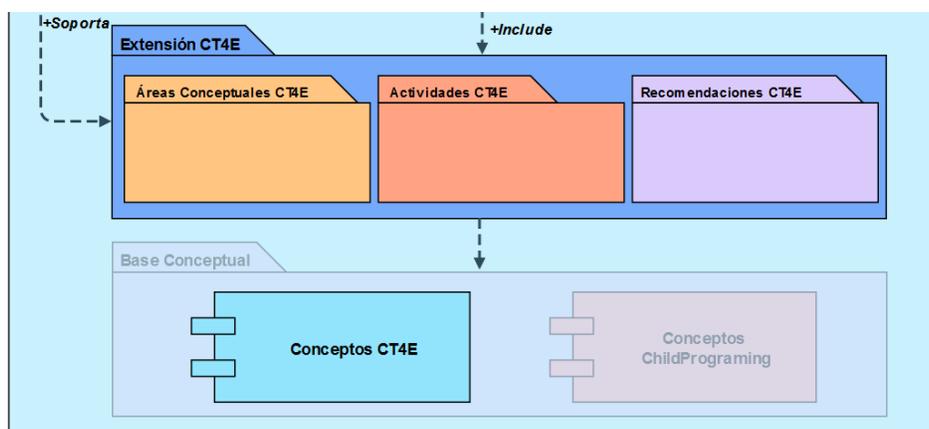
Fuente. Propia de la investigación

Partiendo de esta base arquitectónica, este trabajo aborda la implementación del modelo CT4E, basado en conceptos, procesos, métodos para crear nuevas prácticas y poder incluirlas como complemento en el modelo ya existente ChildProgramming.

### 3.2.1 Contexto del modelo CT4E.

Se denomina CT4E (Pensamiento computacional para la formación de ingenieros en el contexto STEM), por la combinación de las palabras “COMPUTATIONAL THINKING FOR ENGINEERING”. En la **Figura 11** se representa la estructura interna de esta propuesta llamada CT4E, que incluye los conceptos, procesos, métodos y prácticas de resolución de problemas, que a su vez tienen incluidas de forma transversal las competencias STEM y las habilidades de CT, que permiten un incremento significativo en la apropiación de los conceptos del CT y desarrollo de habilidades del mismo en los estudiantes de ingeniería.

Figura 11. Arquitectura del Modelo CT4E



Fuente. Propia de la investigación

Iniciando con esta arquitectura base, este trabajo aborda la re-definición del proceso ChildProgramming como un proceso incluyente de prácticas de resolución de problemas y define algunos elementos fundamentales del modelo como lo son áreas conceptuales, guía de actividades, recomendaciones CT4E.

### 3.2.2 Actores del modelo CT4E.

- **Docente:** encargado de comunicar el conocimiento teórico y definir las actividades prácticas que realizarán los estudiantes en sus equipos de trabajo durante el transcurso del proyecto.
- **Estudiantes:** son los actores fundamentales del aprendizaje, fuente principal del conocimiento y ejecutores de las actividades propuestas por el docente.
- **Observadores (Investigador):** encargados de apoyar el proceso en cualquier momento, teniendo en cuenta las disposiciones del docente y observando de manera detallada el desarrollo de los estudiantes en sus equipos para cada actividad propuesta.

### 3.2.3 Dimensiones de ChildProgramming-CT4E.

A continuación se describe cada una de las dimensiones que hacen parte del modelo ChildProgramming definido previamente en Cruz y Rojas [58].

- **Dimensión Cognitiva:** considerada como el esfuerzo que realizara un niño por comprender, analizar y apropiar 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, organizarse y adaptarse de manera que logren alcanzar su objetivo dentro de la actividad propuesta.
- **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. ChildProgramming a través de prácticas ágiles de desarrollo de software adicional al proceso este componente de tal forma que los niños y niñas 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.
- **Dimensión Colaborativa:** es el aprendizaje por descubrimientos, eje central de los procesos educativos actuales. El aprendizaje colaborativo involucra experiencias que ayudan a la construcción de conocimiento de forma adecuada acorde con el 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 del equipo de trabajo. ChildProgramming 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.

### 3.2.4 Roles del modelo CT4E.

A continuación se describe cada una de las dimensiones que hacen parte del modelo ChildProgramming definido previamente en Cruz y Rojas [58], los cuales no han tenido modificaciones en el nuevo modelo CT4E.

Tabla 7. Roles adaptados por CT4E como una extensión de ChildProgramming

| Roles adaptados por ct4e como una extensión de ChildProgramming |                                   |
|---|-----------------------------------|
| Internos del proceso  | Docente                           |
|   | Líder del equipo                  |
|   | Equipo de trabajo                 |
| Externos del proceso  | Investigadores/Observador externo |

Fuente. Cruz y Rojas [58]

La **Tabla 7** indica los roles definidos para el modelo CT4E, teniendo en cuenta el contexto de trabajo desarrollado con estudiantes de primer semestre de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial. A continuación, se describe cada uno de los roles propuestos por el modelo CT4E como una extensión al modelo ChildProgramming.

#### Internos del Proceso.

- **Docente:** es el responsable del proyecto de aprendizaje en el aula de clase, es el encargado de entregar a los equipos las pautas para la realización de la actividad, la misión a realizar, entrenar y monitoriza la metodología de trabajo y tener la disposición de aclarar las dudas e inquietudes que se presenten en cualquier momento del desarrollo. El docente interviene en el proceso siempre, aunque no debe ser intrusivo en el trabajo en equipo. además, de ser encargado de tomar las decisiones finales con respecto a la misión, también define y determina los objetivos y/o requisitos teniendo en cuenta la temática a tratar.
- **Líder del equipo:** es un estudiante responsable de asegurar el desarrollo de la misión acorde con las características y requisitos de esta. Tiene la función de estar pendiente de que el equipo trabaje acorde con las prácticas, valores y reglas establecidas y se avance en el desarrollo de la actividad según lo previsto, Trabaja igual que el resto del equipo y también es el responsable de eliminar las dificultades y de mantener un ritmo productivo tanto como sea posible. El líder del equipo es escogido en consenso por los integrantes del mismo.
- **Equipo de Trabajo:** es el conformado por (6±2) estudiantes 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 estas. El equipo

de trabajo está implicado en las valoraciones 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 el momento donde se presenten dificultades e inconvenientes en la realización de la Misión en las reuniones de trabajo continuas.

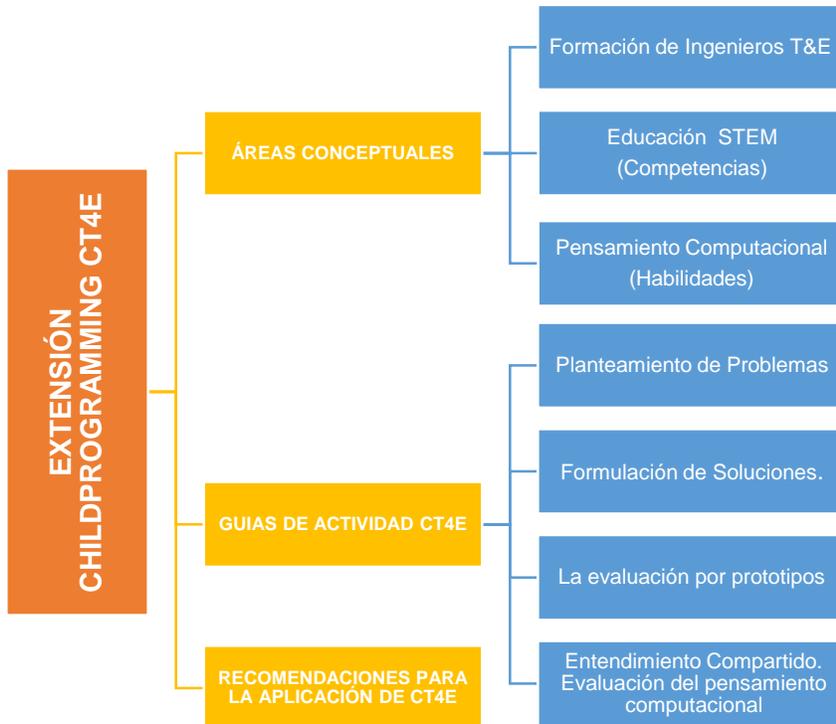
**Externos del Proceso.**

- **Investigador u Observador Externo:** es quien participa de la actividad observando el desarrollo de esta, inicialmente con el docente se encarga 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.

### 3.3 Paquetes metodológicos CT4E

La extensión CT4E, tal como se muestra en la **Figura 12**, se organiza en 3 paquetes metodológicos, descritos a continuación:

**Figura 12. Modelo CT4E y sus paquetes metodológicos**



**Fuente. Propia de la investigación**

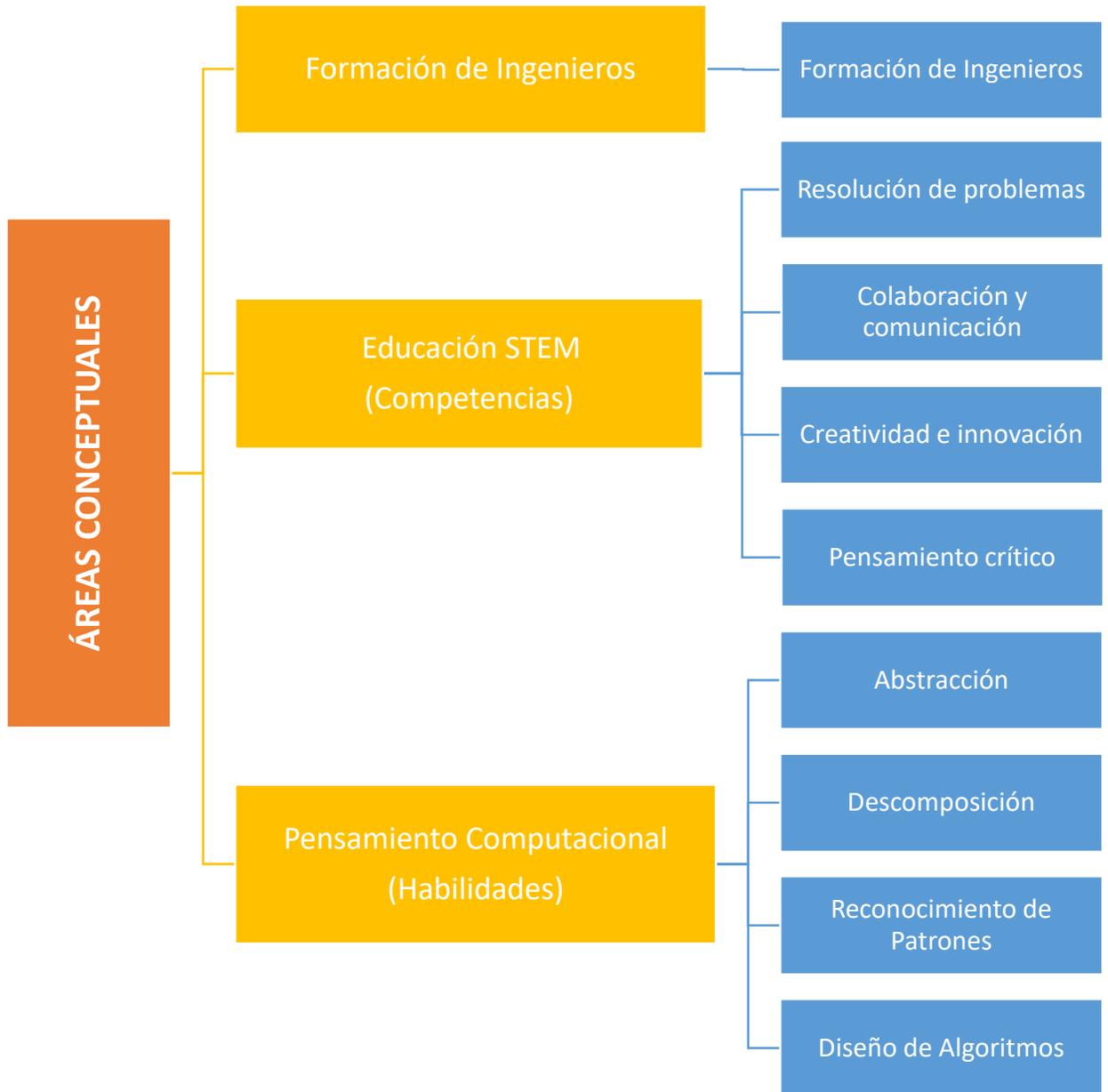
1. **Áreas Conceptuales:** en esta se detallan los conceptos de la formación de ingenieros T&E, los conceptos básicos de educación STEM (Competencias) y los de pensamiento computacional (Habilidades). Que instructores y estudiantes deben conocer para comprender con claridad el desarrollo de las actividades.
2. **Guía de Actividades CT4E:** se exponen cuatro tipos de guías de actividades que buscan fomentar la utilización de las competencias STEM como instrumento para el desarrollo del pensamiento computacional en la resolución de problemas.
  - 2.1. Guía de Actividad para el Planteamiento de Problemas: se centran en la identificación, la formulación y planteamiento de problemas.
  - 2.2. Guía de actividad para la Formulación de Soluciones: esta guía de actividad permite la formulación de soluciones de problemas.
  - 2.3. Guía de actividad para la Evaluación por Prototipos: el objetivo de esta guía actividad es realizar la evaluación por prototipo de la solución del problema.
  - 2.4. Guía de actividad de Entendimiento Compartido y evaluación del pensamiento computacional: cuyo objetivo es realizar sesiones de entendimiento compartido, evaluar y confirmar la asimilación de los conceptos de Pensamiento Computacional.
3. **Recomendaciones para la aplicación de CT4E:** la cual contiene una serie de sesiones específicas de recomendaciones, que permiten orientar al docente en la utilización del modelo CT4E.

En las subsecciones que prosiguen se describe en detalle cada uno de estos paquetes metodológicos.

### 3.3.1 Áreas conceptuales CT4E.

Dentro de este primer paquete metodológico se establecen las áreas conceptuales de formación de ingenieros, la educación STEM y la de pensamiento computacional con el propósito de ofrecer claridad conceptual al instructor orientador de los talleres, siguiendo esta extensión metodológica de ChildProgramming. El paquete está compuesto de los sub-paquetes “Formación Ingenieros T&E”, “Educación STEM (Competencias)” y “Conceptos del Pensamiento Computacional”, que se presentan en la **Figura 13** y se describen a continuación.

Figura 13. Áreas Conceptuales y sus sub-paquetes



Fuente. Propia de la investigación

### 3.3.1.1 Formación de ingenieros (T&E).

La Asociación Internacional de Enseñanza de Tecnología e Ingeniería (ITEEA<sup>20</sup>) define la enseñanza de tecnología e ingeniería como "*la forma en que los seres humanos modifican el mundo que les rodea para satisfacer sus necesidades y deseos o para resolver problemas prácticos.*" Por lo tanto, la enseñanza de tecnología e ingeniería es el aprendizaje basado en problemas por parte de los estudiantes, que utilizan los principios de las matemáticas, las ciencias, la ingeniería y la tecnología. Estos estudios implican: [44]

- Diseñar, desarrollar y utilizar sistemas tecnológicos;
- Actividades de diseño abiertas y basadas en problemas;
- Estrategias de aprendizaje cognitivas, manipulativas y eficaces
- Aplicar los conocimientos y procesos tecnológicos a experiencias del mundo real utilizando recursos actualizados;
- Trabajar individualmente y en equipo para resolver problemas [44].

### 3.3.1.2 Competencias STEM.

El término STEM fue acuñado por Judith Ramaley cuando era directora de la NSF<sup>21</sup> (National Science Foundation) en 2001 para sustituir a su predecesor SMET. El acrónimo SMET fue utilizado por primera vez en 1993 [35]. Este se dio a conocer al público en una serie de audiencias en el año 1997 [36]. La educación STEM es una integración de cuatro disciplinas de ciencia, tecnología, ingeniería y matemáticas en una clase, o lección basada en las diferentes conexiones entre las diferentes materias y los problemas del mundo real" [40].

En general, la educación STEM se refiere a proporcionar oportunidades para que los estudiantes sean capaces de resolver problemas, ser innovadores, inventores, seguros de sí mismos, pensadores lógicos y tecnológicamente alfabetizados [41]. Así mismo la educación STEM se ha contemplado desde la década de 1990 y puede tener éxito en la educación, en la preparación de estudiantes competentes en todos los niveles con las habilidades en la sociedad científica en rápida expansión, especialmente para el éxito futuro de los estudiantes [37]. La educación STEM también tiene el potencial para aumentar la alfabetización de todas las personas en toda la población en áreas tecnológicas y científicas [42].

De la misma forma, Sánchez Ludeña (2019) [4] afirma que a través de la Educación STEM, los estudiantes conseguirán desarrollar competencias claves como se muestra en la **Tabla 8**.

---

<sup>20</sup> ITEEA: International Technology and Engineering Education Association: <https://www.iteea.org/>

<sup>21</sup> NFS: National Science Foundation: <https://www.nsf.gov/>

**Tabla 8. Competencias STEM con su definición[4]**

| Competencias STEM[4]                | Definición   |
|-------------------------------------|--|
| Resolución de problemas             | Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia                                       |
| Colaboración y comunicación         | Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento   |
| Creatividad e innovación            | Resolver de forma original e imaginativa situaciones o problemas en un contexto dado   |
| Pensamiento crítico                 | Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos  |
| Autonomía y emprendimiento          | Acometer y llevar adelante un proyecto o propósito por propia iniciativa   |
| Conocimiento y uso de la tecnología | Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso  |
| Diseño y fabricación de productos   | Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados |

**Fuente. Sánchez Ludeña (2019) [4]**

### 3.3.1.3 Conceptos del pensamiento computacional.

“El pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de modo que queden representados de una manera que pueda ser abordada de forma efectiva por un agente-procesador de la información, como una computadora” [1]. En este tipo de pensamiento se enmarcan los siguientes conceptos fundamentales [55]:

- **Abstracción:** es la habilidad de identificar y extraer información relevante de un problema o una solución.
- **Descomposición:** habilidad de separar los datos, procesos, soluciones o problemas en partes más pequeñas y manejables.
- **Reconocimiento de patrones:** transferir un proceso de solución de un problema a una amplia variedad de problemas.
- **Algoritmo:** crear una serie ordenada de instrucciones para que una máquina o persona al ejecutarlos, resuelva un problema o realice una tarea.
- **Programación basada en bloques:** la programación basada en bloques implica unir piezas prediseñadas (como en un rompecabezas) de forma tal que se genere una lista de pasos o acciones a seguir para solucionar un problema planteado o creado por el mismo usuario. Es una manera práctica y visual de programar en entornos gráficos.

- **Análisis:** consiste en el estudio detallado del problema. Se deben identificar los datos de entrada, de salida y la descripción del problema.
- **Desarrollo de algoritmos:** práctica de escribir, paso a paso, instrucciones específicas y explícitas para llevar a cabo un proceso.
- **Depuración:** es el proceso de identificar y corregir errores de programación.

### 3.3.2 Guía de actividades CT4E.

Este paquete metodológico permite al docente, llámese instructor o profesor, y a los estudiantes, integrar en forma práctica las competencias STEM y las habilidades de CT.

Las actividades en el modelo CT4E se constituyen como el núcleo del proceso, donde los equipos de trabajo las aplican de forma efectiva para el cumplimiento de la Misión. Estas actividades incluyen las prácticas ágiles, colaborativas y cognitivas ya existentes de ChildProgramming [5] y las nuevas actividades que van vinculadas directamente en todo el proceso del modelo CT4E, las cuales son enfocadas en la resolución de problemas complejos de diferentes tipologías, mediante la exploración y la aplicación de forma transversal de las competencias STEM y los conceptos de CT y la utilización de enfoques computacionales. Para lograr alcanzar la misión dentro de cada una de las actividades propuestas por el docente. En la **Figura 14** muestra una representación de la relación transversal que tienen las competencias STEM y los conceptos de CT con las prácticas y actividades descritas en el modelo CT4E.

**Figura 14. Relación Transversal de las Competencias STEM y los Conceptos de CT con ChildProgramming**

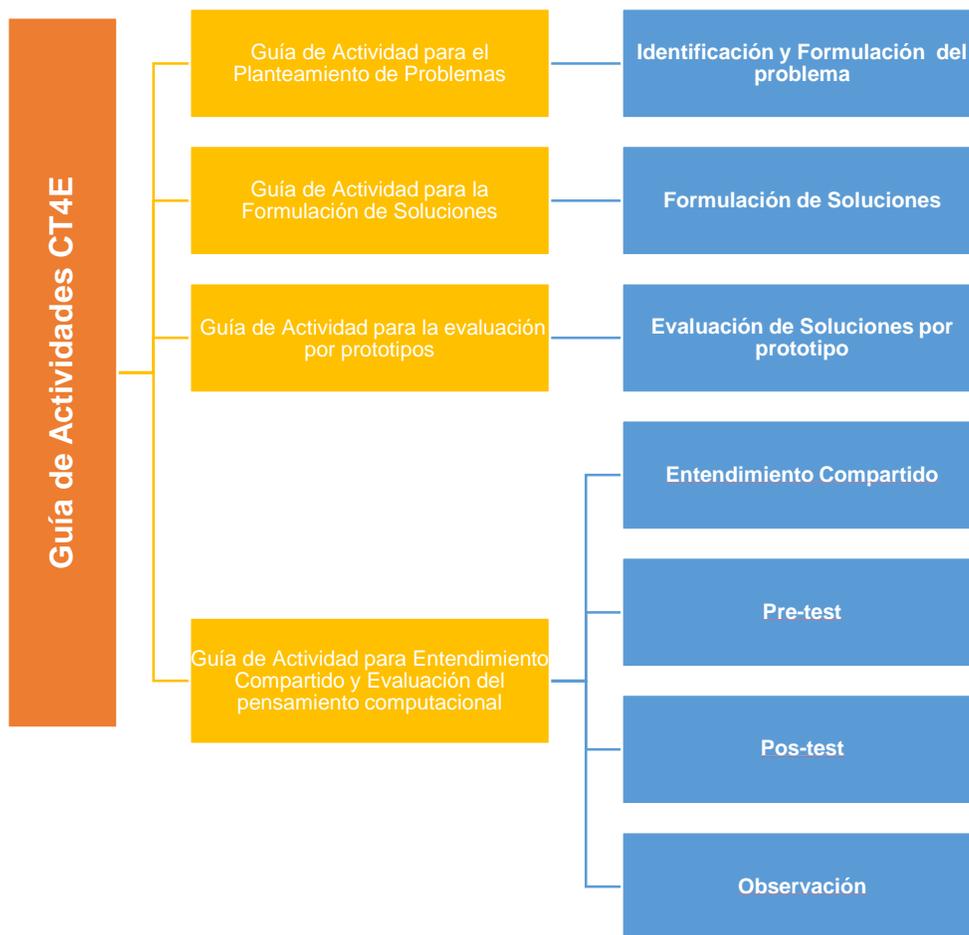


**Fuente. Propia de la investigación**

El paquete de actividades del modelo CT4E se ha dividido en los sub-paquetes: guía de actividad para el Planteamiento de Problemas, guía de actividad para la Formulación de Soluciones, guía de actividad para la Evaluación por Prototipos, guía de actividad para

Entendimiento Compartido y Evaluación del CT. La **Figura 15** muestra en forma gráfica la distribución de las guías de actividades propuestas por la extensión de ChildProgramming-CT4E.

**Figura 15. Guía de actividades modelo CT4E**



**Fuente. Propia de la investigación**

Cada una de las actividades de estos sub-paquetes, se describen mediante una estructura, a manera de plantilla. Las cuales se describen a continuación.

### 3.3.2.1 Especificación de las plantillas de actividades CT4E.

Las actividades contienen una serie de componentes donde se emplean, conceptos, cuestionarios y plantillas para el cumplimiento de las misiones o tareas y cuya finalidad es la incorporación de las competencias STEM y los conceptos CT en la resolución de los problemas para el desarrollo de las habilidades de CT.

Teniendo en cuenta que no todas las actividades poseen las mismas especificaciones, se diseñaron varios modelos de plantillas como se muestra a continuación:

### Plantillas usadas para describir las actividades CT4E según su tipo:

Esta está diseñada para las actividades de Planteamiento de Problemas, Formulación de Soluciones y la Evaluación por Prototipos. Las cuales surgen de una extensión de las prácticas usadas y definidas previamente en ChildGender [59] y nuevos ítems que fueron resultando a través de la experiencia de aplicar el modelo CT4E.

- **Plantilla No. 1 de actividades de resolución de problemas.**

**Tabla 9. Plantilla No. 1 de Actividades de resolución de problemas**

|   |   |  |
|---|---|--|
| <b>Identificador:</b>   | Representa al valor único de tipo alfanumérico de la práctica.  |  |
| <b>Nombre práctica:</b>                                       | Corresponde al nombre de la práctica definida, el cual es dado por el estudio realizado.  |  |
| <b>Descripción:</b>   | Describe de forma clara y sencilla en que se basa la práctica y además incluye una serie de recomendaciones útiles en el momento de usarla. |  |
| <b>Dirigida a:</b>  | Corresponde al rol a quien va orientada la práctica.  |  |
| <b>Contexto:</b>  | Corresponde a las fases del ciclo de vida CT4E a las cuales pertenece.  |  |
| <b>Proceso:</b>   |   |  |
| <b>Requisitos de entradas:</b>                                | <b>Resultados esperados:</b>  |  |
| Representan los recursos necesarios para aplicar la práctica. | Resultados generados después de su aplicación   |  |
| <b>Tareas a realizar:</b>                                     | Representa el flujo normal para la aplicación correcta de la práctica.  |  |
| <b>Sugerencias:</b>   | Recomendaciones útiles para el desarrollo de la actividad aplicando la práctica.  |  |

Fuente. Propia de la investigación.

- **Plantilla No. 2 de actividades de entendimiento compartido.**

**Tabla 10. Plantilla No. 2 de Actividades de entendimiento compartido**

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | Representa al valor único de tipo alfanumérico de la práctica.  |
| <b>Nombre práctica:</b> | Corresponde al nombre de la práctica definida, el cual es dado por el estudio realizado.  |
| <b>Descripción:</b>     | Describe de forma clara y sencilla en que se basa la práctica y además incluye una serie de recomendaciones útiles en el momento de usarla. |
| <b>Dirigida a:</b>      | Corresponde al rol a quien va orientada la práctica.  |

|   |  |  |
|---|--|--|
| <b>Contexto:</b>  | Corresponde a las fases del ciclo de vida CT4E a las cuales pertenece. |  |
| <b>Proceso:</b>   |  |  |
| <b>Requisitos de entradas:</b>                                | <b>Resultados esperados:</b>   |  |
| Representan los recursos necesarios para aplicar la práctica. | Resultados generados después de su aplicación                          |  |
| <b>Tareas a realizar:</b>                                     | Representa el flujo normal para la aplicación correcta de la práctica. |  |

Fuente. Propia de la investigación

- **Plantilla No. 3 actividades de evaluación de CT.**

Estas prácticas permiten la evaluación de conceptos, previos o aprendidos, y recopilar información necesaria para mejorar tanto la forma de orientar las sesiones como su planeación. Son ellas el Pre-test, Pos-test (**Tabla 11**), y la Observación (**Tabla 12**).

**Tabla 11. Plantilla No. 3 de Actividades evaluación CT (Pre-Test, Pos-Test)**

|   |   |  |
|---|---|--|
| <b>Identificador:</b>   | Representa al valor único de tipo alfanumérico de la práctica.  |  |
| <b>Nombre práctica:</b>                                       | Corresponde al nombre de la práctica definida, el cual es dado por el estudio realizado.  |  |
| <b>Dirigida a</b>   | Corresponde al rol a quien va orientada la práctica.  |  |
| <b>Contexto:</b>  | Corresponde a las fases del ciclo de vida CT4E a las cuales pertenece.  |  |
| <b>Descripción:</b>   | Describe de forma clara y sencilla en que se basa la práctica y además incluye una serie de recomendaciones útiles en el momento de usarla. |  |
| <b>Requisitos de entrada:</b>                                 | <b>Resultados obtenidos:</b>  |  |
| Representan los recursos necesarios para aplicar la práctica. | Resultados generados después de su aplicación   |  |
| <b>Tareas a realizar:</b>                                     | Representa el flujo normal para la aplicación correcta de la práctica.  |  |

Fuente. Propia de la investigación

- **Plantilla No. 4 Prácticas de observación.**

**Tabla 12. Plantilla No. 4 de Actividades evaluación CT (Observación)**

|   |   |   |
|---|---|---|
| <b>Identificador:</b>   | Representa al valor único de tipo alfanumérico de la práctica.  |   |
| <b>Nombre práctica:</b>                                       | Corresponde al nombre de la práctica definida, el cual es dado por el estudio realizado.  |   |
| <b>Dirigida a:</b>  | Corresponde al rol a quien va orientada la práctica.  |   |
| <b>Contexto:</b>  | En un curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial. |   |
| <b>Descripción:</b>   | Describe de forma clara y sencilla en que se basa la práctica y además incluye una serie de recomendaciones útiles en el momento de usarla.   |   |
| <b>Requisitos de entrada:</b>                                 |   | <b>Resultados esperados:</b>                  |
| Representan los recursos necesarios para aplicar la práctica. |   | Resultados generados después de su aplicación |
| <b>Tareas a realizar:</b>                                     | Representa el flujo normal para la aplicación correcta de la práctica.  |   |

**Fuente. Propia de la investigación**

### 3.3.2.2 Especificación de las prácticas ChildProgramming.

De esta misma manera, en el modelo inicial de ChildProgramming definido previamente, para especificar como se encuentran formadas las prácticas se han definido unas plantillas de acuerdo a la dimensión que pertenece.

La **Tabla 13** especifica la dimensión a la que pertenece cada una de las prácticas del modelo ChildProgramming de acuerdo a la plantilla.

**Tabla 13. Tipo de prácticas ChildProgramming.**

| <b>Identificador</b>             | <b>Nombre de la Práctica</b>   | <b>Dimensión</b> |
|----------------------------------|--|------------------|
| ChP02<br>ChP08<br>ChP10          | Cumplir las reglas de juego<br>Preguntar lo que no se entiende<br>Entender el tema de la actividad   | Cognitiva        |
| ChP05<br>ChP06<br>ChP07<br>ChP09 | Reunirse con un compañero(a) y realizar la tarea<br>Utilizar todo el lugar de trabajo con el equipo para informarse de la actividad.<br>Hacer la tarea de forma sencilla, realizándola cada vez mejor<br>Entregar la tarea hecha | Ágil             |
| ChP01<br>ChP03<br>ChP04          | Aceptar las condiciones para desarrollar la actividad<br>Desarrollar la actividad en equipo<br>Comprometerse para trabajar en equipo   | Colaborativo     |

**Fuente. Cruz y Rojas [58]**

### Plantilla prácticas cognitivas [59].

- **Identificador:** representa al valor único de tipo alfanumérico de 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 en que se basa la práctica y además incluye una serie de recomendaciones útiles en el momento de usarla.
- **Nota:** en caso de aclarar algo para la realización de la actividad que involucre la práctica.

La **Figura 16** especifica una de las varias prácticas cognitivas que pertenecen las prácticas ChildProgramming.

**Figura 16. Plantilla práctica cognitiva**

|   |  |
|---|--|
| <b>IDENTIFICADOR:</b>                   | ChP10  |
| <b>NOMBRE DE LA PRACTICA COGNITIVA:</b> | "Entender el Tema de la Misión"  |
| <b>DESCRIPCION:</b>                     | <div data-bbox="349 913 519 1102"></div> <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"><li>• El Equipo de trabajo una vez iniciada la actividad deben recibir del "Profesor" la guía para desarrollarla.</li><li>• El "Profesor" debe distribuir en los equipos los contenidos del tema a tratar para la actividad propuesta.</li><li>• El Equipo debe leer bien y comprender la información entregada según la temática correspondiente y dar inicio al desarrollo de la actividad.</li><li>• 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.</li></ul> |
| <b>NOTAS:</b>                           | <ul style="list-style-type: none"><li>❖ 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.</li></ul>  |

Fuente. ChildGender [59]

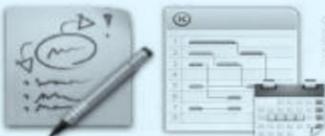
### Plantilla prácticas ágiles [59].

- **Identificador:** representa al valor único de tipo alfanumérico de 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 en que se basa la práctica y además incluye una serie de 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.

La **Figura 17** especifica una de las varias prácticas Ágiles que pertenecen las prácticas ChildProgramming.

**Figura 17. Plantilla práctica ágil**

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

Fuente. ChildGender[59]

### Plantilla prácticas colaborativas [59].

- Identificador: representa al valor único de tipo alfanumérico de 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 en que se basa la práctica y además incluye una serie de 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.

La **Figura 18** especifica una de las varias prácticas Colaborativas que pertenecen las prácticas ChildProgramming.

**Figura 18. Plantilla práctica colaborativa**

|  |   |  |
|--|---|--|
| <b>IDENTIFICADOR:</b>                      | ChP01   |  |
| <b>NOMBRE DE LA PRACTICA COLABORATIVA:</b> | <i>"Aceptar las Condiciones para Desarrollar la Misión"</i>   |  |
| <b>DESCRIPCION:</b>                        |  <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>  |  |
| <b>PROCESO</b>                             |   |  |
| <b>Entradas:</b>                           | <ul style="list-style-type: none"> <li>• Guía de la actividad propuesta.</li> <li>• Instrucciones del profesor.</li> </ul>  | <b>Salidas:</b>  |
|  |   | <ul style="list-style-type: none"> <li>• Compromiso de los integrantes del equipo reflejado en su participación en el transcurso de la actividad.</li> </ul> |
| <b>GUIA:</b>                               |   |  |
|  | <ul style="list-style-type: none"> <li>• El profesor entrega al equipo la guía para desarrollar la actividad.</li> <li>• El profesor imparte las instrucciones y aclara las dudas que se generen durante la explicación, si las hay.</li> <li>• El equipo acepta las condiciones para desarrollar la actividad y empieza a trabajar.</li> </ul> |  |

Fuente. ChildGender[59]

### 3.3.2.3 Actividades agregadas a ChildProgramming.

Las prácticas en el modelo ChildProgramming, definido previamente en Cruz y Rojas [58], se constituyen como el núcleo del proceso. A este le agregamos las nuevas actividades del modelo CT4E, para que los equipos de trabajo las aplican de forma efectiva para el cumplimiento de la misión.

En la **Tabla 14** se definen las actividades incluidas en el modelo CT4E con extensión del modelo ChildProgramming de acuerdo a la dimensión que pertenezca.

**Tabla 14. Descripción de las Actividades y Prácticas de ChildProgramming-CT4E por sus Dimensiones**

| Identificador | Nombre de la Práctica  | Dimensión    |
|---------------|--|--------------|
| CT4E01        | Identificación y Formulación del Problema  | Cognitiva    |
| CT4E02        | Formulación de Soluciones  | Cognitiva    |
| CT4E03        | Evaluación por Prototipos  | Cognitiva    |
| CT4E04        | Entendimiento Compartido   | Cognitiva    |
| CT4E05        | Pre-Test   | Cognitiva    |
| CT4E06        | Pos-Test   | Cognitiva    |
| CT4E07        | Observación  | Cognitiva    |
| ChP01         | Aceptar las condiciones para desarrollar la actividad                            | Colaborativo |
| ChP02         | Cumplir las reglas de 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    |

Fuente. Propia de la investigación.

#### Actividad para el planteamiento de problemas.

Con la guía de actividad para el planteamiento de problemas se busca mejorar la organización de los grupos de trabajo durante una sesión de entrenamiento y que los estudiantes apliquen las competencias STEM y los conceptos de CT en la resolución de problemas, que ayuden a

desarrollar las habilidades de CT. En este sub-paquete se incluyen una actividad práctica llamada Identificación y Formulación del Problema (**Tabla 15**).

### Actividad 1.

**Tabla 15. Identificación y formulación del problema**

|   |   |  |
|---|---|--|
| <b>Identificador:</b>   | CT4E01  |  |
| <b>Nombre práctica:</b>   | Identificación y Formulación del Problema   |  |
| <b>Descripción:</b>   | Los estudiantes a través de un contexto problemático global identifican un problema específico y lo formulan.   |  |
| <b>Dirigida a:</b>  | Estudiantes.  |  |
| <b>Contexto:</b>  | Planificación de la Ronda.  |  |
| <b>Proceso:</b>   |   |  |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>  |  |
| Equipos conformados.<br>Contexto problemático.<br>Ficha de Identificación y Selección del Problema (Anexo I <sup>22</sup> )<br>Ficha de Formulación y Planteamiento del Problema (Anexo J <sup>23</sup> ) | Problema identificado y descrito<br>Ficha de Identificación y Selección del Problema (Diligenciada)<br>Ficha de Formulación y Planteamiento del Problema (Diligenciada)   |  |
| <b>Tareas a realizar:</b>   | <ol style="list-style-type: none"> <li>1. Leer y discutir el contexto problemático</li> <li>2. Identificar problemas específicos en el contexto problemático a través de una lluvia de ideas</li> <li>3. Clasificar los problemas identificados por relevancia y oportunidad de ser resueltos a través de un enfoque de ingeniería</li> <li>4. Seleccionar los problemas relevantes y de oportunidad para la intervención por parte de una solución ingenieril</li> <li>5. Priorizar los problemas seleccionados</li> <li>6. Elegir alguno que se ubique en las prioridades más altas</li> <li>7. Describir el problema: el problema debe tener una descripción concisa de los hechos que precisan ser abordados. Además, debe responder a los cinco Ws de la metodología 5W2H: ¿Quién?, ¿Dónde?, ¿Qué?, ¿Cuándo? y ¿Por qué?.</li> </ol> |  |
| <b>Sugerencias:</b>   | Todos los miembros del equipo deben leer el contexto problemático.  |  |

Fuente. Propia de la investigación

### Actividad de formulación de soluciones.

Las siguientes prácticas proponen actividades que permiten desarrollar el Pensamiento Computacional en misiones de ChildProgramming con aplicaciones de competencias STEM.

<sup>22</sup> Ficha diseñada siguiendo los lineamientos de la metodología de Mario Bunge [7]

<sup>23</sup> Ficha diseñada con base en el método AQP y CCA de Rosario Martínez[60]

## Actividad 2.

**Tabla 16. Actividad de formulación de soluciones**

|   |  |  |
|---|--|--|
| <b>Identificador:</b>   | <b>CT4E02</b>  |  |
| <b>Nombre práctica:</b>   | Formulación de Soluciones  |  |
| <b>Descripción:</b>   | Los estudiantes a través de un problema identificado y descrito plantean la solución.  |  |
| <b>Dirigida a:</b>  | Estudiantes.   |  |
| <b>Contexto:</b>  | Planificación de la Ronda.   |  |
| <b>Proceso:</b>   |  |  |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>   |  |
| Equipos conformados<br>Problema identificado y descrito<br>Ficha de Identificación y Selección del Problema (Diligenciada) Anexo I.<br>Ficha de Formulación y Planteamiento del Problema (Diligenciada) Anexo J.<br>Ficha Formulación de Solución Anexo K <sup>24</sup> | Solución Priorizada y descrita<br>Ficha Formulación de Soluciones (Diligenciada) Anexo K.  |  |
| <b>Tareas a realizar:</b>   | <ol style="list-style-type: none"> <li>1. Leer y analizar el problema.</li> <li>2. Identificar las posibles soluciones a través de una lluvia de ideas.</li> <li>3. Clasificar las soluciones identificadas por relevancia y oportunidad de ser alcanzadas a través de un enfoque de ingeniería.</li> <li>4. Seleccionar las soluciones relevantes y alcanzables.</li> <li>5. Evaluar y priorizar las soluciones seleccionadas.</li> <li>6. Elegir la solución que se ubique en las prioridades más altas.</li> <li>7. Describir la solución.</li> </ol> |  |
| <b>Sugerencias:</b>   | Se debe tener a la mano los Anexos I y J diligenciados.  |  |

Fuente. Propia de la investigación

## Actividad para la evaluación por prototipos.

## Actividad 3.

**Tabla 17. Actividad de evaluación por prototipos**

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | <b>CT4E03</b>   |
| <b>Nombre práctica:</b> | Evaluación por Prototipos   |
| <b>Descripción:</b>     | El objetivo de esta actividad es estimular a los grupos de trabajo para que realicen un plan de trabajo y vayan probando la solución a través de prototipos software. |
| <b>Dirigida a:</b>      | Docentes e instructores, Estudiantes.   |

<sup>24</sup> Ficha diseñada siguiendo los lineamientos de la metodología de Mario Bunge [7]

|                           |  |  |
|---------------------------|--|--|
| <b>Contexto:</b>          | Durante las estaciones seguir la estrategia y revisar la estrategia se hace uso del desarrollo y evaluación de los prototipos.   |  |
| <b>Proceso:</b>           |  |  |
|                           | <b>Requisitos de entradas:</b>   | <b>Resultados esperados:</b>   |
|                           | <ul style="list-style-type: none"> <li>Ficha Formulación de Soluciones (Diligenciada) Anexo K.</li> <li>Ficha para la Evaluación por Prototipos Anexo L.</li> <li>Equipos conformados</li> <li>Docente atento y dispuesto a asesorar a los equipos de trabajo.</li> </ul>  | <ul style="list-style-type: none"> <li>Algoritmos y códigos de programación de cada equipo de trabajo.</li> <li>Los estudiantes empiezan a plantear por sí solos la posible solución a un problema.</li> </ul> |
| <b>Tareas a realizar:</b> | <ol style="list-style-type: none"> <li>Tareas de planeación y de gestión del trabajo diario del equipo a través del tablero de tareas</li> <li>Actividades de desarrollo del prototipo</li> <li>Actividades de prueba y depuración del prototipo</li> <li>Demostración del prototipo al cierre de una ronda</li> </ol> |  |
| <b>Sugerencias:</b>       | Se debe tener a la mano el Anexos K diligenciado.  |  |

Fuente. Propia de la investigación

### Actividad de entendimiento compartido y evaluación del pensamiento computacional.

Estas prácticas permiten la evaluación de conceptos, previos o aprendidos, y recopilar información necesaria para mejorar tanto la forma de orientar las sesiones como su planeación. Son ellas, el Entendimiento Compartido (**Tabla 18**), el Pre-test (**Tabla 19**), el Post-test (**Tabla 20**) y la Observación (**Tabla 21**).

#### Actividad 4.

**Tabla 18. Actividad de entendimiento compartido**

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | <b>CT4E04</b>   |
| <b>Nombre práctica:</b> | Actividad de Entendimiento Compartido   |
| <b>Descripción:</b>     | Se dan unas pautas para combinar las competencias STEM con los conceptos de pensamiento computacional   |
| <b>Dirigida a:</b>      | Estudiantes   |
| <b>Contexto:</b>        | Las competencias STEM y los conceptos de pensamiento computacional se deben tener en cuenta tanto al inicio del planteamiento de la estrategia como durante el desarrollo de la misma, ya sea en una sesión de programación o de una actividad de ChildProgramming. (Esto aplica a todas las estaciones o actividades primarias, puesto que en todas se requiere del conocimiento compartido, pero en especial cuando se revisa el prototipo, donde el docente puede retroalimentar la comprensión del problema y el enfoque que se le viene dando a la solución, re-direccionando los esfuerzos si es el caso) |

| Proceso:  |   |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
|---|---|----------------------|------------|----------------------------|--|-----------------------------|--|-------------------------------------|---|--------------------------|--|-----------------------------------|--|---------------------|---|-------------------------|--|-----------|-------------------------|-------------|--|
| Requisitos de entradas:   | Resultados esperados:   |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| <ul style="list-style-type: none"> <li>Exposición o asesoría del docente sobre cada concepto o competencia de STEM.</li> <li>Exposición o asesoría del docente sobre cada concepto del pensamiento computacional.</li> <li>Códigos de programación.</li> <li>Algoritmos o planes de trabajo.</li> <li>Descomposición y distribución de tareas en los equipos de trabajo.</li> </ul> | <p>Los estudiantes reconocen las competencias STEM y los conceptos de pensamiento computacional ya sea de forma explícita o por medio de ejemplos, al interactuar con el programa.</p>  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| <p><b>Tareas a realizar:</b></p>  | <p>✓ Dar a conocer las competencias STEM y los conceptos del pensamiento computacional, en la misión y aplicarlos a las actividades propias de programación, como lo es algoritmos y códigos de programación, en la primera aproximación que se les hace a los estudiantes.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #ADD8E6;">Competencias STEM[4]</th> <th style="background-color: #ADD8E6;">Definición</th> </tr> </thead> <tbody> <tr> <td>Autonomía y emprendimiento</td> <td>Acometer y llevar adelante un proyecto o propósito por propia iniciativa</td> </tr> <tr> <td>Colaboración y comunicación</td> <td>Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento</td> </tr> <tr> <td>Conocimiento y uso de la tecnología</td> <td>Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso</td> </tr> <tr> <td>Creatividad e innovación</td> <td>Resolver de forma original e imaginativa situaciones o problemas en un contexto dado</td> </tr> <tr> <td>Diseño y fabricación de productos</td> <td>Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados</td> </tr> <tr> <td>Pensamiento crítico</td> <td>Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos</td> </tr> <tr> <td>Resolución de problemas</td> <td>Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia</td> </tr> </tbody> </table> <p>✓ En [55], Atmatzidou y Demetriadis proponen el siguiente modelo de habilidades de CT, que se muestra en la siguiente tabla.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #ADD8E6;">Habilidad</th> <th style="background-color: #ADD8E6;">Guía para su desarrollo</th> </tr> </thead> <tbody> <tr> <td>Abstracción</td> <td> <ul style="list-style-type: none"> <li>Pedir a los estudiantes que separen la información importante de la redundante.</li> <li>Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes.</li> <li>Identificación de abstracciones entre ambientes de programación diferentes.</li> </ul> </td> </tr> </tbody> </table> | Competencias STEM[4] | Definición | Autonomía y emprendimiento | Acometer y llevar adelante un proyecto o propósito por propia iniciativa | Colaboración y comunicación | Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento | Conocimiento y uso de la tecnología | Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso | Creatividad e innovación | Resolver de forma original e imaginativa situaciones o problemas en un contexto dado | Diseño y fabricación de productos | Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados | Pensamiento crítico | Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos | Resolución de problemas | Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia | Habilidad | Guía para su desarrollo | Abstracción | <ul style="list-style-type: none"> <li>Pedir a los estudiantes que separen la información importante de la redundante.</li> <li>Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes.</li> <li>Identificación de abstracciones entre ambientes de programación diferentes.</li> </ul> |
| Competencias STEM[4]  | Definición  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Autonomía y emprendimiento  | Acometer y llevar adelante un proyecto o propósito por propia iniciativa  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Colaboración y comunicación   | Alcanzar metas y objetivos, resolver situaciones, abordar problemas en grupo y compartir el conocimiento  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Conocimiento y uso de la tecnología   | Ser tecnológicamente cultos. Entender y explicar los productos tecnológicos y saber utilizarlos, siendo conscientes de las precauciones y consecuencias de su uso   |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Creatividad e innovación  | Resolver de forma original e imaginativa situaciones o problemas en un contexto dado  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Diseño y fabricación de productos   | Diseñar y construir objetos y aparatos sencillos con una finalidad previa, planificando la construcción y usando materiales, herramientas y componentes apropiados  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Pensamiento crítico   | Interpretar, analizar y evaluar la veracidad de las afirmaciones y la consistencia de los razonamientos   |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Resolución de problemas   | Identificar, analizar, comprender y resolver situaciones problemáticas en las que la estrategia de solución no resulta obvia  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Habilidad   | Guía para su desarrollo   |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |
| Abstracción   | <ul style="list-style-type: none"> <li>Pedir a los estudiantes que separen la información importante de la redundante.</li> <li>Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes.</li> <li>Identificación de abstracciones entre ambientes de programación diferentes.</li> </ul>  |                      |            |                            |  |                             |  |                                     |   |                          |  |                                   |  |                     |   |                         |  |           |                         |             |  |

|                     |                |   |
|---------------------|----------------|---|
|                     | Generalización | <ul style="list-style-type: none"> <li>Expandir una solución existente en un problema dado para cubrir más posibilidades / casos.</li> <li>Emplear variables en una solución</li> </ul> |
|                     | Modularidad    | <ul style="list-style-type: none"> <li>Desarrollar secciones de código autónomas para ser utilizadas para el mismo o para diferentes problemas.</li> </ul>                              |
|                     | Descomposición | <ul style="list-style-type: none"> <li>Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.</li> </ul>   |
| <b>Sugerencias:</b> |                |   |

Fuente. Propia de la investigación

## Actividad 5.

Tabla 19. Actividad evaluación CT (Pre-Test)

|                           |  |   |
|---------------------------|--|---|
| <b>Identificador:</b>     | CT4E05   |   |
| <b>Nombre práctica:</b>   | Pre-Test   |   |
| <b>Dirigida a</b>         | Docentes o instructores.   |   |
| <b>Contexto:</b>          | En un curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial.  |   |
| <b>Descripción:</b>       | Es una evaluación que se hace al inicio de una sesión de entrenamiento o al iniciar un curso, con el fin saber que conocimientos previos se tienen de los temas a desarrollar en las sesiones de entrenamiento.  |   |
|                           | <b>Requisitos de entrada:</b>  | <b>Resultados obtenidos:</b>  |
|                           | Test escrito con preguntas sobre los conceptos a tratar durante el desarrollo del curso  | Nivel de desarrollo del pensamiento computacional del estudiante antes de iniciar cualquier actividad de aprendizaje relacionada. |
| <b>Tareas a realizar:</b> | <p>Esté test, recomienda aplicarlo antes de iniciar la primera sesión de entrenamiento.</p> <ul style="list-style-type: none"> <li>➤ En la primera sesión de clase aplicar un test escrito, ya sea preguntas de selección múltiple o abiertas.</li> <li>➤ Explicar al estudiante que no hay preguntas buenas o malas, sólo se realiza con fines informativos.</li> </ul> |   |

Fuente. Propia de la investigación

## Actividad 6.

**Tabla 20. Actividad evaluación CT (Pos-Test)**

|                          |   |   |
|--------------------------|---|---|
| <b>Identificador:</b>    | <b>CT4E06</b>   |   |
| <b>Nombre práctica</b>   | <b>Pos-Test</b>   |   |
| <b>Dirigida a:</b>       | Docentes o instructores.  |   |
| <b>Contexto:</b>         | En un curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial.   |   |
| <b>Descripción:</b>      | Test o evaluación que se realiza al final del curso, con el fin de evaluar los conceptos aprendidos durante el desarrollo de las sesiones de entrenamiento.   |   |
|                          | <b>Requisitos de entrada:</b>   | <b>Resultados obtenidos:</b>  |
|                          | Test escrito con preguntas relacionadas sobre los conceptos tratados durante el desarrollo del curso.<br><br>Soluciones a las misiones propuestas (código de programación en bloques)   | Nivel de desarrollo del pensamiento computacional del niño después de completar todas las actividades de aprendizaje planeadas. |
| <b>Tarea a realizar:</b> | <p>Está actividades son recomendadas para la penúltima o última sesión.</p> <ul style="list-style-type: none"> <li>➤ Aplicar un test escrito, ya sea con preguntas de selección múltiple o abiertas, similares a las del Pre-test</li> <li>➤ Explicar al estudiante que no hay preguntas buenas o malas, sólo se realiza con fines informativos.</li> </ul> |   |

**Fuente. Propia de la investigación**

## Actividad 7.

**Tabla 21. Actividad de observación**

|                         |   |  |
|-------------------------|---|--|
| <b>Identificador:</b>   | <b>CT4E07</b>   |  |
| <b>Nombre práctica:</b> | Observación   |  |
| <b>Dirigida a:</b>      | Docentes o instructores.  |  |
| <b>Contexto:</b>        | En un curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial.   |  |
| <b>Descripción:</b>     | La observación implica mirar el trabajo de los equipos y, además, indagar sobre las dificultades que se presentan en el desarrollo de las misiones, así como escuchar las soluciones planteadas a los problemas propuestos. |  |
|                         | <b>Requisitos de entrada:</b>   | <b>Resultados esperados:</b>   |
|                         | <ul style="list-style-type: none"> <li>• Distribución de los grupos de trabajo.</li> <li>• Docente atento al trabajo de los alumnos.</li> <li>• Plan de actividades del docente</li> </ul>                                  | Diario de observación que permita identificar problemas, soluciones y el impacto de las observaciones en el aprendizaje. |

|  |   |
|--|---|
| Diario de observación registrar algunos datos observados, como los propuestos por R. Strods [61] |   |
| <b>Tarea a realizar:</b>   | <ul style="list-style-type: none"> <li>➤ Tomar nota de los motivos que generan inconformidad en cada equipo.</li> <li>➤ Preguntar continuamente a los estudiantes sobre su progreso y, de alguna manera, hacer preguntas que les permitan identificar si están aplicando correctamente los conceptos explicados previamente.</li> <li>➤ Separar, aislar y, de ser necesario, reemplazar cualquier material o componente esencial que esté presentando fallas y que, en consecuencia, no permita un buen desarrollo de la misión.</li> </ul> |

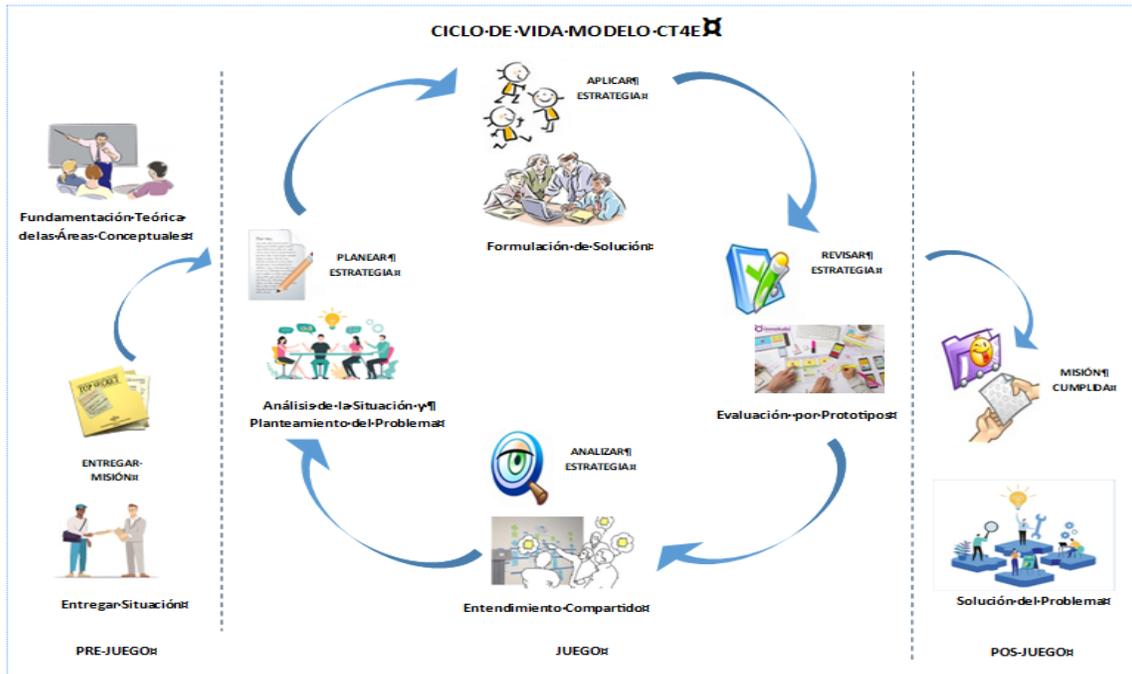
Fuente. Propia de la investigación

### 3.3.2.4 Definición CT4E como proceso.

El modelo CT4E se encuentra basado en la aplicación de actividades de resolución de problemas y en prácticas ágiles, colaborativas y cognitivas usadas en el modelo ChildProgramming definido previamente en Cruz y Rojas [58]. Las cuales fueron exploradas y empíricamente evaluadas con equipos de niños y niñas de edad escolar (sexto y séptimo de secundaria). Su ciclo de vida se encuentra basado en un marco de trabajo que comprende las siguientes fases: Pre-Juego, Juego, Post-Juego y en rondas de desarrollo. La **Figura 19** representa el ciclo de vida del modelo propuesto, sus fases, actividades y las nuevas categorías de actividades. Donde en cada una de las fases se van a agregar actividades de resolución de problemas. Así mismo, el modelo CT4E introduce las siguientes extensiones:

- El proceso contribuye a las fases de *pre-juego*, *juego* y *pos-juego*, las actividades de resolución de problemas, como una estrategia global que busca que los estudiantes asimilen los conceptos y desarrollen las habilidades de CT a través de la utilización de las competencias STEM para la resolución de problemas.
- El proceso vincula prácticas de resolución de problemas en la actividad de *planear estrategia*, *aplicar estrategia*, *revisar estrategia* y *analizar estrategia* dentro de todas las rondas de la fase de *juego* en ChildProgramming.

Figura 19. Ciclo de vida Modelo CT4E



Fuente. Propia de la investigación

### Pre-Juego.

El proceso de esta fase es establecer el objetivo de la misión, para hacer entrega de los requisitos involucrados en la actividad a realizar. En el contexto de CT4E, en esta fase el docente debe realizar una *Fundamentación teórica de las áreas conceptuales*.

Una vez culminado esto, el *Docente* hace entrega de la misión (Situación a analizar) al *Equipo de Trabajo*. La misión contiene una descripción detallada, donde cada integrante del equipo podrá conocer el objetivo y metas propuestas para la actividad. También ocurre que el *Docente* hace entrega al *Líder del Equipo* del 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.

### Juego.

En esta fase se realiza una serie de iteraciones denominadas *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 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 de planificación de la estrategia se lleva a cabo el análisis de la situación y el planteamiento del problema. Involucra la definición de las tareas y su priorización. El Docente dará al Equipo de Trabajo un tiempo para que en consenso con sus integrantes organizan una estrategia o una forma para abordar las tareas y su respectivo desarrollo. De esta misma forma, el Líder del Equipo solicita a sus integrantes la realización de un diseño inicial o bosquejo para que se les ayude a definir las tareas a través de las cuales afrontaran la misión.
- **Aplicar la estrategia:** Luego de realizar el análisis de la situación y el planteamiento del problema en su planeación, el *Equipo de Trabajo* entra a formular la solución del problema, guiado por su *Líder de Equipo* quien trabaja en paralelo con sus compañeros de equipo. En esta estación, es donde acumulan la mayor cantidad de aportes para su desarrollo y muestran el mayor índice de actividad evidenciando compromiso y participación activa por parte de cada integrante en el *Equipo de Trabajo*. De igual manera, en esta estación el *Equipo de Trabajo* realiza una apropiación de conceptos, prácticas de género, ágiles y colaborativas impartidas por el *Docente*, las cuales les ayudaran en el proceso de desarrollo para ejecutar las tareas de forma más sencilla y rápida. Logrando alcanzar mejores resultados al momento de entregar la misión.
- **Revisar la estrategia:** En esta estación el *Equipo de Trabajo* realiza la *evaluación por prototipos* y junto con el *Docente* verifica las tareas realizadas y el cumplimiento de las mismas, de manera que se logre evidenciar en el resultado. Esta estación permite a los integrantes del *Equipo de Trabajo* evaluar el avance de su *misión*, conocer el ritmo de trabajo, los cuales ayudaran para la planeación de la estrategia en la siguiente *Ronda*.
- **Analizar la estrategia:** En esta estación el *Equipo de Trabajo* y el *Líder del Equipo*, realiza *actividad de entendimiento compartido* y evalúan su trabajo como equipo entorno al desempeño de los integrantes, sus aportes y colaboraciones. De igual manera, evalúa el compromiso reflejado por cada uno de los integrantes y determina si la estrategia empleada funciona o no. Se puede dar la situación que el *Equipo de Trabajo* deba replantear la forma en la que están afrontando la tarea para darle una mejor solución y obtener mejores resultados, al final de la *misión*. Debido a que la *misión* haya sido acertada o por que no genero resultados, haciendo que sea reorganizada de manera inmediata la forma de trabajar para la siguiente *Ronda*.

En esta estación se permite al *Equipo de Trabajo* y al *Líder del Equipo*, analizar posibles situaciones futuras y tomar medidas oportunas para mitigar errores o complicaciones potenciales.

### **Post-Juego.**

En esta fase se hace entrega de la solución del problema evaluada como una *misión cumplida*, la cual corresponde a la solución completamente implementada acompañada de todo el material asociado al desarrollo de la actividad. El *Equipo de Trabajo* y el *Líder del Equipo* entregan la *misión cumplida* al *Docente*; el cual dará cierre de esta *misión*, evaluando que el trabajo del *Equipo de Trabajo* refleje un proceso de aprendizaje y haya permitido de forma efectiva, cumplir con los objetivos de aprendizaje propuestos.

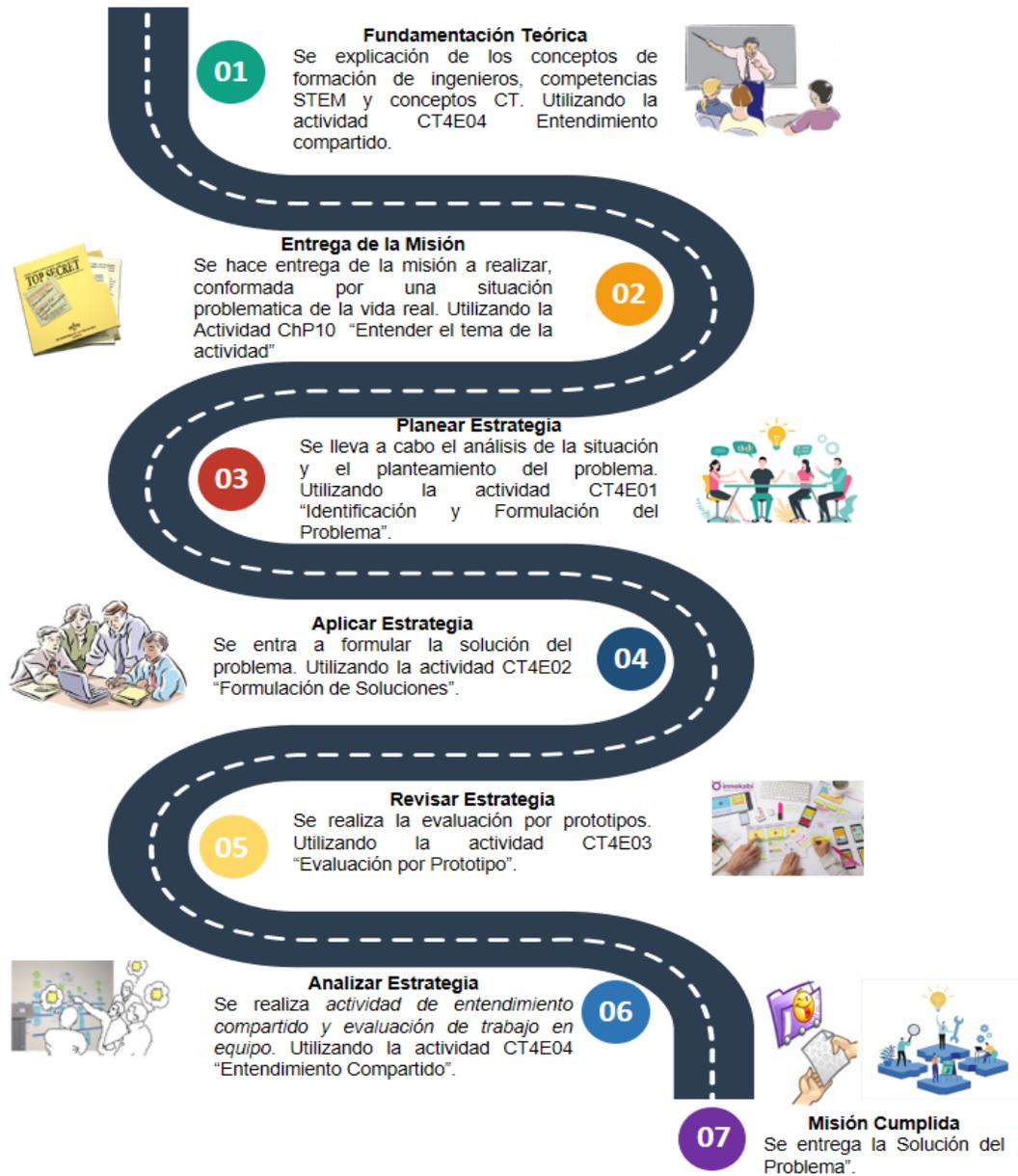
### **3.3.3 Recomendaciones para la aplicación de CT4E.**

Las recomendaciones para la aplicación del modelo CT4E, tiene por objetivo servir como orientación para aquellas personas interesadas en aplicar el modelo y trabajar el desarrollo del Pensamiento Computacional en los estudiantes de primer semestre de Ingeniería, empleando para ello como herramienta las competencias STEM. A continuación, en la Figura xx, se muestra la ruta de aplicación del modelo CT4E (Guía para el docente).

Figura 20. Ruta de Aplicación del Modelo CT4E

## GUÍA PARA EL DOCENTE

### Ruta de aplicación Modelo CT4E



Fuente. Propia de la investigación

# Capítulo 4

## Planeación, diseño, ejecución y resultados del estudio

En este capítulo se describen, en ese orden, la planeación, el diseño, la ejecución y el análisis del estudio realizado en el desarrollo del proyecto de investigación.

### 4.1 Planeación del estudio

En la planeación del estudio se tuvo en cuenta el objetivo de estudio y, a continuación, se plantearon tres hipótesis.

**Objetivo del Estudio:** evaluar que tan viable es la integración de las competencias STEM y ChildProgramming en el contexto de un caso de Introducción a la Ingeniería, con estudiantes de primer semestre de la carrera de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

**Hipótesis 1:** Al integrar las competencias STEM en el contexto de ChildProgramming como herramienta para el desarrollo del pensamiento computacional, se produce un incremento significativo en la asimilación de los conceptos propios de las actividades formativas de Ingeniería (T&E) en los estudiantes que participan en el experimento.

**Hipótesis 2:** Al integrar las competencias STEM en el contexto de ChildProgramming como herramienta para el desarrollo del pensamiento computacional, se produce un incremento significativo en la apropiación de los conceptos del CT (abstracción, descomposición, reconocimiento de patrones y diseño de algoritmos) por parte en los estudiantes que participan en el experimento.

**Hipótesis 3:** Al integrar las competencias STEM en el contexto de ChildProgramming como herramienta para el desarrollo del pensamiento computacional, se produce un incremento significativo en el desarrollo de las habilidades de CT en los estudiantes que participan en el experimento.

#### 4.1.1 Metodología.

El desarrollo de este proyecto se efectuará en 3 fases fundamentadas en los lineamientos del ciclo de investigación científica definida por Hurtado [6], donde se integra lineamientos generales de Bunge [7] y particulares de Shaw [8]. Destacando tres fases principales las cuales

son: exploración, formulación y evaluación. Siguiendo este orden inicialmente se realiza una revisión de la literatura existente, en la cual se formula el problema y se plantean las hipótesis. A continuación, se diseña e instrumentaliza un modelo metodológico, a través de la incorporación de los conceptos de STEM, en el enfoque metodológico de ChildProgramming con el fin de integrar el desarrollo del Pensamiento computacional (CT) en las actividades formativas de los estudiantes de primer semestre de carreras de Ingeniería en Telecomunicaciones. Por último, se evalúa el efecto del modelo en la asimilación de los conceptos propios del área disciplinar de Ingeniería (T&E), así como los propios del desarrollo del Pensamiento Computacional (CT) por parte de los estudiantes de primer semestre de carreras de Ingeniería en telecomunicaciones, a través de fase comparativa del estudio de caso descriptivo. Esta metodología propone exponer la organización y el cumplimiento de los objetivos expuestos anteriormente. Siendo apoyados en metodologías de investigación determinadas descritas en cada fase.

#### **4.1.1.1 Exploración.**

La finalidad de esta actividad es buscar el contexto de STEM, pensamiento computacional (CT), ChildProgramming, las herramientas, posibles modelos de integración y sistemas de evaluación del CT. Esto incluye consultar desde fuentes relevantes, la integración del desarrollo del pensamiento computacional en las actividades formativas en Ingeniería (T&E), incluyendo una exploración de las herramientas software y hardware elegidas para el proyecto (equipo de cómputo, lenguajes de programación, etc.). Para esto se condujo una revisión de la literatura utilizando el diagrama de flujo de cuatro fases de PRISMA<sup>25</sup> como procedimiento de calidad [24], y adaptando la guía diseñada por Turner *et al.* [25]. Además, se diseñó y ejecutó un estudio de caso exploratorio para lo cual se siguió la guía de Runeson *et al.* [9]. Los cuales 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 [9].

Así mismo, Yin [62] considera el estudio de caso como método de investigación apropiado para temas que se consideran prácticamente nuevos. De la misma forma, Chetty [63] 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, permitiendo estudiar un tema determinado, explorar en forma más profunda y obtener un conocimiento

---

<sup>25</sup> PRISMA: Preferred Reporting Items for Systematic reviews and Meta-Analyses: <http://www.prisma-statement.org/>

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.

Trabajar con este método 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.

#### **4.1.1.2 Formulación (Diseño del Modelo).**

En esta actividad se diseña e instrumentaliza un modelo metodológico, a través de la incorporación de los conceptos de STEM, en el enfoque metodológico de ChildProgramming con el fin de integrar el desarrollo del Pensamiento computacional (CT) en las actividades formativas de los estudiantes de primer semestre de carreras de Ingeniería en Telecomunicaciones., siguiendo el enfoque de ingeniería de método situacional propuesto por Mirbel y Ralyté [26] lo cual incluye capturar los requerimientos del método, seleccionar componentes de método y ensamblar los componentes del método.

## **4.2 Diseño del estudio de caso descriptivo**

El diseño del estudio comprende los siguientes aspectos: selección de los participantes, el contexto de la investigación, la pregunta de investigación, el objetivo del estudio, selección del estudio, selección de los participantes del estudio y los sujetos de investigación.

Para cumplir los objetivos de esta investigación se ha diseñado un estudio de caso descriptivo conformado por dos fases; una exploratoria y una fase comparativa desarrolladas con los estudiantes de primer semestre de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial, con el fin de obtener los insumos y las características necesarias para adecuar la propuesta de investigación al modelo ChildProgramming y de esta manera generar el modelo CT4E planteado para este trabajo.

### **4.2.1 Selección de los participantes del estudio.**

La población se seleccionó considerando la disponibilidad que brindaba la propia institución en el marco de los cursos de introducción a la ingeniería, además se consideró la disponibilidad de los individuos para fines experimentales. Se realizó un estudio de caso: descriptivo. En la fase exploratoria del estudio de caso descriptivo se trabajó con los estudiantes con el modelo de ChildProgramming; en la segunda fase del estudio de caso descriptivo se trabajó con el modelo CT4E.

## 4.2.2 Contexto de la investigación.

El desarrollo de esta investigación se realizó en las instalaciones de la Facultad de Ingeniería Electrónica y Telecomunicaciones (FIET) de la Universidad del Cauca, en el Edificio de Ingenierías, ubicado en la ciudad de Popayán (Colombia) en la carrera 2 # 4N -140, en el sector de Tulcán. Ver **Figura 21**.

**Figura 21. Edificio de Ingenierías, Universidad del Cauca**



Fuente. <http://www.unicauca.edu.co/fiet/acerca-de-la-facultad>

En la FIET confluyen estudiantes de los colegios del departamento del Cauca, así como otros en menor proporción de los departamentos Nariño, Valle, Huila y otras partes del país. El ingreso se realiza mediante pruebas ICFES (ponderados de ingreso) la edad de los jóvenes oscila entre los 16 a 23 años de edad. En la actualidad la FIET, cuenta con una población estudiantil de pregrado de 1.897 estudiantes entre mujeres y hombres, los cuales están distribuidos en sus tres programas profesionales y un tecnológico, como se muestra en la **Tabla 22**.

**Tabla 22. Distribución de estudiantes FIET**

| Programa                                    | Primer Semestre | Regulares    | Total        |
|---|-----------------|--------------|--------------|
| Ingeniería en Automática Industrial         | 40              | 386          | 426          |
| Ingeniería Electrónica y Telecomunicaciones | 60              | 628          | 688          |
| Ingeniería de Sistemas                      | 60              | 621          | 681          |
| Tecnología en Telemática                    | 0               | 102          | 102          |
| <b>Total</b>                                | <b>160</b>      | <b>1.737</b> | <b>1.897</b> |

Fuente. Propia de la investigación

En cuanto a recursos, la FIET está dotada de los elementos necesarios para el desarrollo de actividades académicas, deportivas y de recreación, destacando los recursos tecnológicos

como los habilitados en más 10 salas de informática. Las salas cuentan en promedio con 20 computadores de escritorio cada una, además de tener la disponibilidad de 20 computadores portátiles adicionales. De la misma forma, todas las salas tienen acceso a internet.

El estudio se realizó entre el mes de octubre del año 2022 al mes de enero de 2023. Las actividades se desarrollaron en 10 sesiones de trabajo de 2 horas cada una. Los lugares de reunión fueron los salones 328 Ingeniería Electrónica y Telecomunicaciones y el salón 313 Ingeniería automática Industrial. Se contó con la disponibilidad de equipos de cómputo sillas y mesas cómodas. En el momento de aplicar las actividades los docentes de los grupos estuvieron presentes como observadores del proceso. Ver **Figura 22**.

**Figura 22. Participantes del Estudio de Caso Descriptivo**



Fuente. Propia de la investigación

#### **4.2.3 Variables observadas.**

Conjunto de aspectos metodológicos considerados por los profesores para sacar adelante el curso.

Aspectos positivos identificados en el curso que van en línea con la integración de CT, STEM en la formación de ingenieros

Perfil de los estudiantes respecto a sus conocimientos de CT, STEM, Programación y Trabajo en equipo

#### **4.2.4 Preguntas de Investigación.**

¿Qué aspectos positivos y negativos se pueden identificar de la integración en la práctica de los conceptos de CT y STEM en la formación de ingenieros en el marco de los cursos de introducción a la ingeniería?

¿Qué tanto conocen los estudiantes de CT, STEM, Programación y Trabajo en equipo?

¿Qué tanto se ajusta la metodología ChildProgramming en el contexto del desarrollo de las habilidades CT y STEM en la formación de ingenieros en el marco de los cursos de introducción a la ingeniería?

#### **4.2.5 Objetivo del estudio.**

Entender cómo logran los docentes la integración en la práctica de los conceptos de CT y STEM en la formación de ingenieros en el marco de los cursos de introducción a la ingeniería.

#### **4.2.6 Selección del estudio.**

Nuestro proyecto está enfocado hacia la educación superior por lo tanto nuestra unidad de análisis son los estudiantes de los primeros semestres de ingeniería distribuidos en grupos, el estudio de caso es descriptivo tiene la finalidad de indagar y recoger datos los cuales nos proporcionaran información necesaria para proponer nuestro modelo. La selección se hizo considerado este contexto de estudio y de acuerdo a la disponibilidad que brindaba la propia institución en el marco de los cursos de introducción a la ingeniería.

#### **4.2.7 Los sujetos de investigación.**

Como unidades de análisis en el estudio, se tienen los grupos de trabajo definidos para el desarrollo de las actividades, compuestos por jóvenes con edades de 16 a 23 años. Los participantes son estudiantes de primer semestre de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial de la Universidad del Cauca, quienes conforman las fuentes primarias de información. El estudio es descriptivo, con el objetivo de entender cómo logran los docentes la integración en la práctica de los conceptos de CT y las competencias STEM en la formación de ingenieros en el marco de los cursos de introducción a la ingeniería.

#### 4.2.8 Instrumentos y técnicas para recolección de información estudio de caso descriptivo.

##### Fase Exploratoria estudio de caso descriptivo.

Los resultados del estudio de caso descriptivo se obtuvieron a partir instrumentos y técnicas de recolección de datos que posteriormente son analizados y evaluados dando autenticidad a la información recolectada. Con el propósito de recoger la información producida durante el estudio, se seleccionaron los siguientes instrumentos: entrevista, Pre-test, Pos-test, se adoptó el Protocolo de Observación propuesto por Strods et al. [61] y la herramienta de evaluación Dr. Scratch [64].

**La entrevista:** es una técnica personal que permite la recolección de información a profundidad donde el informante expresa o comparte oralmente y por medio de una relación interpersonal con el investigador su saber (opiniones, creencias, sentimientos, puntos de vista y actitudes) respecto de un tema o hecho [59].

**El Pre-Test fase exploratoria (véase el Anexo D):** es un cuestionario de 26 preguntas que se aplicó al inicio de las actividades. De esas 26, 9 corresponden a preguntas tipo Likert con escala de 5 valores y 17 preguntas abiertas, repartidas así: 8 sobre motivación y participación (ítems 1,2,3,4,5,6,7,8), 5 acerca de las preferencias de los participantes (ítems 9,10,18,19,25), 6 sobre conocimientos previos de Pensamiento Computacional (ítems 11,12,13,14,15,16), 7 sobre conocimientos de tecnología (17, 20, 21, 22, 23, 24, 26). En la **Tabla 23** se muestra la distribución de las preguntas en el Pre-Test.

**Tabla 23. Distribución de las preguntas según las variables a medir en el Pre-Test**

| Variables                        | Indicador   | Ítem                       |
|----------------------------------|---|----------------------------|
| Motivación                       | Expresa directamente su motivación                      | 1, 2, 3, 4, 5              |
| Participación                    | Expresa directamente su participación en equipo         | 6, 7, 8,                   |
| Preferencia de los participantes | Expresa directamente su preferencia                     | 9, 10, 18, 19, 25          |
| Conocimientos de CT              | Definición de CT  | 11                         |
|                                  | Abstracción   | 12                         |
|                                  | Algoritmos  | 13                         |
|                                  | Descomposición  | 14                         |
|                                  | Reconocimiento de patrones                              | 15                         |
|                                  | Programación  | 16                         |
| Conocimiento de Tecnología       | Posee conocimientos básicos relacionados con tecnología | 17, 20, 21, 22, 23, 24, 26 |

Fuente. Propia de la investigación

**El Pos-Test fase exploratoria (véase el Anexo F):** es un cuestionario de 26 preguntas, similar al Pre-Test, aplicado al final de las actividades exploratorias. De esas 26, 9

corresponden a preguntas tipo Likert con escala de 5 valores y 17 preguntas abiertas, repartidas así: 8 sobre motivación y participación (ítems 1,2,3,4,5,6,7,8), 5 acerca de las preferencias de los participantes (ítems 9,10,18,19,25), 6 sobre conocimientos previos de Pensamiento Computacional (ítems 11,12,13,14,15,16), 7 sobre conocimientos de tecnología (17, 20, 21, 22, 23, 24, 26). La **Tabla 24** resume la clasificación de las preguntas en el Pos-test.

**Tabla 24 Distribución de las preguntas según las variables a medir en el Pos-test**

| Variables                        | Indicador   | Ítem                       |
|----------------------------------|---|----------------------------|
| Motivación                       | Expresa directamente su motivación                      | 1, 2, 3, 4, 5              |
| Participación                    | Expresa directamente su participación en equipo         | 6, 7, 8,                   |
| Preferencia de los participantes | Expresa directamente su preferencia                     | 9, 10, 18, 19, 25          |
| Conocimientos de CT              | Definición de CT  | 11                         |
|                                  | Abstracción   | 12                         |
|                                  | Algoritmos  | 13                         |
|                                  | Descomposición  | 14                         |
|                                  | Reconocimiento de patrones                              | 15                         |
|                                  | Programación  | 16                         |
| Conocimiento de Tecnología       | Posee conocimientos básicos relacionados con tecnología | 17, 20, 21, 22, 23, 24, 26 |

Fuente. Propia de la investigación

**El Protocolo de observación (véase el Anexo E):** fue una herramienta utilizada para medir la motivación y el trabajo en grupo de los participantes en el desarrollo de las misiones. El protocolo se sigue con cada estudiante y cubre 11 campos, a cada uno de los cuales se asigna un puntaje en la escala de 1 a 4, cada uno con su correspondiente significado (0 = “no se puede observar”; 1 = “nivel bajo”; 2 = “se puede observar casi en todas las situaciones”; 3 = “se puede observar durante todos los proyectos”; 4 = “hace más de lo esperado”) [61].

Además de emplear los anteriores instrumentos, se hizo uso de la herramienta Dr. Scratch, para revisar el código de programación en bloques que los estudiantes generaron en las diferentes actividades. Dr. Scratch es una aplicación web que permite analizar proyectos codificados en Scratch para verificar si han sido programados correctamente, aprender de sus errores y obtener retroalimentación que puede ser usada para mejorar sus destrezas de programación y desarrollar sus habilidades de CT [64].

La **Tabla 25** muestra el nivel de competencia que evalúa la herramienta Dr. Scratch, al analizar los códigos de la programación en Scratch.

Tabla 25 Nivel de Competencia para cada aspecto del pensamiento computacional

| Aspecto del CT                            | Nivel de Competencia |   |  |  |
|---|----------------------|---|--|--|
|   | Ninguno (0 puntos)   | Básico (1 punto)                        | En desarrollo (2 puntos)   | Competente (3 puntos)  |
| Abstracción y descomposición de problemas | -                    | Más de un programa y más de un objeto   | Definición de bloques  | Uso de clones  |
| Paralelismo                               | -                    | Dos programas “en bandera verde”        | Dos programas en ‘tecla presionada’, dos programas en ‘al presionar’ el mismo objeto       | Dos programas en ‘cuando reciba mensaje’, crear clon, dos programas en ‘cuando %s es > %’, dos programas en ‘cuando el escenario cambie a’ |
| Pensamiento Lógico                        | -                    | Si                                      | Si - sino  | Operaciones lógicas  |
| Sincronización                            | -                    | Esperar                                 | Enviar, cuando reciba mensaje, parar todos, parar programas, parar programas del objeto    | Esperar hasta, cuando el escenario cambie a, enviar y esperar  |
| Control de flujo                          | -                    | Secuencia de bloques                    | Repetir, por siempre   | Repetir hasta  |
| Interactividad con el usuario             | -                    | Bandera verde                           | Tecla presionada, objeto presionado, preguntar y esperar, bloques de operaciones con ratón | Cuando %s es > %s, vídeo, audio  |
| Representación de la información          | -                    | Modificadores de propiedades de objetos | Operaciones con variables  | Operaciones con listas   |

Fuente. Dr. Scratch, M. Román-González [64]

### Fase Comparativa estudio de caso descriptivo.

Para evaluar el efecto del modelo en la asimilación de los conceptos propios del desarrollo del Pensamiento Computacional (CT) (abstracción, descomposición, reconocimiento de patrones y diseño de algoritmos) por parte de estudiantes de primer semestre de carreras de ingeniería en telecomunicaciones e Ingeniería Automática Industrial. Se utilizó el test de Marcos Román González [65], el cual evalúa los conocimientos sobre pensamiento computacional de acuerdo a criterios relacionados con la capacidad de: abstracción, descomposición, reconocimiento de patrones y habilidad para comprender y corregir algoritmos. Este test consta de 28 preguntas de múltiple respuesta con cuatro opciones, relacionadas con el marco conceptual propuesto

por Brennan & Resnick (2012) [34], las cuales se encuentran distribuidas en orden gradual de dificultad. Cada pregunta evalúa diferentes capacidades que se resumen en la siguiente **Tabla 26**.

**Tabla 26 Capacidades evaluadas por las preguntas del Test[65]**

| Pregunta | Abstracción | Descomposición | Reconocimiento de Patrones | Algoritmo |
|----------|-------------|----------------|----------------------------|-----------|
| 1        | X           |                |                            | X         |
| 2        | X           |                |                            | X         |
| 3        | X           |                |                            | X         |
| 4        |             | X              | X                          | X         |
| 5        |             | X              | X                          | X         |
| 6        |             | X              | X                          | X         |
| 7        | X           | X              |                            |           |
| 8        |             |                | X                          | X         |
| 9        |             |                | X                          | X         |
| 10       |             | X              | X                          | X         |
| 11       | X           | X              | X                          | X         |
| 12       | X           | X              | X                          | X         |
| 13       | X           | X              |                            | X         |
| 14       | X           |                |                            | X         |
| 15       | X           | X              | X                          | X         |
| 16       |             |                |                            | X         |
| 17       |             |                | X                          | X         |
| 18       |             |                | X                          | X         |
| 19       |             |                |                            | X         |
| 20       |             |                | X                          | X         |
| 21       | X           | X              |                            | X         |
| 22       | X           | X              | X                          | X         |
| 23       | X           | X              | X                          | X         |
| 24       |             |                |                            | X         |
| 25       | X           | X              | X                          | X         |
| 26       | X           | X              | X                          | X         |
| 27       | X           | X              | X                          | X         |
| 28       | X           | X              | X                          | X         |

Fuente. Barboza Díaz [29]

El Pre-test y el Pos-Test fueron diseñados en dos formularios de Google, siguiendo el test de Marcos Román González [65], para que fuera resuelto online. Los links son los siguientes:

El Pre-Test fase comparativa (véase el Anexo G): <https://forms.gle/QnSyfhUqffoUQ5WY9>

El Pos-Test fase exploratoria (véase el Anexo H): <https://forms.gle/pevzWm7CFGTHDR3DA>

#### 4.2.9 Ejecución, resultados y análisis de la fase exploratoria estudio de caso descriptivo.

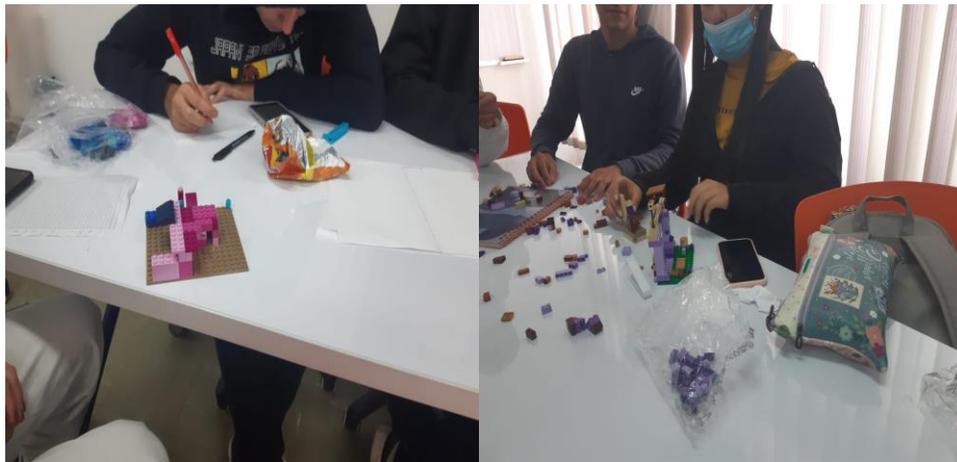
La fase exploratoria del estudio de caso descriptivo se realizó en el periodo comprendido entre el 15 y el 30 de noviembre del 2022, en la Facultad de Ingeniería Electrónica y Telecomunicaciones de Universidad del Cauca. Se realizaron cuatro sesiones de 2 horas cada una, a los estudiantes de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industria. En las cuales se efectuaron las siguientes actividades enmarcadas en el modelo ChildProgramming, Pre-Test, la actividad conectada “construyendo tu primer programa en Scratch”, conformada por tres misiones y la actividad desconectada “Red de clasificación de seis entradas” que incluía una sola misión y por último se presentó el Post-Test. El estudio se realizó considerando que ya los profesores contaban con plan curricular, en temas, actividades y fechas, así que el estudio tuvo que ajustarse a cada situación. A lo largo de todas las sesiones, los participantes, la distribución de los equipos y el ambiente de trabajo se mantuvieron sin cambios. A continuación, se describen las sesiones. tal y como se indica en las **Tabla 27** a la **Tabla 30**.

**Tabla 27 Sesión 1 Presentación, aplicación de protocolo de observación y entrevista al docente en clase de Introducción a la Ingeniería**

| <b>Sesión 1 Presentación, aplicación de protocolo de observación y entrevista al docente en clase de Introducción a la Ingeniería.</b> |   |
|--|---|
| <b>Nombre de la Actividad:</b>   | Presentación, aplicación de protocolo de observación y entrevista al docente.   |
| <b>Descripción:</b>  | Se realizó la presentación ante los estudiantes y docentes de primer semestre de ingeniería electrónica e ingeniería automática. Así mismo se hizo la entrevista a los docentes y se aplicó el protocolo de observación a los estudiantes, que realizaban una actividad basada en la metodología Design Thinking (Pensamiento de diseño) en la cual se deben crear prototipos para dar solución a un problema. La actividad planteada era encontrar problemas en la planta física de la universidad (sección de Tulcán). Las falencias pueden ser encontradas en baños, cafeterías, vigilancia etc. Y prototipar la posible solución. Por último se realizó la entrevista a cada docente. |
| <b>Materiales:</b>   | Hojas, lápiz, protocolo de observación, grabadora.  |
| <b>Metodología:</b>  | 1. Presentación ante los docentes y estudiantes de primer semestre de Ingeniería Electrónica e Ingeniería en Automática Industrial.<br>2. Se aplicó el protocolo de observación a los grupos de trabajo ya preestablecidos para esta actividad.<br>3. Realización de entrevista al docente.   |
| <b>Entregables de la Actividad:</b>  | Datos de protocolo de observación y datos de la entrevista.   |
| <b>Defectos:</b>   |   |

Fuente. Propia de la investigación

Figura 23. Evidencia fotográfica – Sesión 1



Fuente. Propia de la investigación

Tabla 28. Sesión 2 Pre-Test y Actividad “Construyendo tu primer programa en Scratch”

| Sesión 2 Pre-Test y Actividad “Construyendo tu primer programa en Scratch” |  |
|--|--|
| <b>Nombre de la Actividad:</b>   | Pre-test y Actividad “Construyendo tu primer programa en Scratch”  |
| <b>Descripción:</b>  | Se realiza el Pre-Test a los estudiantes de primer semestre de Ingeniería Electrónica e Ingeniería Automática, a continuación, se establecen los grupos de trabajo conformados por 4 integrantes, después se les entrego el plegable con las indicaciones pertinentes para el desarrollo de la actividad “Construyendo tu primer programa Scratch”.  |
| <b>Materiales:</b>   | <ul style="list-style-type: none"> <li>• Formato de Pre-Test por participante</li> <li>• Folleto guía de trabajo</li> <li>• Computador</li> <li>• Lenguaje de programación Scratch</li> <li>• Lápiz por participante</li> <li>• Borrador</li> </ul>  |
| <b>Metodología:</b>  | <p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> <li>1. Conformar cada grupo de trabajo de 4 integrantes.</li> <li>2. Entregar el Pre-Test a todos los participantes y pedirles que lo respondan, aclarando que el fin que persigue es investigativo y que no tiene incidencia en su continuación en el curso.</li> <li>3. Se entregó el plegable con las indicaciones pertinentes para el desarrollo de la actividad “Construyendo tu primer programa Scratch”.</li> </ol> |

Figura 24. Plegable Construye tu Primer Programa en Scratch.



Universidad  
del Cauca

Facultad de Ingeniería Electrónica y  
Telecomunicaciones

Programa de Ingeniería  
Electrónica y Telecomunicaciones.

Grupo IDIS — Investigación y  
Desarrollo en Ingeniería del  
Software



Proyecto  
“Integrando el Desarrollo del  
Pensamiento Computacional y  
los Conceptos T&E STEM en  
el Enfoque Metodológico de  
ChildProgramming”

Ángel Alveiro Gómez Noguera  
Pedro del S. Gómez Álvarez

Director: Ph.D. Julio Ariel Hurtado  
Alegria





Pensamiento  
Computacional

CONSTRUYE TU  
PRIMER PROGRAMA  
EN SCRATCH.

MISIÓN No. 1-3

FACULTAD DE INGENIERÍA  
ELECTRÓNICA Y TELECOMUNICACIONES

UNICAUCA

POPAYÁN –2022

Construyendo tu Primer Programa

Es una practica colaborativa siguiendo la metodología de Childprogramming, centrada en el marco de trabajo propuesto por Scrum y basado en prácticas ágiles de desarrollo propuestas por la metodología XP. Esta forma de trabajo plantea además, que esta práctica la puedan desarrollar de forma divertida, como un juego para que los estudiantes logren asimilar mejor los conceptos que se den dentro del proceso.

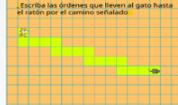
Los estudiantes realizaran las actividades diseñadas en lenguaje de programación Scratch. La practica se debe realizar en equipos.

Las siguientes son las misiones a realizar:

**Misión No. 1:** se pregunta cuáles son las órdenes que llevan al Gato hasta el Ratón por el camino señalado. Es decir, llevar al Gato EXACTAMENTE a la casilla en la que se encuentra el Ratón (sin pasarse ni quedarse corto), y siguiendo estrictamente el camino señalado en verde (sin salirse), los cuadrados verdes del camino miden 30 píxeles. Como se muestra a continuación:



**Misión No. 2:** se pregunta cuáles son las órdenes que llevan al Gato hasta el Ratón por el camino señalado. Es decir, llevar al Gato EXACTAMENTE a la casilla en la que se encuentra el Ratón (sin pasarse ni quedarse corto), y siguiendo estrictamente el camino señalado en verde (sin salirse), los cuadrados verdes del camino miden 30 píxeles. Como se muestra a continuación:



**Misión No. 3:** se pregunta cuáles son las órdenes para que el Gato de una vuelta al cuadrado. Es decir, llevar al Gato a la casilla en la que se encuentra inicialmente dándole la vuelta al cuadrado verde, los cuadrados verdes del camino miden 30 píxeles.



**Meta:**  
Programar la solución a las misiones, de la forma mas eficiente y corta.

**Materiales:**

- Guia de Trabajo
- Lenguaje de programación Scratch.
- 3 Fondos de Misión.

**Tiempo:**  
Tendrán 15 minutos para lograr programar la solución de cada misión.

**Pistas:**

- Cada equipo debe repartirse el trabajo adecuadamente para obtener el logro de la meta.
- Cada equipo debe tener un nombre como su identidad dentro de la misión, escribiendo en una hoja con sus integrantes.
- Observar detalladamente el recorrido en cada misión y determinar la estrategia a seguir (tiempo máximo 3 minutos para la planeación)
- Iniciar la programación de las soluciones de cada misión.

**Puntuación:**  
Se dará la puntuación de acuerdo al desarrollo de cada misión ( 50 puntos máximo y 5 puntos mínimo)

**Descuentos:**  
Por solicitar tiempo adicional se restaran 5 puntos.

*¡Buena Suerte!*

Fuente. Propia de la investigación.

**Entregables de la Actividad:**

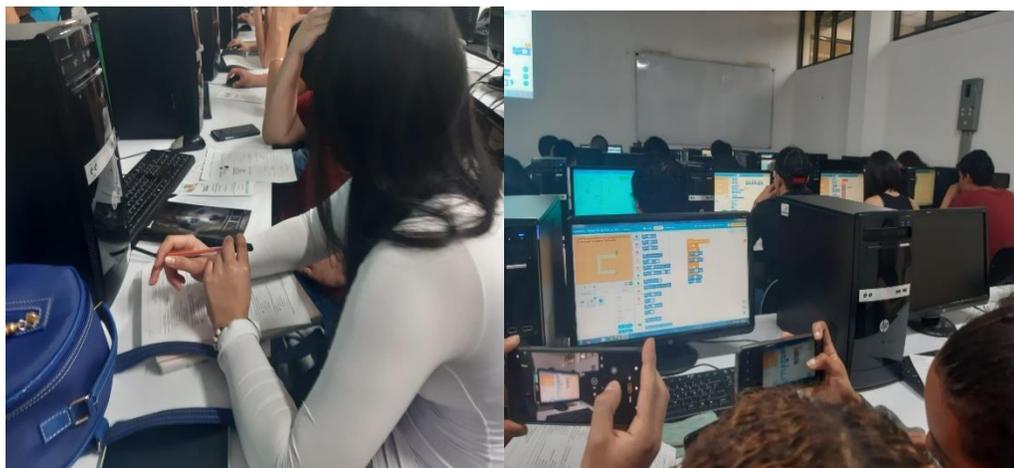
Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:

- Pre-Test Diligenciado.
- Archivo de programa de la solución en Scratch a cada una de las tres misiones..

**Defectos:**

Fuente. Propia de la investigación

**Figura 25. Evidencia fotográfica - Sesión 2**



**Fuente. Propia de la investigación**

**Tabla 29 Sesión 3 Actividad Desconectada “Red de clasificación de seis entradas”**

| <b>Sesión 3 Actividad Desconectada “Red de clasificación de seis entradas”</b> |  |
|--|--|
| <b>Nombre de la Actividad:</b>   | Actividad Desconectada “Red de clasificación de seis entradas”   |
| <b>Descripción:</b>  | Se entregó el plegable con las indicaciones pertinentes para el desarrollo de la actividad Desconectada “Red de clasificación de seis entradas”, a continuación, fueron dadas las indicaciones para la realización de la misión.   |
| <b>Materiales:</b>   | <ul style="list-style-type: none"> <li>• Formato de actividad Desconectada “Red de clasificación de seis entradas”</li> <li>• Folleto guía de trabajo</li> <li>• Lápiz por participante</li> <li>• Borrador</li> </ul>   |
| <b>Metodología:</b>  | <p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> <li>1. Conformar cada grupo de trabajo de 4 integrantes.</li> <li>2. Entregar el plegable con las indicaciones pertinentes para el desarrollo de la actividad Desconectada “Red de clasificación de seis entradas”</li> </ol> <p style="text-align: center;"><b>Figura 26. Plegable “Red de clasificación de seis entradas”.</b></p> |



Facultad de Ingeniería Electrónica y Telecomunicaciones

Programa de Ingeniería Electrónica y Telecomunicaciones.

Grupo IDIS — Investigación y Desarrollo en Ingeniería del Software



Proyecto “Integrando el Desarrollo del Pensamiento Computacional y los Conceptos T&E STEM en el Enfoque Metodológico de ChildProgramming”

Ángel Alveiro Gómez Noguera  
Pedro del S. Gómez Álvarez

Director: PhD. Julio Ariel Hurtado  
Alegría



ACTIVIDAD  
DESCONECTADA

MISIÓN No. 4

GUIA DE TRABAJO

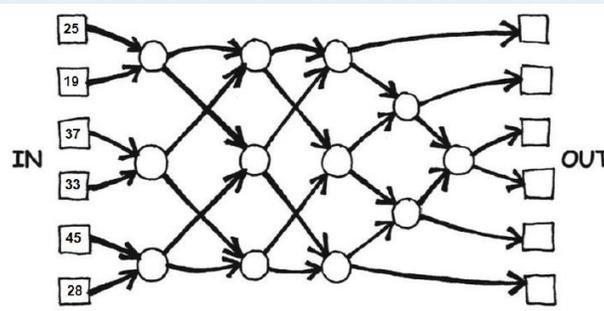
FACULTAD DE INGENIERÍA  
ELECTRÓNICA Y TELECOMUNICACIONES

UNICAUCA

POPAYAN –2022

Actividad Desconectada  
Red de clasificación de seis entradas (Bell et al, 2012 )

El objetivo de esta red es el de ordenar una serie de números de menor a mayor, los números estarán desordenados en la Entrada (IN) como se muestra a continuación:



Los estudiantes realizaran la misión en equipos. Las siguiente es la misión a realizar:

Misión No. 4: Ordenar los números de menor a mayor en la salida de la red de clasificación, documentando el algoritmo y la regla utilizada, y explicando que conceptos de pensamiento computacional utilizaron (Abstracción, Diseño de algoritmos, Descomposición, Reconocimiento de patrones), cuando finalmente los estudiantes alcancen los cuadros de la derecha (OUT) los números estarán ordenados de menor a mayor.

Meta:

Ordenar seis números utilizando la red de clasificación de seis entradas...

Materiales:

- Guía de Trabajo
- Red de Clasificación de seis Entradas (Hoja).

Tiempo:

Tendrán 15 minutos para lograr la solución de la misión.

Pistas:

- Cada equipo debe repartirse el trabajo adecuadamente para obtener el logro de la meta.
- Cada equipo debe tener un nombre, se debe escribir en la hoja respuesta, con el de los integrantes y su rol dentro de la misión.
- Observar detalladamente la red de clasificación para obtener la regla clave para solución de la misión. (tiempo máximo 3 minutos para la planeación).
- Ayuda: en Cada nodo (Círculo), se realiza una operación.
- Iniciar la Misión.

Puntuación:

Se dará la puntuación de acuerdo al desarrollo de cada misión (50 puntos máximo y 5 puntos mínimo)

Descuentos:

Por solicitar tiempo adicional se restaran 5 puntos.

¡Buena Suerte!

Fuente. Propia de la investigación.

Entregables de la Actividad:

Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:

- Algoritmo de la red de clasificación de seis entradas.

Defectos:

Fuente. Propia de la investigación

**Figura 27. Evidencia fotográfica - Sesión 3**



**Fuente. Propia de la investigación**

**Tabla 30 Sesión 4 Pos-Test y Actividad de entendimiento compartido**

| <b>Sesión 4 Pos-Test y Actividad de entendimiento compartido</b> |  |
|--|--|
| <b>Nombre de la Actividad:</b>                                   | Pues -Test y Actividad de entendimiento compartido   |
| <b>Descripción:</b>  | Se realiza el Pos-Test a los estudiantes de primer semestre de Ingeniería Electrónica e Ingeniería Automática, a continuación, se establecen los grupos de trabajo conformados por 4 integrantes, después se les entrego la actividad de entendimiento compartido con las indicaciones pertinentes para el desarrollo de la misma.   |
| <b>Materiales:</b>   | <ul style="list-style-type: none"> <li>• Formato de Pos -Test por participante</li> <li>• Presentación PowerPoint “competencias STEM y conceptos de CT”</li> <li>• Computador</li> <li>• Lápiz por participante</li> <li>• Borrador</li> </ul>   |
| <b>Metodología:</b>  | <p>La metodología empleada para esta actividad se describe a continuación:</p> <ol style="list-style-type: none"> <li>1. Conformar cada grupo de trabajo de 4 integrantes.</li> <li>2. Entregar el Pos-Test a todos los participantes y pedirles que lo respondan, aclarando que el fin que persigue es investigativo y que no tiene incidencia en su continuación en el curso.</li> <li>3. Presentación de competencias STEM y conceptos de CT para actividad de entendimiento compartido.</li> </ol> |
| <b>Entregables de la Actividad:</b>                              | <p>Como resultado de esta actividad los equipos de trabajo debían entregar los siguientes productos entregables:</p> <ul style="list-style-type: none"> <li>• Pos-Test Diligenciado.</li> <li>• Respuestas de afianzamiento de conceptos.</li> </ul>   |
| <b>Defectos:</b>   |  |

**Fuente. Propia de la investigación**

Figura 28. Evidencia fotográfica - Sesión 4



Fuente. Propia de la investigación.

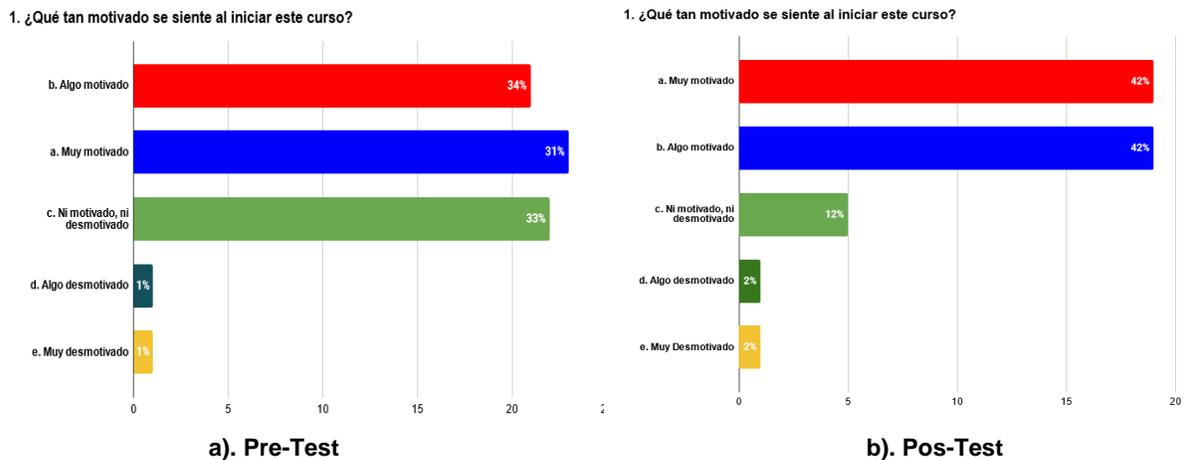
#### 4.2.10 Resultados cuantitativos.

Los resultados cuantitativos fueron obtenidos del análisis de las mediciones directas de los test aplicados. Entre estas encontramos las referentes a la motivación, la participación y la apropiación de conceptos de CT.

#### Motivación.

Con el propósito de contrastar la variación en la motivación de los participantes entre el inicio y el final del curso en la fase exploratoria del estudio de caso descriptivo, se realizó una comparación de los resultados obtenidos en el Pre-test y el Pos-test, tomando la pregunta N.1 como la más representativa. En ambos casos, se examina directamente la motivación de los estudiantes, obteniendo los resultados que muestra la **Figura 29**.

Figura 29. Distribución de las respuestas a la pregunta N.1 a) en el Pre-test y b) en el Pos- Test de Fase exploratoria del estudio de caso descriptivo



Fuente. Propia de la investigación

Con relación a los datos de la **Figura 29** se observa que, entre el inicio y el final del curso, el porcentaje de estudiantes que afirmo estar muy motivado paso del 34% al 42%, el de estudiantes que dijo estar algo motivado pasó del 31% al 42% y el de estudiantes que dijo estar ni motivado ni desmotivado pasó de 33% al 12%. Observándose, en general, un aumento en la motivación de los participantes del curso en la fase exploratoria del estudio de caso descriptivo.

La aplicación del Protocolo de Observación, en cuanto a los campos relacionados con la motivación de los participantes del estudio, mostro los resultados que se exponen en las **Tablas 31, 32 y 33**, Los cuales se encuentran resaltados.

**Tabla 31 Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación. Grupo A Ingeniería Electrónica y Telecomunicaciones**

| No. | Grupos<br>Criterios de Observación / Estudiantes:                               | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 3       | 4  | 1  | 4  | 3       | 4  | 3  | 4  | 3       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   | 4   | 4   | 3       | 3   | 3   | 3   |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 4       | 4  | 2  | 4  | 3       | 4  | 3  | 4  | 3       | 3   | 3   | 3   | 4       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   | 3   | 3   |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 4       | 4  | 1  | 4  | 3       | 3  | 4  | 4  | 3       | 4   | 4   | 4   | 4       | 4   | 4   | 3   | 3       | 4   | 3   | 3   | 3       | 4   | 4   | 4   |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 4       | 4  | 2  | 3  | 3       | 3  | 4  | 4  | 4       | 4   | 4   | 3   | 4       | 3   | 3   | 3   | 3       | 4   | 3   | 4   | 3       | 4   | 3   | 4   |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 3  | 2  | 3  | 4       | 3  | 4  | 3  | 3       | 3   | 4   | 3   | 4       | 3   | 4   | 3   | 4       | 3   | 3   | 4   | 3       | 3   | 3   | 4   |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 3       | 3  | 2  | 3  | 4       | 4  | 4  | 3  | 4       | 3   | 3   | 3   | 4       | 4   | 3   | 3   | 4       | 3   | 3   | 4   | 3       | 4   | 4   | 3   |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 3       | 4  | 1  | 3  | 4       | 4  | 3  | 3  | 4       | 3   | 4   | 4   | 4       | 3   | 3   | 3   | 4       | 4   | 3   | 3   | 4       | 4   | 3   | 3   |
| 8   | Está motivado para conocer los errores y corregirlos                            | 4       | 4  | 1  | 4  | 4       | 3  | 3  | 3  | 4       | 3   | 4   | 3   | 4       | 4   | 4   | 4   | 4       | 4   | 4   | 3   | 4       | 4   | 3   | 4   |
| 9   | Realiza actividades durante las clases.   | 4       | 3  | 2  | 4  | 3       | 4  | 4  | 3  | 4       | 3   | 3   | 3   | 3       | 4   | 3   | 4   | 4       | 4   | 4   | 4   | 3       | 4   | 4   | 3   |
| 10  | Participa en el trabajo en grupo  | 4       | 3  | 2  | 4  | 3       | 3  | 4  | 3  | 3       | 4   | 4   | 4   | 3       | 4   | 3   | 4   | 3       | 4   | 4   | 3   | 3       | 3   | 3   | 4   |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 4       | 3  | 2  | 4  | 3       | 4  | 4  | 3  | 3       | 4   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 4   | 4   | 4       | 4   | 4   | 4   |

Fuente. Propia de la investigación

**Tabla 32 Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación. Grupo B Ingeniería Electrónica y Telecomunicaciones**

| No. | Grupos<br>Criterios de Observación / Estudiantes:                               | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 3       | 3  | 4  | 3  | 3       | 4  | 4  | 4  | 3       | 3   | 3   | 3   | 3       | 1   | 3   | 4   | 4       | 4   | 3   | 3   | 3       | 4   | 3   |     |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 3       | 3  | 4  | 3  | 4       | 4  | 3  | 3  | 4       | 3   | 3   | 3   | 3       | 2   | 4   | 4   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 3       | 3  | 4  | 3  | 4       | 3  | 3  | 4  | 4       | 4   | 3   | 4   | 3       | 2   | 3   | 3   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 3       | 3  | 3  | 4  | 4       | 3  | 4  | 4  | 4       | 4   | 3   | 3   | 4       | 1   | 3   | 4   | 3       | 3   | 4   | 4   | 4       | 3   | 3   |     |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 4  | 4  | 3  | 3       | 4  | 3  | 3  | 3       | 3   | 3   | 3   | 3       | 1   | 3   | 4   | 3       | 3   | 4   | 3   | 3       | 3   | 3   |     |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 4       | 3  | 3  | 3  | 3       | 3  | 4  | 3  | 4       | 4   | 4   | 3   | 4       | 2   | 3   | 4   | 4       | 3   | 4   | 4   | 4       | 4   | 4   |     |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 4       | 3  | 3  | 3  | 3       | 3  | 3  | 3  | 3       | 4   | 3   | 4   | 3       | 4   | 2   | 3   | 4       | 4   | 3   | 3   | 4       | 3   | 4   |     |
| 8   | Está motivado para conocer los errores y corregirlos                            | 4       | 3  | 3  | 3  | 4       | 4  | 3  | 3  | 3       | 4   | 3   | 4   | 3       | 4   | 2   | 4   | 3       | 4   | 4   | 4   | 4       | 4   | 4   |     |
| 9   | Realiza actividades durante las clases.   | 3       | 3  | 3  | 4  | 4       | 3  | 4  | 4  | 3       | 3   | 4   | 4   | 4       | 1   | 4   | 3   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 10  | Participa en el trabajo en grupo  | 3       | 4  | 4  | 3  | 4       | 4  | 3  | 4  | 3       | 4   | 3   | 3   | 4       | 1   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 3   |     |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 4       | 3  | 3  | 3  | 4       | 4  | 3  | 3  | 4       | 4   | 3   | 3   | 2       | 4   | 3   | 3   | 3       | 3   | 4   | 4   | 4       | 4   | 3   |     |

Fuente. Propia de la investigación

**Tabla 33 Puntuación de los criterios relacionados con la motivación en el Protocolo de Observación.  
Grupo de Ingeniería Automática Industrial**

| No. | Criterios de Observación / Estudiantes:   | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 4       | 4  | 4  | 3  | 3       | 2  | 3  | 3  | 3       | 3   | 3   | 4   | 4       | 4   | 4   | 4   | 3       | 3   | 3   | 3   | 3       | 4   |     |     |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 3       | 3  | 3  | 3  | 3       | 1  | 3  | 4  | 4       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 3   | 3   | 3   | 3       | 3   |     |     |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 4       | 4  | 3  | 4  | 4       | 2  | 3  | 4  | 4       | 3   | 3   | 4   | 4       | 4   | 4   | 4   | 4       | 3   | 3   | 3   | 3       | 3   |     |     |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 4       | 4  | 3  | 4  | 4       | 2  | 3  | 4  | 4       | 4   | 3   | 3   | 3       | 3   | 3   | 3   | 3       | 3   | 4   | 3   | 3       | 3   |     |     |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 3  | 3  | 3  | 4       | 2  | 4  | 3  | 3       | 4   | 3   | 3   | 3       | 4   | 4   | 3   | 3       | 3   | 3   | 3   | 3       | 4   | 4   |     |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 3       | 4  | 4  | 4  | 3       | 1  | 4  | 3  | 3       | 4   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 4   | 3   | 4       | 3   | 4   | 4   |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 3       | 3  | 4  | 4  | 3       | 1  | 4  | 3  | 3       | 3   | 3   | 4   | 4       | 4   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   | 4   |     |
| 8   | Está motivado para conocer los errores y corregirlos                            | 3       | 3  | 4  | 4  | 3       | 2  | 3  | 4  | 3       | 4   | 3   | 3   | 3       | 3   | 3   | 4   | 3       | 4   | 3   | 4   | 3       | 4   | 3   | 4   |
| 9   | Realiza actividades durante las clases.   | 3       | 3  | 3  | 3  | 3       | 2  | 3  | 3  | 4       | 4   | 3   | 3   | 3       | 4   | 4   | 4   | 3       | 4   | 3   | 4   | 3       | 4   | 3   | 3   |
| 10  | Participa en el trabajo en grupo  | 3       | 4  | 4  | 3  | 3       | 2  | 3  | 3  | 4       | 3   | 4   | 3   | 3       | 3   | 3   | 4   | 4       | 4   | 4   | 3   | 3       | 3   | 3   |     |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 3       | 4  | 4  | 3  | 4       | 1  | 4  | 3  | 3       | 3   | 3   | 3   | 3       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 3       | 3   | 3   |     |

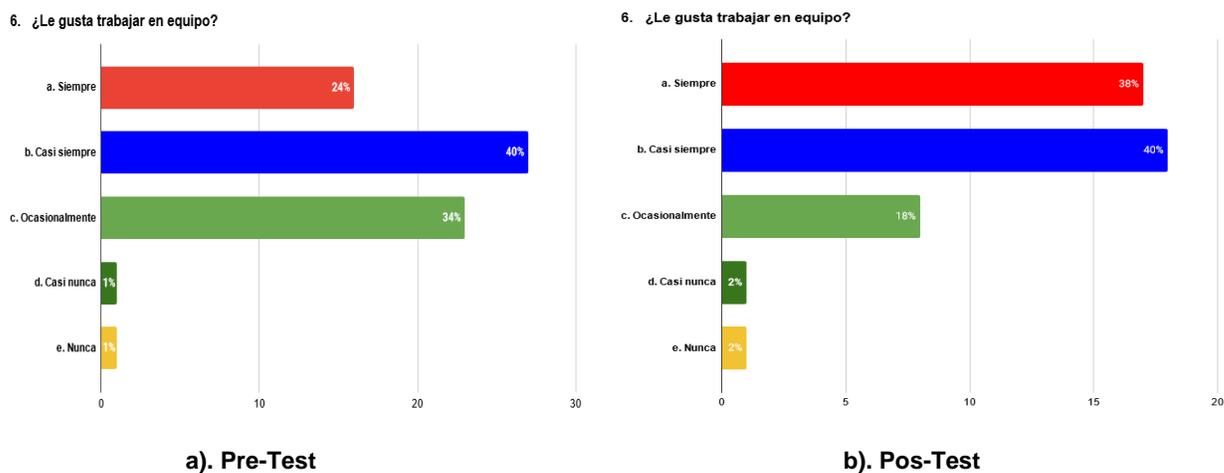
Fuente. Propia de la investigación

En los resultados de las **Tablas 31, 32 y 33**, se muestra que hay tres participantes con un puntaje menor o igual a 2 en ambos criterios (se puede observar en casi todas las situaciones), los demás participantes tienen un puntaje de 3 o 4. Para los dos criterios de observación el rango de los promedios va desde 2 hasta 4.

### Participación.

Con el fin de medir la participación de los estudiantes en la solución de las misiones en grupo se valoró su disposición para trabajar en equipo; para ello, se analizó la pregunta 6 realizada tanto en el Pre-test como en Pos-test.

**Figura 30. Distribución de las respuestas a la pregunta N.6 a) en el Pre-test y b) en el Pos-test de Fase exploratoria del estudio de caso descriptivo**



Fuente. Propia de la investigación

En lo concerniente a los resultados obtenidos en la pregunta numero 6 mostrados en la **Figura 30** se evidenció la preferencia por trabajar en equipo, observándose que el porcentaje de estudiantes que afirmo su gusto por siempre trabajar en equipo paso de un 24% a un 38%, el de estudiantes que dijo casi siempre se mantuvo estable con un 40% y el de estudiantes que dijo ocasionalmente pasó de un 34% a un 18%. Observándose, en general, un aumento en la preferencia por el trabajo en equipo.

En cuanto a los campos relacionados con la participación de los estudiantes en el estudio se evidencia, en la aplicación del Protocolo de Observación los resultados que se muestran en las **Tablas 34, 35, 36**.

**Tabla 34 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo A Ingeniería Electrónica y Telecomunicaciones**

| No. | Criterios de Observación / Estudiantes:   | Grupos  |    |    |    |         |    |    |    |         |     |     |     |         |     |     |     |         |     |     |     |         |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 3       | 4  | 1  | 4  | 3       | 4  | 3  | 4  | 3       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   | 4   | 3   | 3       | 3   | 3   | 3   |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 4       | 4  | 2  | 4  | 3       | 4  | 3  | 4  | 3       | 3   | 3   | 3   | 4       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   | 3   | 3   |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 4       | 4  | 1  | 4  | 3       | 3  | 4  | 4  | 3       | 4   | 4   | 4   | 4       | 4   | 4   | 3   | 3       | 4   | 3   | 3   | 3       | 4   | 4   | 4   |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 4       | 4  | 2  | 3  | 3       | 3  | 4  | 4  | 4       | 4   | 4   | 3   | 4       | 3   | 3   | 3   | 3       | 4   | 3   | 4   | 3       | 4   | 3   | 4   |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 3  | 2  | 3  | 4       | 3  | 4  | 3  | 3       | 3   | 4   | 3   | 4       | 3   | 4   | 3   | 4       | 3   | 3   | 4   | 3       | 3   | 3   | 4   |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 3       | 3  | 2  | 3  | 4       | 4  | 4  | 3  | 4       | 3   | 3   | 4   | 4       | 3   | 3   | 3   | 4       | 3   | 3   | 4   | 3       | 4   | 3   | 3   |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 3       | 4  | 1  | 3  | 4       | 4  | 3  | 3  | 4       | 3   | 4   | 4   | 4       | 3   | 3   | 3   | 4       | 4   | 3   | 3   | 4       | 4   | 3   | 3   |
| 8   | Está motivado para conocer los errores y corregirlos                            | 4       | 4  | 1  | 4  | 4       | 3  | 3  | 3  | 4       | 3   | 4   | 3   | 4       | 4   | 4   | 4   | 4       | 4   | 4   | 3   | 4       | 4   | 3   | 4   |
| 9   | Realiza actividades durante las clases.   | 4       | 3  | 2  | 4  | 3       | 4  | 4  | 3  | 4       | 3   | 3   | 3   | 3       | 4   | 3   | 4   | 4       | 4   | 4   | 4   | 3       | 4   | 4   | 3   |
| 10  | Participa en el trabajo en grupo  | 4       | 3  | 2  | 4  | 3       | 3  | 4  | 3  | 3       | 4   | 4   | 4   | 3       | 4   | 3   | 4   | 3       | 4   | 4   | 3   | 3       | 3   | 3   | 4   |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 4       | 3  | 2  | 4  | 3       | 4  | 4  | 3  | 3       | 4   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 3   | 4   | 4       | 4   | 4   | 4   |

Fuente. Propia de la investigación

**Tabla 35 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo B Ingeniería Electrónica y Telecomunicaciones**

| No. | Criterios de Observación / Estudiantes:   | Grupos  |    |    |    |         |    |    |    |         |     |     |     |         |     |     |     |         |     |     |     |         |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 3       | 3  | 4  | 3  | 3       | 4  | 4  | 4  | 3       | 3   | 3   | 3   | 3       | 1   | 3   | 4   | 4       | 4   | 3   | 3   | 3       | 4   | 3   |     |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 3       | 3  | 4  | 3  | 4       | 4  | 3  | 3  | 4       | 3   | 3   | 3   | 3       | 2   | 4   | 4   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 3       | 3  | 4  | 3  | 4       | 3  | 3  | 4  | 4       | 4   | 3   | 4   | 3       | 2   | 3   | 3   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 3       | 3  | 3  | 4  | 4       | 3  | 4  | 4  | 4       | 4   | 3   | 3   | 4       | 1   | 3   | 4   | 3       | 3   | 4   | 4   | 4       | 4   | 3   |     |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 4  | 4  | 4  | 3       | 3  | 4  | 3  | 3       | 3   | 3   | 3   | 3       | 1   | 3   | 4   | 3       | 3   | 4   | 3   | 3       | 3   | 3   |     |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 4       | 3  | 3  | 3  | 3       | 3  | 4  | 3  | 4       | 4   | 4   | 3   | 4       | 2   | 3   | 4   | 4       | 3   | 4   | 4   | 4       | 4   | 4   |     |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 4       | 3  | 3  | 3  | 3       | 3  | 3  | 3  | 3       | 4   | 3   | 4   | 3       | 4   | 2   | 3   | 4       | 4   | 3   | 3   | 4       | 3   | 4   |     |
| 8   | Está motivado para conocer los errores y corregirlos                            | 4       | 3  | 3  | 3  | 4       | 4  | 3  | 3  | 3       | 4   | 3   | 4   | 2       | 4   | 3   | 4   | 4       | 4   | 4   | 4   | 4       | 4   | 4   |     |
| 9   | Realiza actividades durante las clases.   | 3       | 3  | 3  | 4  | 4       | 3  | 4  | 4  | 3       | 3   | 4   | 4   | 4       | 1   | 4   | 3   | 3       | 4   | 4   | 3   | 4       | 3   | 3   |     |
| 10  | Participa en el trabajo en grupo  | 3       | 4  | 4  | 3  | 4       | 4  | 3  | 4  | 3       | 4   | 3   | 3   | 4       | 1   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 3   |     |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 4       | 3  | 3  | 3  | 4       | 4  | 3  | 3  | 3       | 4   | 4   | 3   | 3       | 2   | 4   | 3   | 3       | 3   | 3   | 4   | 4       | 4   | 3   |     |

Fuente. Propia de la investigación.

Tabla 36 Puntuación de los criterios relacionados con la participación en el Protocolo de Observación. Grupo de Ingeniería Automática Industrial

| No. | Grupos<br>Criterios de Observación / Estudiantes:                               | Grupo 1 |    |    |    | Grupo 2 |    |    |    | Grupo 3 |     |     |     | Grupo 4 |     |     |     | Grupo 5 |     |     |     | Grupo 6 |     |     |     |
|-----|---|---------|----|----|----|---------|----|----|----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|---------|-----|-----|-----|
|     |   | P1      | P2 | P3 | P4 | P5      | P6 | P7 | P8 | P9      | P10 | P11 | P12 | P13     | P14 | P15 | P16 | P17     | P18 | P19 | P20 | P21     | P22 | P23 | P24 |
| 1   | Coopera con otros miembros del grupo durante las actividades de forma positiva. | 4       | 4  | 4  | 3  | 3       | 2  | 3  | 3  | 3       | 3   | 3   | 4   | 4       | 4   | 4   | 4   | 3       | 3   | 3   | 3   | 3       | 4   |     |     |
| 2   | Está listo para realizar tareas adicionales para mejorar los logros.            | 3       | 3  | 3  | 3  | 1       | 3  | 4  | 4  | 3       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 3   | 3   | 3   | 3       | 3   |     |     |
| 3   | Obedece las reglas de comportamiento durante los proyectos.                     | 4       | 4  | 3  | 4  | 2       | 3  | 4  | 4  | 3       | 3   | 4   | 4   | 4       | 4   | 4   | 4   | 4       | 3   | 3   | 3   | 3       | 3   |     |     |
| 4   | Conoce el objetivo de la tarea y cómo conseguirlo                               | 4       | 4  | 3  | 4  | 2       | 3  | 4  | 4  | 4       | 3   | 3   | 3   | 3       | 3   | 3   | 3   | 3       | 3   | 4   | 3   | 3       | 3   |     |     |
| 5   | Resuelve los problemas propuestos por sí mismo                                  | 3       | 3  | 3  | 3  | 2       | 4  | 3  | 3  | 4       | 3   | 3   | 3   | 3       | 4   | 4   | 3   | 3       | 3   | 3   | 3   | 4       | 4   |     |     |
| 6   | Resuelve situaciones problemáticas de forma tranquila                           | 3       | 4  | 4  | 4  | 1       | 4  | 3  | 3  | 4       | 3   | 3   | 3   | 3       | 4   | 3   | 3   | 3       | 3   | 4   | 3   | 4       | 4   |     |     |
| 7   | Está motivado para superar las dificultades en la realización de tareas         | 3       | 3  | 4  | 3  | 1       | 4  | 3  | 3  | 3       | 3   | 4   | 4   | 4       | 4   | 3   | 3   | 3       | 4   | 4   | 4   | 4       | 4   |     |     |
| 8   | Está motivado para conocer los errores y corregirlos                            | 3       | 3  | 4  | 4  | 2       | 3  | 4  | 3  | 4       | 3   | 3   | 3   | 3       | 3   | 3   | 4   | 3       | 4   | 3   | 4   | 3       | 4   |     |     |
| 9   | Realiza actividades durante las clases.   | 3       | 3  | 3  | 3  | 2       | 3  | 3  | 4  | 4       | 3   | 3   | 3   | 3       | 4   | 4   | 4   | 3       | 4   | 3   | 4   | 3       | 3   |     |     |
| 10  | Participa en el trabajo en grupo  | 3       | 4  | 4  | 3  | 2       | 3  | 3  | 4  | 3       | 4   | 3   | 3   | 3       | 3   | 3   | 4   | 4       | 4   | 3   | 3   | 3       | 3   |     |     |
| 11  | Ayuda a sus compañeros a resolver problemas                                     | 3       | 4  | 4  | 3  | 4       | 1  | 4  | 3  | 3       | 3   | 3   | 3   | 3       | 3   | 3   | 3   | 3       | 4   | 4   | 3   | 3       | 3   |     |     |

Fuente. Propia de la investigación.

En los resultados de las **Tablas 34, 35 y 36**, se muestra que tres de los participantes obtuvieron un puntaje menor o igual a 2 en los cuatro criterios (se puede observar en casi todas las situaciones), los demás participantes tienen un puntaje de 3 o 4. Para los cuatro criterios de observación el rango de los promedios va desde 2 hasta 4.

En promedio, según se desprende de las tablas, los alumnos tienen un rango de participación de 2 a 4, la clasificación del puntaje para este protocolo se encuentra en el **Anexo E**

### Apropiación del CT.

Para medir la apropiación del CT, se tuvieron en cuenta las preguntas relacionadas en las **Tablas 23 y 24**. Dada su naturaleza de preguntas abiertas, eran respondidas por los estudiantes con sus propias palabras. En este punto, se evaluó cada uno de los conceptos de CT teniendo como base las definiciones del modelo Atmatzidou y Demetriadis [55]. Al que se adaptó la generalización al reconocimiento de patrones, para obtener el modelo adaptado que se empleó en este proyecto y que se muestra en la **Tabla 37**.

**Tabla 37 Nivel de Competencia para cada aspecto del pensamiento computacional**

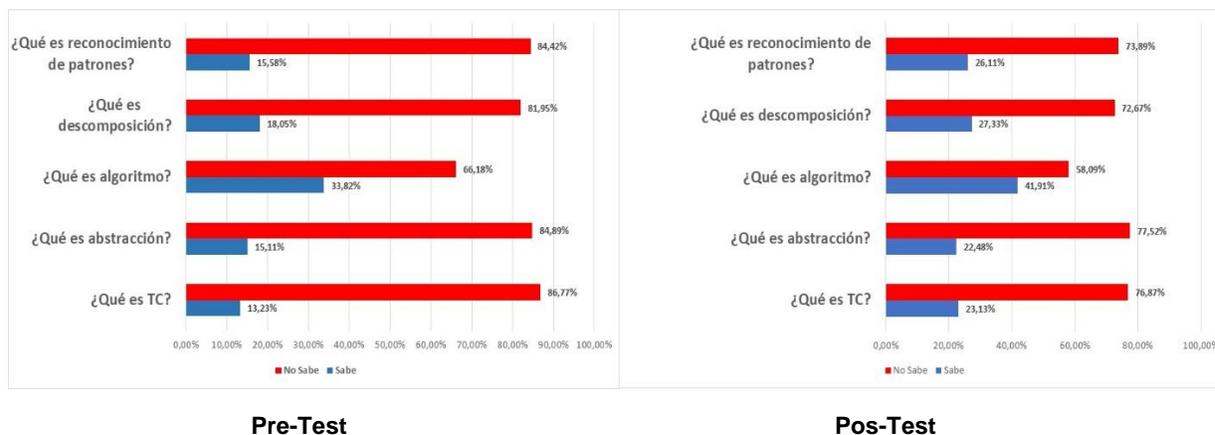
| Habilidad de CT   | Definición  | Guía para su desarrollo   |
|---|---|---|
| Abstracción   | Es el proceso de crear algo simple a partir de algo complicado, omitiendo los detalles irrelevantes, encontrando los patrones relevantes y separando ideas de los detalles tangibles. | <ol style="list-style-type: none"> <li>1. Separar la información importante de la redundante.</li> <li>2. Analizar y especificar comportamientos comunes o estructuras de programación entre scripts diferentes.</li> <li>3. Identificación de abstracciones entre ambientes de programación diferentes.</li> </ol> |
| Generalización<br>(Adaptado a reconocimiento de patrones) | Es transferir un proceso de solución de un problema a una amplia variedad de problemas.   | <ol style="list-style-type: none"> <li>1. Expandir una solución existente en un problema dado para cubrir más posibilidades / casos.</li> </ol>   |
| Algoritmos  | Es la práctica de escribir, paso a paso, instrucciones específicas y  | <ol style="list-style-type: none"> <li>1. Describir explícitamente los pasos del algoritmo.</li> </ol>  |

|                |  |  |
|----------------|--|--|
|                | explícitas, para llevar a cabo un proceso.   | 2. Posibilidad de diferentes algoritmos para el mismo problema.<br>3. Esfuerzo para encontrar el algoritmo más efectivo. |
| Descomposición | Es el proceso de separar los problemas en partes más pequeñas que pueden resolverse con mayor facilidad. | Separar problemas en partes más pequeñas / sencillas que son más fáciles de resolver.                                    |

Fuente. Modelo Atmatzidou y Demetriadis [55]

En la **Figura 31** se resume la evaluación de los conocimientos de los conceptos de CT en cada uno de los tests aplicados en la fase exploratoria del estudio de caso descriptivo, tanto en el Pre-test, como en el Pos-test.

**Figura 31. Comparativo a) Pre-test y b) Pos-test sobre la apropiación de los conceptos de CT**



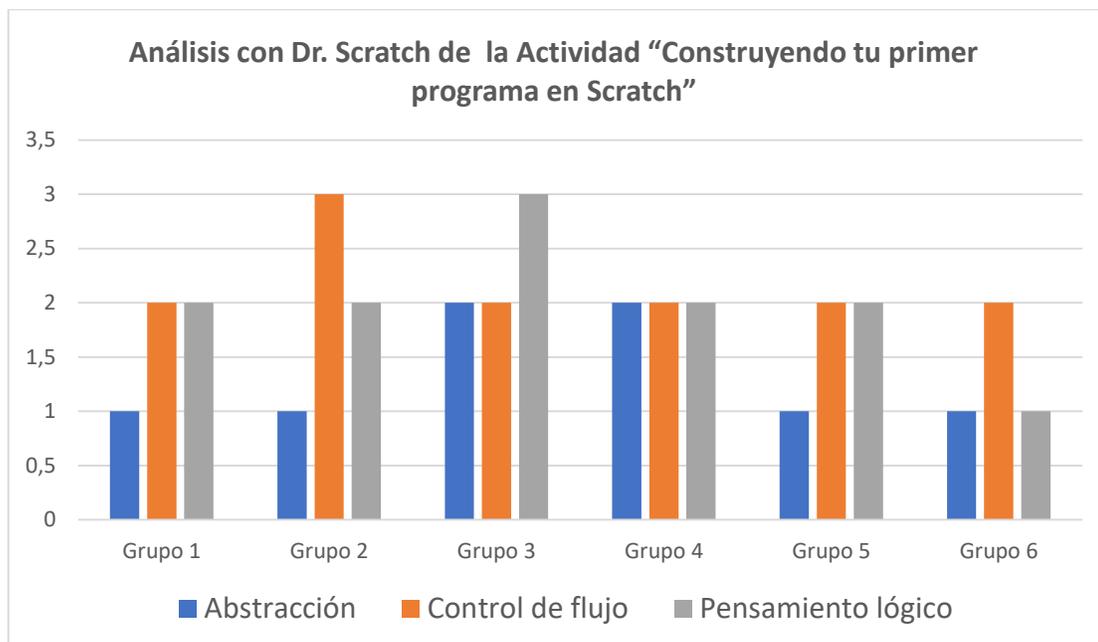
Fuente. Propia de la investigación

Las gráficas mostraron que al iniciar el curso había un bajo porcentaje de conocimiento sobre los conceptos de CT y reconocimiento de patrones, en su mayoría desconocían los conceptos de algoritmos, descomposición, abstracción y pensamiento computacional. Los resultados del Pos-test muestran una mejora en la apropiación de dichos conceptos.

Al mismo tiempo se analizó sobre la identificación de los conceptos de CT, se investigó sobre la aplicación de las habilidades de CT. Para lo cual, se utilizó la herramienta Dr. Scratch para validar el código entregado por los equipos.

Para identificar si los alumnos además de reconocer los conceptos CT aplican de manera empírica algunas habilidades de CT, se hizo el análisis de las mediciones de Dr. Scratch. La comparación entre los códigos de los equipos, los resultados de analizar los respectivos códigos con Dr. Scratch, se muestra en la **Figura 32**.

**Figura 32. Resultados del análisis con Dr. Scratch del código de a) la sesión 2. “Construyendo tu primer programa en Scratch”**



**Fuente. Propia de la investigación**

De acuerdo con los resultados de la **Figura 32**, fueron tres los aspectos del CT evaluados con la herramienta Dr. Scratch: pensamiento lógico, control de flujo y abstracción.

A continuación, se muestran en la **Tabla 38** las habilidades según la evaluación de la herramienta Dr. Scratch.

**Tabla 38 Habilidades de Pensamiento Computacional, según son descritas por la herramienta Dr. Scratch y su aplicación al fase exploratoria**

| Habilidad de CT    | Criterios de competencia Dr. Scratch   | Aplicación al estudio                         |
|--------------------|--|---|
| Abstracción        | Abstraer consiste en dividir un problema (o reto, o desafío) grande, en problemas pequeños y de fácil solución.                                  | Mide la abstracción de la solución.           |
| Control de flujo   | Orden de ejecución de las instrucciones y uso de operadores de control (uso de operadores condicionales)   | Está relacionado con el diseño del algoritmo. |
| Pensamiento lógico | Permite reconocer como es la calidad de la solución al problema planteado, según su codificación en boques. (uso de operadores, repetir, hasta ) | Permite medir el desarrollo del CT            |

**Fuente. Propia de la investigación**

Por último, se efectuó el análisis de las entrevistas realizadas a los docentes, acerca de la metodología utilizada para enseñar la materia Introducción a la Ingeniería, y el respectivo programa usado para el desarrollo de dicha asignatura. En esta entrevista se encontró que

dos de los tres docentes utilizaron nuevas metodologías como es el caso de Design Thinking. El cual es un método para generar ideas innovadoras que centra su eficacia en entender y dar solución a las necesidades reales de los usuarios. A través de esta se prepara a los estudiantes para la elaboración de proyectos de ingeniería. Lo que fue un aporte importante en el diseño del modelo CT4E, ya que este permite integrar el Desarrollo del Pensamiento Computacional en las actividades formativas en Ingeniería (T&E) en los primeros semestres de la carrera, junto a los conceptos de STEM, en el contexto del enfoque metodológico de ChildProgramming.

#### 4.2.11 Ejecución, resultados y análisis de la fase comparativa de estudio de caso descriptivo.

La fase comparativa del estudio de caso descriptivo se realizó en el periodo comprendido entre el 1 de diciembre del 2022 al 18 de enero del 2023, en la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Se realizaron cinco sesiones de 2 horas cada una, a los estudiantes de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial. En las cuales se efectuaron las siguientes actividades encaminadas a la aplicación y evaluación de los instrumentos diseñados en el modelo CT4E, las sesiones se dividieron en dos partes de una hora cada una para realizar las siguientes actividades, la primera correspondió a la capacitación con la herramienta de programación Scratch basada en el marco conceptual que analiza el desarrollo del pensamiento computacional desde tres dimensiones clave: conceptos, prácticas y perspectivas computacionales propuesto por Brennan & Resnick (2012) [34]. Y la segunda a la aplicación del modelo, como se describen a continuación: Curso Scratch, Pre-Test, actividad de Identificación y Formulación del Problema, actividad de Formulación de Soluciones, actividad de Evaluación por Prototipos, por último, se presentó el Post-Test. A continuación, se describen las sesiones. tal y como se indica en las **Tabla 39** a la **Tabla 47**.

#### Sesión 1 Actividad de evaluación (Pre-Test) y Scratch Unidad 1: Direcciones.

Tabla 39. Actividad Evaluación CT (Pre-Test)

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | CT4E05  |
| <b>Nombre práctica:</b> | Pre-Test  |
| <b>Dirigida a</b>       | Docentes de Introducción a la ingeniería  |
| <b>Contexto:</b>        | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial. |
| <b>Descripción:</b>     | Aplicación del Test diseñado por Marcos Román González, el cual evalúa los conocimientos sobre pensamiento computacional.               |

|  |  |  |
|--|--|--|
| <b>Tareas a realizar:</b>  | Control de asistencia<br>Organizar a los estudiantes en la sala de computo.<br>Recomendaciones para el desarrollo de la actividad<br>Entrega del link para ingreso al Test |  |
| <b>Requisitos de entrada:</b>  |  | <b>Resultados obtenidos:</b>   |
| <b>El Pre-Test fase Comparativa (véase el Anexo G):</b><br><a href="https://forms.gle/QnSyfhUqffoJQ5WY9">https://forms.gle/QnSyfhUqffoJQ5WY9</a> [65]. |  | Test diligenciado en la plataforma de Google por cada uno de los participantes |

Fuente. Propia de la investigación

Tabla 40. Actividad de Scratch Unidad 1: Direcciones

|   |  |   |
|---|--|---|
| <b>Identificador:</b>   | <b>CT4E04</b>  |   |
| <b>Nombre práctica:</b>   | Actividad de Scratch Unidad 1: Direcciones   |   |
| <b>Descripción:</b>   | En la presente unidad se abordará el concepto computacional denominado Dirección, con el fin de comprender la forma en que se puede desplazar, un personaje, atendiendo a las direcciones y las instrucciones de las generar.            |   |
| <b>Dirigida a:</b>  | Estudiantes  |   |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial   |   |
| <b>Tareas a realizar:</b>   | Se realizarán las siguientes actividades: <ul style="list-style-type: none"> <li>• Paso a paso: Direcciones</li> <li>• Aprendiendo a orientarse</li> <li>• Ubicación y orientación</li> <li>• Reto: Resolviendo el laberinto.</li> </ul> |   |
| <b>Proceso:</b>   |  |   |
| <b>Requisitos de entradas:</b>  |  | <b>Resultados esperados:</b>  |
| <ul style="list-style-type: none"> <li>• Sala de computo.</li> <li>• Programa Scratch</li> <li>• Presentación PowerPoint</li> </ul> |  | Asimilación de los estudiantes de los conceptos computacionales de dirección. |
| <b>Sugerencias:</b>   |  |   |
| Es importante que los equipos de cómputo con la actividad estén listos antes de iniciar.  |  |   |

Fuente. Propia de la investigación

## Sesión 2 Scratch Unidad 2: Bucles y Actividad de Identificación y formulación del problema.

Tabla 41. Actividad de Scratch Unidad 2: Bucles

|   |   |  |
|---|---|--|
| <b>Identificador:</b>   | <b>CT4E04</b>   |  |
| <b>Nombre práctica:</b>   | Actividad de Scratch Unidad 2: Bucles   |  |
| <b>Descripción:</b>   | En la presente unidad se aborda el concepto computacional denominado Bucles(Ciclos). Las actividades propuestas apuntan a dicho concepto y la forma como se usan estas para realizar labores repetitivas.                     |  |
| <b>Dirigida a:</b>  | Estudiantes   |  |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial  |  |
| <b>Tareas a realizar:</b>   | <p>Se realizarán las siguientes actividades:</p> <ul style="list-style-type: none"> <li>• Paso a Paso: Bucles</li> <li>• Repitiendo Instrucciones</li> <li>• Utilización de Bucles</li> <li>• Reto: Dando vueltas.</li> </ul> |  |
| <b>Proceso:</b>   |   |  |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>  |  |
| <ul style="list-style-type: none"> <li>• Sala de computo.</li> <li>• Programa Scratch</li> <li>• Presentación PowerPoint</li> </ul> | Asimilación de los estudiantes de los conceptos computacionales de Bucles.  |  |
| <b>Sugerencias:</b>   |   |  |
| Es importante que los equipos de cómputo con la activad estén listos antes de iniciar.  |   |  |

Fuente. Propia de la investigación

Tabla 42. Identificación y Formulación del Problema

|                           |   |
|---------------------------|---|
| <b>Identificador:</b>     | <b>CT4E01</b>   |
| <b>Nombre práctica:</b>   | Identificación y Formulación del Problema   |
| <b>Descripción:</b>       | Los estudiantes a través de una situación real buscan identificar y formular el problema.   |
| <b>Dirigida a:</b>        | Estudiantes.  |
| <b>Contexto:</b>          | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial  |
| <b>Tareas a realizar:</b> | <ol style="list-style-type: none"> <li>1. Tareas de planeación y de gestión del trabajo diario del equipo a través del tablero de tareas</li> <li>2. Lectura y discusión de la situación.</li> <li>3. Identificación de problemas específicos en el contexto problemático a través de una lluvia de ideas.</li> </ol> |

|   |   |
|---|---|
|   | <ol style="list-style-type: none"> <li>4. Clasificar los problemas identificados por relevancia.</li> <li>5. Seleccionar los problemas relevantes.</li> <li>6. Priorizar los problemas seleccionados</li> <li>7. Elegir el problema de mayor prioridad</li> <li>8. Descripción del problema:</li> </ol> |
| <b>Proceso:</b>   |   |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>  |
| Equipos conformados.<br>Ficha de Identificación y Selección del Problema (Anexo I <sup>26</sup> )<br>Ficha de Formulación y Planteamiento del Problema (Anexo J <sup>27</sup> ) | Problema identificado y descrito<br>Ficha de Identificación y Selección del Problema (Diligenciada)<br>Ficha de Formulación y Planteamiento del Problema (Diligenciada)   |
| <b>Sugerencias:</b>   |   |
|   |   |

Fuente. Propia de la investigación

### Sesión 3 Scratch Unidad 3: Condicionales y actividad de formulación de soluciones.

Tabla 43. Actividad de Scratch Unidad 3: Condicionales

|   |   |
|---|---|
| <b>Identificador:</b>   | <b>CT4E04</b>   |
| <b>Nombre práctica:</b>   | Actividad de Scratch Unidad 3: Condicionales  |
| <b>Descripción:</b>   | En la presente sección se aborda el concepto computacional denominado condicionales. Las actividades propuestas apuntan a dicho concepto computacional y la forma en que se utilizan los condicionales para toma de decisiones propuestas.    |
| <b>Dirigida a:</b>  | Estudiantes   |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial  |
| <b>Tareas a realizar:</b>   | Se realizarán las siguientes actividades: <ul style="list-style-type: none"> <li>• Paso a paso: condicionales</li> <li>• Tomando decisiones</li> <li>• Utilización de condicionales</li> <li>• Reto: Moviéndose en las direcciones</li> </ul> |
| <b>Proceso:</b>   |   |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>  |
| <ul style="list-style-type: none"> <li>• Sala de computo.</li> <li>• Programa Scratch</li> <li>• Presentación PowerPoint</li> </ul> | Asimilación de los estudiantes de los conceptos computacionales de Condicionales.   |

<sup>26</sup>26 Ficha diseñada siguiendo los lineamientos de la metodología de Mario Bunge [7]

<sup>27</sup>27 Ficha diseñada con base en el método AQP y CCA de Rosario Martínez[60]

| <b>Sugerencias:</b>  |
|--|
| Es importante que los equipos de cómputo con la actividad estén listos antes de iniciar. |

**Fuente. Propia de la investigación**

**Tabla 44. Actividad de Formulación de Soluciones**

|   |  |  |
|---|--|--|
| <b>Identificador:</b>   | <b>CT4E02</b>  |  |
| <b>Nombre práctica:</b>   | Formulación de Soluciones  |  |
| <b>Descripción:</b>   | Los estudiantes a través del problema identificado y descrito en la Ficha de Formulación y Planteamiento del Problema (Anexo j <sup>28</sup> ) plantean la solución.   |  |
| <b>Dirigida a:</b>  | Estudiantes.   |  |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial   |  |
| <b>Tareas a realizar:</b>   | <ol style="list-style-type: none"> <li>1. Tareas de planeación y de gestión del trabajo diario del equipo a través del tablero de tareas</li> <li>2. Lectura y análisis del problema identificado.</li> <li>3. Identificación de las posibles soluciones a través de una lluvia de ideas.</li> <li>4. Clasificación de las soluciones identificadas por relevancia.</li> <li>5. Selección de las soluciones relevantes y alcanzables.</li> <li>6. Evaluación y priorización de las soluciones seleccionadas.</li> <li>7. Elección de la solución con mayor prioridad.</li> <li>8. Descripción de la solución.</li> </ol> |  |
| <b>Proceso:</b>   |  |  |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>   |  |
| Equipos conformados<br>Problema identificado y descrito<br>Ficha de Identificación y Selección del Problema (Diligenciada) Anexo I.<br>Ficha de Formulación y Planteamiento del Problema (Diligenciada) Anexo j.<br>Ficha Formulación de Solución Anexo k <sup>29</sup> | Solución Priorizada y descrita<br>Ficha Formulación de Soluciones (Diligenciada) Anexo k.  |  |
| <b>Sugerencias:</b>   |  |  |
|   |  |  |

**Fuente. Propia de la investigación**

<sup>28</sup> Ficha diseñada con base en el método AQP y CCA de Rosario Martínez[60]

<sup>29</sup> Ficha diseñada siguiendo los lineamientos de la metodología de Mario Bunge [7]

## Sesión 4 Scratch Unidad 4: Funciones y actividad de evaluación por prototipos.

Tabla 45. Actividad de Scratch Unidad 4: Funciones

|   |   |  |
|---|---|--|
| <b>Identificador:</b>   | CT4E04  |  |
| <b>Nombre práctica:</b>   | Actividad de Scratch Unidad 4: Funciones  |  |
| <b>Descripción:</b>   | En la presente sección se aborda el concepto computacional denominado funciones las actividades propuestas apuntan a dicho concepto computacional y la forma en la que se utilizan las funciones para simplificar procesos en aras de proponer soluciones a situaciones propuestas. |  |
| <b>Dirigida a:</b>  | Estudiantes   |  |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial  |  |
| <b>Tareas a realizar:</b>   | Se realizarán las siguientes actividades: <ul style="list-style-type: none"> <li>• Paso a paso: Funciones</li> <li>• Trabajando con funciones</li> <li>• Funciones</li> <li>• Reto: Construyendo funciones.</li> </ul>  |  |
| <b>Proceso:</b>   |   |  |
| <b>Requisitos de entradas:</b>  | <b>Resultados esperados:</b>  |  |
| <ul style="list-style-type: none"> <li>• Sala de computo.</li> <li>• Programa Scratch</li> <li>• Presentación PowerPoint</li> </ul> | Asimilación de los estudiantes de los conceptos computacionales de Funciones.   |  |
| <b>Sugerencias:</b>   |   |  |
| Es importante que los equipos de cómputo con la actividad estén listos antes de iniciar.  |   |  |

Fuente. Propia de la investigación

Tabla 46. Actividad de Evaluación por Prototipos

|                                |  |  |
|--------------------------------|--|--|
| <b>Identificador:</b>          | CT4E03   |  |
| <b>Nombre práctica:</b>        | Evaluación por Prototipos  |  |
| <b>Descripción:</b>            | Los estudiantes evalúan por prototipo software la solución planteada en la Ficha de Formulación de Solución Anexo k <sup>30</sup>  |  |
| <b>Dirigida a:</b>             | Docentes e instructores, Estudiantes.  |  |
| <b>Contexto:</b>               | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial.  |  |
| <b>Tareas a realizar:</b>      | <ol style="list-style-type: none"> <li>1. Tareas de planeación y de gestión del trabajo diario del equipo a través del tablero de tareas</li> <li>2. Desarrollo del prototipo</li> <li>3. Prueba y depuración del prototipo</li> <li>4. Demostración del prototipo al cierre de una ronda</li> </ol> |  |
| <b>Proceso:</b>                |  |  |
| <b>Requisitos de entradas:</b> | <b>Resultados esperados:</b>   |  |

<sup>30</sup> Ficha diseñada siguiendo los lineamientos de la metodología de Mario Bunge [7]

|   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Ficha Formulación de Soluciones (Diligenciada) Anexo k.</li> <li>• Ficha para la Evaluación por Prototipos Anexo L.</li> <li>• Equipos conformados</li> <li>• Docente atento y dispuesto a asesorar a los equipos de trabajo.</li> </ul> | <ul style="list-style-type: none"> <li>• Algoritmos y códigos de programación de cada equipo de trabajo.</li> <li>• Los estudiantes plantean la solución al problema.</li> </ul> |
| <b>Sugerencias:</b>   |  |
|   |  |

Fuente. Propia de la investigación

## Sesión 5 Actividad de evaluación (Pos-Test).

Tabla 47. Actividad Evaluación CT (Pos-Test)

|   |  |   |
|---|--|---|
| <b>Identificador:</b>   | CT4E05   |   |
| <b>Nombre práctica:</b>   | Post-Test  |   |
| <b>Dirigida a</b>   | Docentes de Introducción a la ingeniería   |   |
| <b>Contexto:</b>  | Curso de introducción a la ingeniería de la facultad de ingeniería electrónica y telecomunicaciones e ingeniería automática industrial.  |   |
| <b>Descripción:</b>   | Aplicación del Test diseñado por Marcos Román González, el cual evalúa los conocimientos sobre pensamiento computacional.  |   |
| <b>Tareas a realizar:</b>   | <ol style="list-style-type: none"> <li>1. Control de asistencia</li> <li>2. Organizar a los estudiantes en la sala de computo.</li> <li>3. Recomendaciones para el desarrollo de la actividad</li> <li>4. Entrega del link para ingreso al Test</li> </ol> |   |
| <b>Requisitos de entrada:</b>   |  | <b>Resultados obtenidos:</b>  |
| El Pos-Test fase exploratoria (véase el Anexo H): <a href="https://forms.gle/pevzWm7CFGTHDR3DA">https://forms.gle/pevzWm7CFGTHDR3DA</a> [65]. |  | Post-Test diligenciado en la plataforma de Google por cada uno de los participantes |

Fuente. Propia de la investigación

### 4.2.11.1 Resultados cuantitativos.

Los resultados cuantitativos fueron obtenidos del análisis de las mediciones directas de los test aplicados. Entre estas encontramos las referentes Concepto computacional, Tarea Requerida, Capacidad Evaluada.

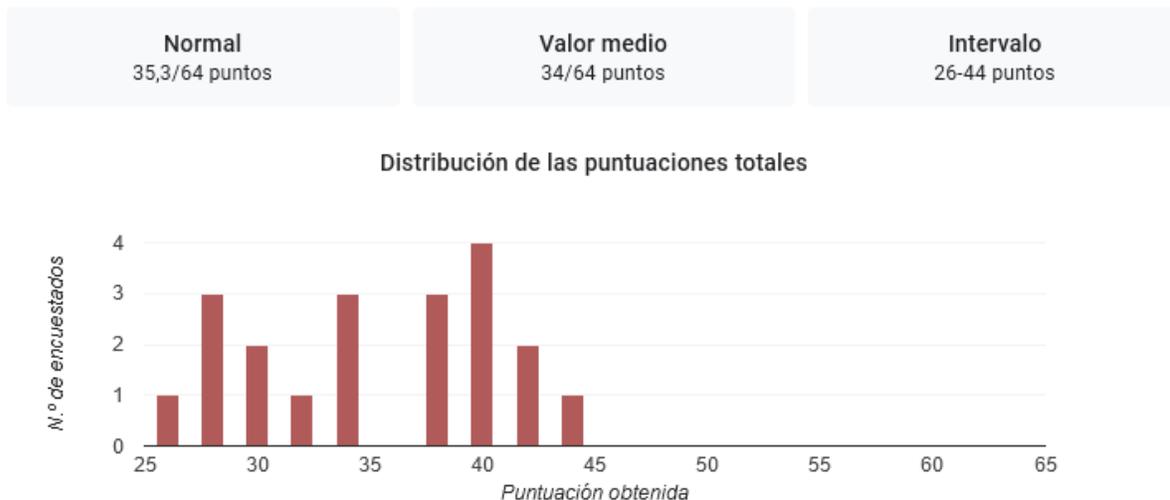
#### Concepto Computacional.

Con el fin de medir la variación en la asimilación de los conceptos de pensamiento computacional de los participantes entre el inicio y el final de la fase comparativa del estudio

de caso descriptivo, se hizo un paralelo entre los resultados del Pre-test y del Pos-test, tomando las 28 preguntas como referente.

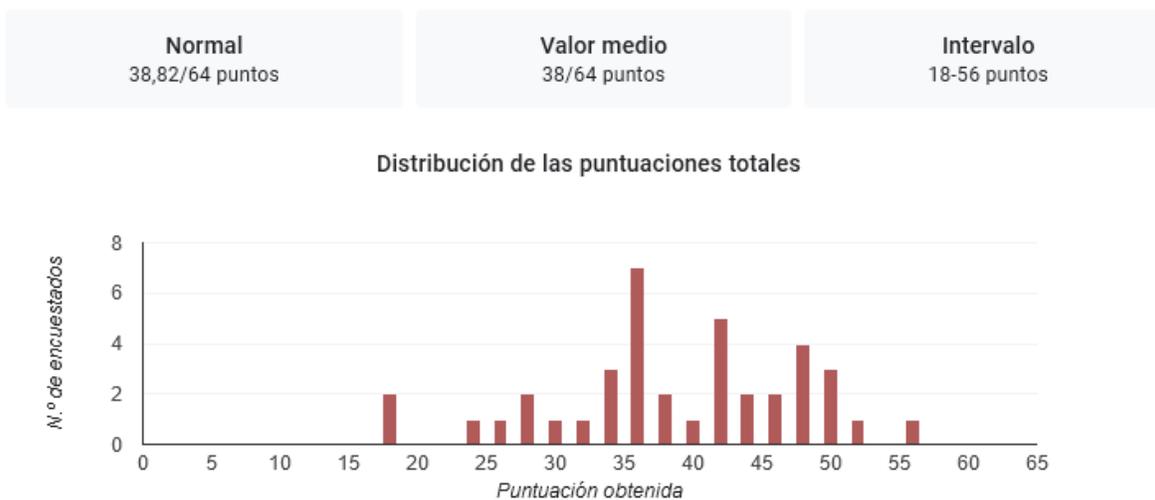
En ambos casos, se indagó directamente por la asimilación de los conceptos de pensamiento computacional en los estudiantes que participaron del estudio. obteniéndose los resultados que aparecen en la **Figura 33 y 34**.

**Figura 33 Cantidad de aciertos Pre Test**



Fuente. Propia de la investigación

**Figura 34 Cantidad de aciertos Pos-Test**



Fuente. Propia de la investigación

Al observar el rango del Post Test, se evidencia un incremento de 4 puntos aproximadamente en la media de la puntuación obtenida por estudiantes, luego de la intervención. Esto permite

que el rango de puntajes de aciertos haya aumentado y se empezaran a concentrar los puntajes hacia la zona derecha de la gráfica; es decir, con tendencia a aumentar.

Se procedió al análisis de las dimensiones trabajadas en el Test de Pensamiento Computacional, las cuales son: Concepto Computacional, el cuál apunta a la parte conceptual de la lógica con la cual se deben resolver las situaciones, es decir, la estructura que llevan las secuencias propuestas para abordar una situación; Tarea requerida, que hace referencia a la forma de abordar la pregunta para emitir la respuesta, por ello se encuentran preguntas en las cuales se debía establecer una secuencia correcta, se debía completar una secuencia existente o se debía hacer seguimiento a una secuencia para verificar si es correcta; y, la Capacidad evaluada, que se relaciona con la forma en que se debe dar la solución ya sea ideando una solución desde cero, dividiendo un problema en sus partes, verificando los patrones que siguen las soluciones o proponiendo una solución, a determinada situación, expresada en pasos lógicos.

Teniendo como base que el Test aplicado, trabaja 4 conceptos computacionales Direcciones, Bucles, Condicionales y Funciones. Cada uno de ellos tenía asociado una cantidad de preguntas, dentro del test, las cuales se relacionan en la **Tabla 48**. Estas están relacionadas con el concepto computacional, que se muestra a continuación. Cada pregunta aborda un solo concepto computacional, y se encuentran en distintas proporciones puesto que el 10% de las preguntas trabaja el concepto de Direcciones, un 30% el de Bucles, un 50% el de Condicionales y el 10% restante el de Funciones.

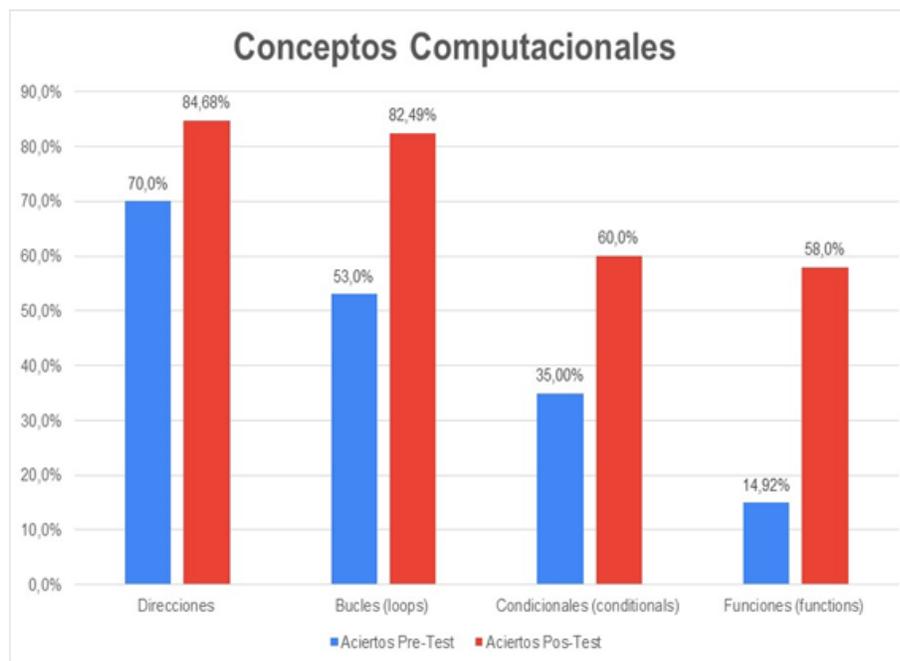
**Tabla 48. Actividad Evaluación CT (Pos-Test)**

| Concepto Computacional       | Pregunta   | Total     |
|------------------------------|--|-----------|
| Direcciones                  | 1, 2, 3, 4   | 4         |
| Bucles (loops)               | 5, 6, 7, 8, 9, 10, 11                                | 7         |
| Condicionales (conditionals) | 12, 13, 14, 15, 16,<br>17, 18, 19, 20, 21, 22, 23,24 | 13        |
| Funciones (functions)        | 25, 26, 27, 28                                       | 4         |
| <b>TOTAL</b>                 |  | <b>28</b> |

**Fuente. Propia de la investigación**

En la **Figura 35**. Comparativo Pre test Post Test – Concepto computacional, se muestra el resumen de los resultados obtenidos.

**Figura 35. Comparativo Pre Test y Post Test - Concepto Computacional**



Fuente. Propia de la investigación.

Se observa que en las subcategorías Direcciones, Bucles, Condicionales y Funciones, se presenta un incremento porcentual de aciertos. Esto permite inferir que los participantes luego de la intervención con el modelo CT4E, han logrado comprender y asimilar los conceptos computacionales trabajados, en términos generales, lograron mayor claridad en el manejo de los conceptos computacionales.

### Tarea Requerida.

El test propuso 3 tareas específicas Secuenciar, Completar y Depurar. En la **Tabla 49**. Se muestran las preguntas relacionadas con la tarea requerida, se relacionan las preguntas que trabaja cada tipo de tarea. Cabe anotar que cada pregunta se relaciona sólo con una de las 3 tareas mencionadas anteriormente, así el 50% de las preguntas se relacionan con la tarea de Secuenciar, un 30% con la de Completar y el 20% restante con la de Depurar.

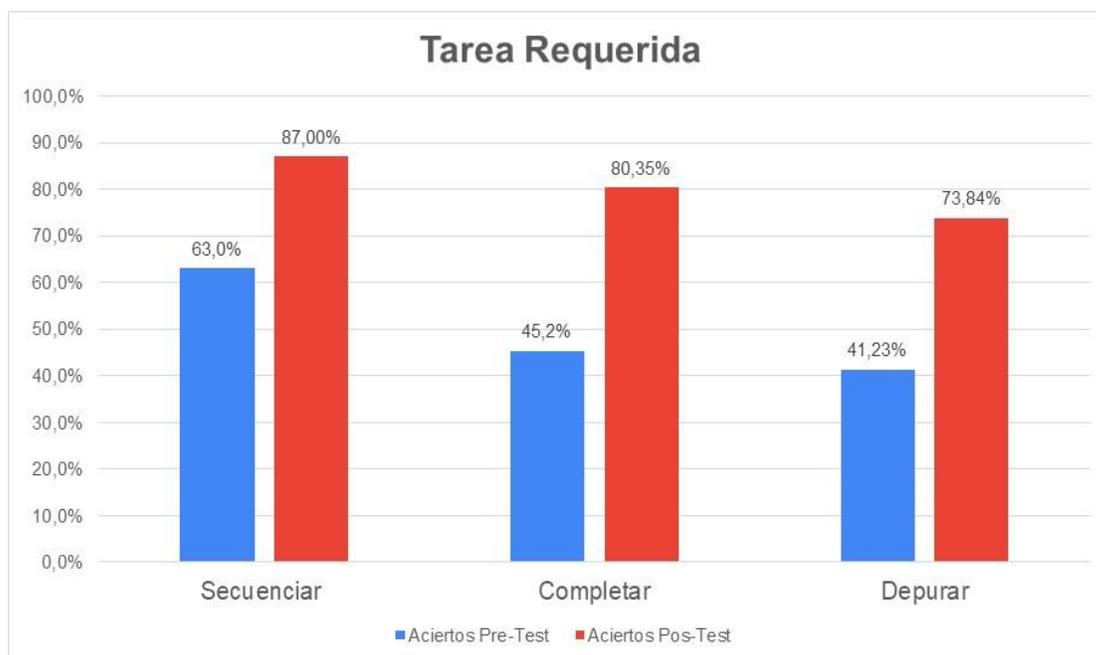
**Tabla 49. Preguntas relacionadas con la tarea requerida**

| Tarea Requerida | Preguntas   | Total |
|-----------------|---|-------|
| Secuenciar      | 1, 4, 5, 8, 9, 12, 13, 14, 17, 18, 21, 22, 25, 27 | 14    |
| Completar       | 2, 6, 10, 15, 19, 23, 24, 26, 28                  | 9     |
| Depurar         | 3, 7, 11, 16, 20                                  | 5     |
| TOTAL           |   | 28    |

Fuente. Propia de la investigación

En la **Figura 36**. Comparativo Pre test Post Test – Tarea requerida, se muestra el resumen de los resultados obtenidos.

**Figura 36. Comparativo Pre Test y Post Test - Tarea Requerida**



**Fuente. Propia de la investigación**

Las categorías denominadas Secuenciar, Completar y Depurar, registran aumento con relación a los aciertos obtenidos en la comparación del Pre-Test y Pos-Test, cerca de 24 puntos porcentuales se incrementa la categoría Secuenciar; por su parte la categoría de Completar registra un significativo aumento de 35 puntos porcentuales, por último, la categoría Depurar presenta un incremento 32 puntos porcentuales, indicando que los estudiantes han incrementado el desarrollo de estos aspectos.

### **Capacidad evaluada.**

Por último, el Test evalúa 4 capacidades relacionadas con el pensamiento computacional Abstracción, Descomposición, Reconocimiento de patrones y Algoritmos. En las preguntas del test, se puede evaluar más de 1 capacidad, por lo tanto, al calcular el porcentaje de preguntas de cada capacidad, la totalización da un porcentaje mayor al 100%. Un 60% de las Preguntas evalúan la capacidad de Abstracción, otro 60% evalúa la Descomposición, un 90% evalúa el Reconocimiento de patrones y otro 90% los Algoritmos.

Las preguntas que evalúan estas capacidades se relacionan en la **Tabla 50**. Preguntas relacionadas con la capacidad evaluada.

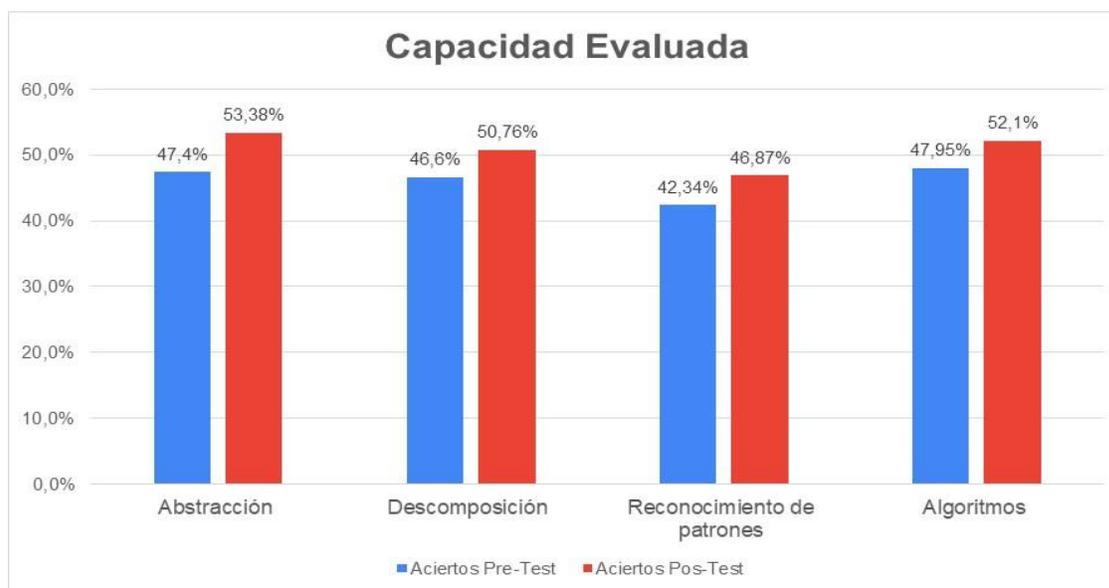
**Tabla 50. Preguntas relacionadas con la capacidad evaluada**

| Capacidad Evaluada         | Preguntas  | Total |
|----------------------------|--|-------|
| Abstracción                | 1, 2, 3, 7, 11, 12, 13, 14, 15, 21, 22, 23, 25, 26, 27, 28                                     | 16    |
| Descomposición             | 4, 5, 6, 7, 10, 11, 12, 13, 15, 21, 22, 23, 25, 26, 27, 28                                     | 16    |
| Reconocimiento de patrones | 4, 5, 6, 8, 9, 10, 11, 12, 15, 17, 18, 20, 22, 23, 25, 26, 27, 28                              | 26    |
| Algoritmos                 | 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 | 26    |
| TOTAL                      |  | 84    |

Fuente. Propia de la investigación

En la **Figura 37**. Comparativo Pre Test y Post Test – Capacidad evaluada, se muestra el resumen de los resultados obtenidos.

**Figura 37. Comparativo Pre Test y Post Test - Capacidad Evaluada**



Fuente. Propia de la investigación

Al comparar el Post Test con el Pre Test, se observa un aumento de por lo menos 5% en cada una de las capacidades evaluadas. Se presenta un incremento razonable, teniendo en cuenta que la integración podría no ser la única causa, está juega un papel importante en ellos. lo cual permite inferir que las actividades propuestas en el modelo y el trabajo con Scratch, permitieron

generar en los estudiantes un desarrollo en las habilidades de pensamiento computacional y la asimilación de los conceptos.

#### **Resultados cualitativos.**

En los resultados cualitativos tenemos las apreciaciones de los investigadores del experimento y las amenazas de validez

#### **Apreciaciones de los investigadores durante el desarrollo del caso de estudio descriptivo.**

Al hacer un contraste entre los resultados que miden la apropiación de los conceptos de pensamiento computacional de los participantes del en el estudio, se observa que presenta un incremento al incluir las competencias STEM el modelo de ChildProgramming. La motivación y la participación fueron alta durante el transcurso del experimento y se logra evidenciar el aumento de la asimilación de los conceptos de pensamiento computacional. En consecuencia, de acuerdo con los resultados obtenidos, es posible validar las hipótesis 1 y 2, que plantea un incremento significativo en la asimilación y apropiación de los conceptos de pensamiento computacional en los estudiantes que participan en el experimento.

Algo similar ocurre en cuanto a la participación. En este aspecto se observa que al integrar las competencias STEM y realizar actividades de trabajo con Scratch, si mejora la participación tanto activa como en grupos de trabajo de los estudiantes, al mismo tiempo que incrementan sus habilidades computacionales. Por lo tanto, se da validez a la hipótesis 1 y 2 planteada en el estudio de caso descriptivo.

#### **Amenazas de validez.**

En el estudio de caso descriptivo una de las amenazas de validez fue el reducido número de muestras disponibles para el estudio. Por esta razón no se aplicaron pruebas estadísticas descriptivas para sustentar las hipótesis desde un punto de vista estadístico. Sin embargo, se hicieron comparaciones muy puntuales en varias variables que permiten verificar los variados efectos que tiene la metodología sobre el desarrollo de habilidades en los diferentes dominios estudiados.

### **4.3 Discusión**

Una parte importante de la investigación fue el estudio de tipo descriptivo conformado por dos fases, una exploratoria y una comparativa. La fase exploratoria ayudo a consolidar el modelo ya que nos permitió ver el fenómeno en el sitio y el enfoque de ingeniería de método nos ayudó a llegar a un modelo coherente. Ver el modelo ya creado es una respuesta a la pregunta de investigación. De ellos se pudo extraer que la integración de las competencias STEM y los

conceptos de CT en la metodología ChildProgramming permitió el desarrollo de las habilidades de CT en los estudiantes de primer semestre de Ingeniería Electrónica y Telecomunicaciones e Ingeniería Automática Industrial. Además, las actividades realizadas fueran ejecutadas con una motivación y participación alta de los estudiantes de ingeniería (Hipótesis 1). En cuanto a la hipótesis 2, formulada en la planeación del estudio, en relación con la apropiación de los conceptos de CT, puede decirse que, al incluir las competencias STEM y los conceptos de CT en el contexto de ChildProgramming se evidenció un incremento en la asimilación de los conceptos del CT. Aunque los estudiantes no definían los conceptos en su totalidad, sí manifestaban su realización práctica por medio de sus acciones y comportamiento; por ejemplo, la descomposición, el reconocimiento de patrones, y el diseño de algoritmos.

En cuanto a la hipótesis 3, incluir las competencias STEM y los conceptos de CT en el contexto de ChildProgramming como herramienta del desarrollo de pensamiento computacional, produce un incremento en la apropiación de conceptos del PC; esta hipótesis es válida teniendo en cuenta los resultados obtenidos en la fase comparativa del estudio de caso descriptivo lo demuestran.

La integración viene dada porque el modelo fue factible y se diseñó en un tiempo razonable.

Se presenta un incremento razonable, teniendo en cuenta que la integración podría no ser la única causa, está juega un papel importante en ellos.

# Capítulo 5

## Conclusiones y trabajos futuros

En este capítulo se presentan, las conclusiones a las que se ha llegado tras la elaboración de este trabajo, así como las recomendaciones, dirigidas a investigadores y personas en general interesadas en el tema, sobre particularidades a tener en cuenta al abordar éste u otro proyecto de características similares; finalmente, se presenta una orientación de lo que, a nuestro criterio, constituyen direcciones hacia las que podrían encaminarse futuros trabajos que pudieran tomar éste en consideración, ya sea como punto de partida, como complemento o como una referencia.

### 5.1 Conclusiones

Hemos obtenido un modelo que integra el desarrollo del pensamiento computacional y los conceptos STEM en el contexto del enfoque metodológico de ChildProgramming, con una prueba piloto que nos da resultados alentadores.

A través del análisis de los datos obtenidos en el estudio de caso descriptivo, se observa que la integración propuesta por el modelo CT4E es viable, y si bien la integración podría no ser la única causa de los resultados de la evaluación, si juega un papel muy importante en ellos.

El modelo CT4E contribuye de manera satisfactoria en el manejo de conceptos y aplicación de conocimientos en situaciones del contexto real, tanto de manera conectada y desconectada lo que promovió la motivación, frente a la solución de problemas planteados en diversos espacios.

El modelo de integración CT4E es una contribución académica a la Facultad de Ingeniería Electrónica y Telecomunicaciones, dado que puede ser usado para fortalecer el curso de introducción a la ingeniería.

Las actividades académicas fundamentadas en este modelo de integración, son unas de las principales contribuciones que se centran en el desarrollo del CT.

El modelo permite a los docentes tener una guía para elaborar actividades dentro de sus asignaturas, que den a los estudiantes pautas para la resolución de problemas. En donde estas incentivan al estudiante a desarrollar su capacidad de abstraer e identificar un problema, priorizar su importancia, determinar las posibles soluciones y prototipar la solución más adecuada por medio de un trabajo colaborativo.

## 5.2 Trabajos futuros

El desarrollo de la investigación: “Integrando el Desarrollo del Pensamiento Computacional y los Conceptos T&E STEM en el Enfoque Metodológico de ChildProgramming”, nos ha permitido alcanzar los objetivos propuestos en su realización, dejando conocer que, en la literatura estudiada, se muestra que la educación STEM y el Pensamiento Computacional poseen una línea de conocimiento activa y en constante desarrollo, que nos permite percibir las dinámicas de sus procesos de aprendizaje, apropiación y ampliación.

Se recomienda para los trabajos futuros en los nuevos diseños manejar varias fuentes de información.

De esta monografía se derivan algunas posibles líneas de trabajos futuros que pueden ser una prolongación de, o que se encuentren relacionados con este; entre ellas, se destacan:

- Plantear la adición del modelo CT4E en el currículo de la carrera de Ingeniería, como contribución para lograr una adaptación curricular que incluya estas nuevas competencias.
- Desarrollar herramientas evaluativas que permitan comparar con mayor facilidad las habilidades de Pensamiento Computacional.
- Proponer la aplicación del modelo CT4E en universidades de diferentes ciudades, con una muestra mayor y variada.
- Adaptar el modelo CT4E para aplicar en los colegios en los grados 10 y 11, para ayudar a los estudiantes a definir su vocación profesional.
- Ampliar la integración a todas las competencias STEM y habilidades de CT en la metodología ChildProgramming y además de extender su uso.

# Bibliografía

- [1] J. Wing, «Computational Thinking», *Commun. ACM*, vol. 49, pp. 33-35, mar. 2006, doi: 10.1145/1118178.1118215.
- [2] M. Hacker, «Integrating Computational Thinking into Technology and Engineering Education», *Technol. Eng. Teach.*, vol. 77, n.º 4, pp. 8-14, 2018.
- [3] J. Wu, Y. Wang, H. Kong, y L. Zhu, «How to Cultivate Computational Thinking-Enabled Engineers: A Case Study on the Robotics Class of Zhejiang University», presentado en 2019 ASEE Annual Conference & Exposition, jun. 2019. Accedido: 22 de abril de 2022. [En línea]. Disponible en: <https://peer.asee.org/how-to-cultivate-computational-thinking-enabled-engineers-a-case-study-on-the-robotics-class-of-zhejiang-university>
- [4] E. S. Ludeña, «La educación STEAM y la cultura “maker”», *Rev. Padres Maest. J. Parents Teach.*, n.º 379, pp. 45-51, 2019.
- [5] J. Hurtado, C. C. Alberto, C. S. Tatiana, y R. O. Eduardo, «Child Programming: Una Estrategia de Aprendizaje y Construcción de Software Basada en la Lúdica, la Colaboración y la Agilidad».
- [6] J. Hurtado, «Toward a Scientific Method in Software Engineering (Position Paper)».
- [7] M. Bunge, *La ciencia, su método y su filosofía*. Sudamericana, 1997.
- [8] M. Shaw, «What makes good research in software engineering?», *Int. J. Softw. Tools Technol. Transf.*, 2002, doi: 10.1007/s10009-002-0083-4.
- [9] P. Runeson, C. Wohlin, M. Hst, M. C. Ohlsson, B. Regnell, y A. Wessln, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.
- [10] J. Wing, «Computational thinking’s influence on research and education for all», *Ital. J. Educ. Technol.*, vol. 25, n.º 2, pp. 7-14, 2017.
- [11] C. Wang, J. Shen, y J. Chao, «Integrating Computational Thinking in STEM Education: A Literature Review», *Int. J. Sci. Math. Educ.*, nov. 2021, doi: 10.1007/s10763-021-10227-5.
- [12] P. Sengupta *et al.*, «Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework», *Educ. Inf. Technol.*, vol. 18, pp. 351-380, jun. 2013, doi: 10.1007/s10639-012-9240-x.
- [13] F. Kalelioglu, Y. Gulbahar, y V. Kukul, «A Framework for Computational Thinking Based on a Systematic Research Review», *Balt. J. Mod. Comput.*, vol. 4, pp. 583-596, may 2016.
- [14] C. Mohtadi, M. Kim, y J. Schlosser, «Why integrate computational thinking into a 21 st century engineering curriculum?», ene. 2013.
- [15] A. M. Chimunja, C. A. Collazos, y J. A. Hurtado, «ChildProgramming-C: como una mejora de la dimensión colaborativa del modelo ChildProgramming», *Entre Cienc. E Ing.*, vol. 11, n.º 22, Art. n.º 22, 2017, doi: 10.31908/19098367.3551.

- [16] R. F. Zuñiga, J. A. H. Alegría, y C. A. C. Ordoñez, «Comprendiendo los procesos de abstracción computacional en los niños: un estudio de caso exploratorio», *T C-Rev. Investig. Tecnol. Cienc.*, vol. 1, n.º 8, pp. 57-66, 2014.
- [17] H. F. O. Muñoz, A. García, J. A. H. Alegría, y C. A. Collazos, «Analizando y aplicando la gamificación en el proceso ChildProgramming», *Rev. Colomb. Comput.*, vol. 14, n.º 2, pp. 7-23, 2013.
- [18] C. A. Manzano Quiñonez y J. C. Moreno Vásquez, «Child Gender : Extendiendo child programming con aspectos de diversidad de género», sep. 2017, Accedido: 20 de mayo de 2022. [En línea]. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1745>
- [19] J. A. Hurtado Alegría, V. Y. Gómez Calvache, y A. C. Zambrano Lasso, «Estudiando el modelo ChildProgramming-G para encontrar elementos que permitan desarrollar un sistema de memoria transactiva», *Campus Virtuales Rev. Científica Iberoam. Tecnol. Educ.*, 2017, doi: 10/8.pdf.
- [20] R. Zuñiga, J. Hurtado, C. Collazos, y H. Fardoun, «ChildProgramming Evolution, A Method to Increase the Computational Thinking Skills in School: 4th Iberoamerican Workshop, HCI-Collab 2018, Popayán, Colombia, April 23–27, 2018, Revised Selected Papers», 2019, pp. 57-69. doi: 10.1007/978-3-030-05270-6\_5.
- [21] I. Mejía, J. A. Hurtado, R. F. Z. Muñoz, y B. G. S. España, «Robótica educativa como herramienta para el desarrollo del pensamiento computacional. Una revisión de la literatura», *Rev. Educ. En Ing.*, vol. 17, n.º 33, Art. n.º 33, ene. 2022, doi: 10.26507/rei.v17n33.1216.
- [22] M. Tenorio Melenje María, M. A. Trujillo, J. A. Hurtado Alegría, y C. Collazos, «Debugging Block-Based Programs», en *Human-Computer Interaction*, Cham, 2019, pp. 98-112. doi: 10.1007/978-3-030-05270-6\_8.
- [23] J. Muñoz, «Block-Based Design: Un Método Para El Diseño de Programas Basados en Bloques», Trabajo de grado, Unicauca, 2022.
- [24] M. J. Page *et al.*, «Declaración PRISMA 2020: una guía actualizada para la publicación de revisiones sistemáticas», *Rev. Esp. Cardiol.*, vol. 74, n.º 9, pp. 790-799, sep. 2021, doi: 10.1016/j.recesp.2021.06.016.
- [25] J. R. Turner, R. Baker, y F. Kellner, «Theoretical literature review: Tracing the life cycle of a theory and its verified and falsified statements», *Hum. Resour. Dev. Rev.*, vol. 17, n.º 1, pp. 34-61, 2018.
- [26] I. Mirbel y J. Ralyte, «Situational method engineering: combining assembly-based and roadmap-driven approaches», *Requir Eng*, vol. 11, pp. 58-78, mar. 2006, doi: 10.1007/s00766-005-0019-0.
- [27] S. Atmatzidou y S. Demetriadis, «How to Support Students' Computational Thinking Skills in Educational Robotics Activities», jul. 2014.
- [28] T. Djambong y V. Freiman, *Task-Based Assessment of Students' Computational Thinking Skills Developed through Visual Programming or Tangible Coding Environments*.

- International Association for the Development of the Information Society, 2016. Accedido: 5 de mayo de 2022. [En línea]. Disponible en: <https://eric.ed.gov/?id=ED571389>
- [29] A. F. Barboza Díaz, C. L. Vergara Bula, E. A. Múskus González, y E. N. De Arce Bula, «Estrategia didáctica mediada por el recurso digital SCRATCH, para el desarrollo del pensamiento computacional, en estudiantes de octavo grado de la Institución Educativa Liceo Sahagún», Trabajo de grado - Maestría, Universidad de Cartagena, 2021. Accedido: 26 de noviembre de 2022. [En línea]. Disponible en: <https://hdl.handle.net/11227/14724>
- [30] C. Selby y J. Woollard, «Computational thinking: the developing definition», 2013. <https://eprints.soton.ac.uk/356481/> (accedido 5 de mayo de 2022).
- [31] E. Segredo, G. Miranda Valladares, y C. Leon, «Towards the Education of the Future: Computational Thinking as a Generative Learning Mechanism - Hacia la educación del futuro: El pensamiento computacional como mecanismo de aprendizaje generativo», *Educ. Knowl. Soc. EKS*, vol. 18, p. 33, jul. 2017, doi: 10.14201/eks2017182335.
- [32] H. İ. Haseski, U. Ilic, y U. Tugtekin, «Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends.», 2018, doi: 10.5539/IES.V11N4P29.
- [33] D. Alimisis, «Educational robotics: Open questions and new challenges», *Themes Sci. Technol. Educ.*, vol. 6, pp. 63-71, ene. 2013.
- [34] K. Brennan y M. Resnick, «New frameworks for studying and assessing the development of computational thinking», en *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, 2012, vol. 1, p. 25.
- [35] «NSF 93-143 Guide to Programs in the Division of Research, Evaluation and Dissemination with specific guidelines for individual programs, September 1993».
- [36] D. Donahoe, «“The Definition of STEM? “», *Today's Eng. IEEE-USA*, dic. 2013.
- [37] T. R. Kelley y J. G. Knowles, «A conceptual framework for integrated STEM education», *Int. J. STEM Educ.*, vol. 3, n.º 1, p. 11, jul. 2016, doi: 10.1186/s40594-016-0046-z.
- [38] W. E. Dugger y S. Fellow, «Evolution of STEM in the United States».
- [39] B. Yildirim, «An Analyses and Meta-Synthesis of Research on STEM Education», *J. Educ. Pract.*, vol. 7, n.º 34, pp. 23-33, 2016.
- [40] T. R. Kelley y J. G. Knowles, «A Conceptual Framework for Integrated STEM Education», *Int. J. STEM Educ.*, vol. 3, 2016, doi: 10.1186/s40594-016-0046-z.
- [41] J. S. Morrison, «Attributes of STEM education: The students, the academy, the classroom», *TIES STEM Educ. Monogr. Ser.*, 2006.
- [42] J. Pleasants, M. P. Clough, J. K. Olson, y G. Miller, «Fundamental Issues Regarding the Nature of Technology», *Sci. Educ.*, vol. 28, n.º 3, pp. 561-597, jul. 2019, doi: 10.1007/s11191-019-00056-y.
- [43] M. Domingo Blázquez, «Desarrollo de competencias STEM mediante Scratch», masters, Ice, 2016. Accedido: 29 de noviembre de 2022. [En línea]. Disponible en: <https://oa.upm.es/43799/>

- [44] W. Dugger y R. Satchwell, «ITEEA - Technology for All Americans: A Rationale and Structure for the Study of Technology (Rationale and Structure)». <https://www.iteea.org/42618.aspx> (accedido 1 de mayo de 2022).
- [45] «Herramienta Scratch - Imagine, Program, Share». <https://scratch.mit.edu/> (accedido 29 de agosto de 2022).
- [46] A. C. Zambrano Lasso y V. Y. Gómez Calvache, «Un sistema de memoria transactiva para ChildProgramming-G», jul. 2017, Accedido: 15 de mayo de 2022. [En línea]. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1736>
- [47] D. Yang, Y. Baek, Y.-H. Ching, S. Swanson, B. Chittoori, y S. Wang, «Infusing Computational Thinking in an Integrated STEM Curriculum: User Reactions and Lessons Learned», *Eur. J. STEM Educ.*, vol. 6, n.º 1, 2021.
- [48] D. D. Yepes Miranda, «Stem y sus oportunidades en el ámbito educativo», jun. 2020, Accedido: 2 de noviembre de 2022. [En línea]. Disponible en: <https://repositorio.unicordoba.edu.co/handle/ucordoba/2774>
- [49] H. Shoaib y S. P. Brophy, «A Literature-based Perspective Towards Learning and Pedagogy of Computational Thinking», presentado en 2020 ASEE Virtual Annual Conference Content Access, jun. 2020. Accedido: 22 de abril de 2022. [En línea]. Disponible en: <https://peer.asee.org/a-literature-based-perspective-towards-learning-and-pedagogy-of-computational-thinking>
- [50] S. Gross, M. Kim, J. Schlosser, D. Lluch, C. Mohtadi, y D. Schneider, «Fostering computational thinking in engineering education: Challenges, examples, and best practices», presentado en IEEE Global Engineering Education Conference, EDUCON, abr. 2014, pp. 450-459. doi: 10.1109/EDUCON.2014.6826132.
- [51] L. D. Miller, L.-K. Soh, V. Chiriacescu, E. Ingraham, D. Shell, y M. P. Hazley, «Integrating computational and creative thinking to improve learning and performance in CS1», *SIGCSE*, 2014, doi: 10.1145/2538862.2538940.
- [52] C. E. Vergara *et al.*, «Aligning Computing Education with engineering workforce computational needs: New curricular directions to improve computational thinking in engineering graduates», en *2009 39th IEEE Frontiers in Education Conference*, oct. 2009, pp. 1-6. doi: 10.1109/FIE.2009.5350463.
- [53] E. F. Crawley, A. «Peko» Hosoi, y A. «Babi» Mitra, «Redesigning Undergraduate Engineering Education at MIT – the New Engineering Education Transformation (NEET) initiative», presentado en 2018 ASEE Annual Conference & Exposition, jun. 2018.
- [54] S. Atmatzidou y S. Demetriadis, «Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences», *Robot. Auton. Syst.*, vol. 75, pp. 661-670, ene. 2016, doi: 10.1016/j.robot.2015.10.008.
- [55] S. Atmatzidou y S. Demetriadis, «A Didactical Model for Educational Robotics Activities: A Study on Improving Skills Through Strong or Minimal Guidance», en *Educational Robotics in the Makers Era*, Cham, 2017, pp. 58-72. doi: 10.1007/978-3-319-55553-9\_5.

- [56] V. Dolgopolas, T. Jevsikova, L. Savulionienė, y V. Dagiene, «On Evaluation of Computational Thinking of Software Engineering Novice Students», jul. 2015. doi: 10.13140/RG.2.1.2855.9206.
- [57] G. Yakman y H. Lee, «Exploring the Exemplary STEAM Education in the U.S. as a Practical Educational Framework for Korea», *J. Korean Assoc. Res. Sci. Educ.*, vol. 32, ago. 2012, doi: 10.14697/jkase.2012.32.6.1072.
- [58] S. T. Cruz, O. E. Rojas, y J. Alegría, «Un Modelo Para la Enseñanza de la Programación de Software en niños a través de Estrategias Colaborativas», *Univ. Cauca*, 2013.
- [59] C. A. Manzano Quiñonez y J. C. Moreno Vásquez, «Child Gender: Extendiendo Child Programming con aspectos de diversidad de género», sep. 2017, Accedido: 24 de diciembre de 2022. [En línea]. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1745>
- [60] R. Martínez, *El secreto detrás de una tesis: Cómo logré convertir mi experiencia al hacer una tesis en la solución que todo profesional necesita*. Biblioteca Nacional del Perú, 2020.
- [61] R. Strods, L. Daniela, y I. France, «Activities with Educational Robotics: Research Model and Tools for Evaluation of Progress», en *Smart Learning with Educational Robotics: Using Robots to Scaffold Learning Outcomes*, L. Daniela, Ed. Cham: Springer International Publishing, 2019, pp. 251-266. doi: 10.1007/978-3-030-19913-5\_10.
- [62] R. K. Yin, *Case Study Research: Design and Methods*. SAGE, 2009.
- [63] S. Chetty, «The Case Study Method for Research in Small-and Medium-Sized Firms», *Int. Small Bus. J.*, vol. 15, n.º 1, pp. 73-85, oct. 1996, doi: 10.1177/0266242696151005.
- [64] J. Moreno-León, G. Robles, y M. Román-González, «Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking», *RED-Rev. Educ. Distancia*, sep. 2015.
- [65] M. Román-González, J.-C. Pérez-González, y C. Jiménez-Fernández, «Test de Pensamiento Computacional: diseño y psicometría general [Computational Thinking Test: design & general psychometry]», oct. 2015. doi: 10.13140/RG.2.1.3056.5521.