

MODELO DE EVALUACIÓN BASADA EVIDENCIA PARA ESTIMAR EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL



Universidad
del Cauca

LAURA MARÍA OROZCO GARCÍA

Tesis de Maestría en Computación

Directora:

Carolina González Serrano

PhD. en Ciencias de la Información y las Comunicaciones

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo de Investigación en Inteligencia Computacional - GICO

Áreas de Investigación: Inteligencia Computacional

Popayán, Febrero 2020

LAURA MARÍA OROZCO GARCÍA

MODELO DE EVALUACIÓN BASADA EN EVIDENCIA
PARA ESTIMAR EL DESARROLLO DEL PENSAMIENTO
COMPUTACIONAL

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

Magíster en:
Computación

Directora:
Carolina González Serrano
PhD. en Ciencias de la Información y las Comunicaciones

Popayán
2020

*A la sagrada molécula que me permitió sanar
y aprender a vivir*

A Otac

A Kambo

Resumen estructurado

Debido a los cambios presentados en el trabajo y la sociedad, el reto de recibir información a través de un proceso de aprendizaje tradicional de transmisión de conocimientos ya no es suficiente. Se hace necesario que los estudiantes, alcancen diferentes competencias que les ayuden a desarrollar capacidades de análisis, síntesis y aplicación de lo aprendido para afrontar procesos de resolución de problemas, diseñar soluciones pertinentes y saber comunicarse y coordinar con los demás. Otro de los retos en el ámbito educativo, corresponde a la evaluación como un proceso formativo que permita evaluar habilidades cognitivas y competencias de alto nivel como: creatividad e innovación, pensamiento computacional, resolución de problemas, trabajo en equipo, toma de decisiones, entre otras.

Una de las habilidades que facilita desarrollar capacidades de síntesis y análisis es el pensamiento computacional, esta es una habilidad fundamental que debe ser adquirida por las personas para desempeñarse correctamente en la sociedad del conocimiento, y cuyo objetivo se centra en la resolución de problemas, diseño de soluciones y comprensión del comportamiento humano partiendo de los conceptos fundamentales de la informática.

El propósito principal del siguiente trabajo es diseñar un modelo de evaluación que empleando el marco de referencia basado en evidencia, soporte la incorporación de solución de problemas a través prácticas de programación, facilitando el análisis por medio de una aplicación automática que permite estimar el desarrollo del Pensamiento Computacional en estudiantes de Educación Básica. Para lograr este propósito, se han planteado otros objetivos que soportan el principal como: realizar una caracterización de criterios mínimos para conocer que se debe evaluar y de los mecanismos para conocer cómo se debe evaluar, definir un modelo de clasificación para determinar la correspondencia entre la evidencia entregada por el estudiante y los criterios definidos por el docente y finalmente, desarrollar un prototipo de herramienta de evaluación y

probarlo mediante un estudio de caso con estudiantes de educación básica de Popayán.

Para lograr los objetivos planteados se ha definido una metodología que incluye como método para la caracterización de criterios y mecanismos el modelo de estudiante de ingeniería, que se realiza definiendo núcleos temáticos para el análisis de información, la clasificación se desarrolla siguiendo el marco de referencia para minería de datos CRISP-DM, el prototipo fue desarrollado siguiendo metodologías de desarrollo ágil, y finalmente el diseño del estudio de caso se realizó teniendo en cuenta la guía para el diseño de estudios de caso en ingeniería de software.

Los resultados muestran que el modelo propuesto contribuye a estimar el desarrollo del pensamiento computacional en los estudiantes, ya que las fases que se proponen guían al docente en este proceso, brindando información útil y necesaria como los criterios mínimos, que facilitan al docente conocer más acerca de la habilidad y de esta manera ellos pueden incorporarla en sus sesiones en el aula.

Es valioso que se desarrolle el pensamiento computacional desde edades tempranas, para que esto suceda se necesita que los docentes conozcan los factores que deben evaluar y la manera de hacerlo, y cuenten con estrategias y herramientas que faciliten el proceso; lo que permite la adecuada incorporación de la habilidad en la enseñanza y de esta manera podrá ser incluido transversalmente en las sesiones de clase de las diferentes áreas del conocimiento.

Structured abstract

Due to the changes presented in work and society, the challenge of receiving information through a traditional learning process of knowledge transmission is no longer enough. Students must achieve different competencies that help them to develop analytically, synthesis and application skills to face problem-solving processes, design pertinent solutions and know how to communicate and coordinate with others. Another challenge in the educational field corresponds to the evaluation as a training process that allows assessing cognitive skills and high level skills such as creativity and innovation, computational thinking, problem-solving, teamwork, decision making, among others.

One of the abilities that facilitate developing skills of synthesis and analysis is computational thinking, this is a fundamental ability that must be acquired by people to perform correctly in the knowledge society, and whose objective is focused on problem solving, design of solutions and understanding of human behavior based on the fundamental concepts of computer science.

The principle goal of the following work is to design an evaluation model that uses the reference framework ECD, supports the incorporation of problem-solving through programming practices, facilitating the analysis through an automatic application that allows estimating the development of the Computational Thinking in Basic Education students. To achieve this purpose, other objectives that support the main one have been proposed, such as: performing a characterization of minimum criteria to know what should be evaluated and mechanisms to know how to evaluate them, define a classification model to determine the correspondence between the evidence delivered by the student and the criteria defined by the teacher; and finally, develop a prototype evaluation tool and test it through a case study with elementary school students in Popayan.

To achieve the proposed objectives, a methodology has been defined that includes as a method for the characterization of criteria and mechanisms the engineering student model, which is carried out by defining thematic cores for the analysis of information, the classification is developed following the reference framework for data mining CRISP-DM, the prototype was developed following agile development methodologies, and finally, the design of the case study was made taking into account the guidelines for the design of case studies in software engineering.

The results show that the proposed model supports to estimate the development of computational thinking in students since they propose a series of steps that guide the teacher in this process, providing useful and necessary information such as minimum criteria, which make it easier for the teacher to know the skill and in this way they can incorporate it into their classroom sessions

It is important to develop computational thinking from an early age, it is necessary that teachers know the factors that must evaluate and how to do it, and have strategies and tools that facilitate the process; what allows the adequate incorporation of the teaching ability and in this way can be included transversally in their class sessions of the different areas of knowledge.

Tabla de contenido

Capítulo 1	23
1. Introducción.....	23
1.1. Planteamiento del problema.....	23
1.2. Justificación.....	25
1.3. Aporte investigativo	26
1.4. Objetivos	27
1.4.1. Objetivo general.....	27
1.4.2. Objetivos específicos.....	28
1.5. Hipótesis	28
1.6. Estructura del documento	28
Capítulo 2	31
2. Marco teórico y estado del arte	31
2.1. Pensamiento Computacional	31
2.2. Evaluación educativa	33
2.2.1. Evaluación formativa	33
2.2.2. Evaluación Basada en Evidencia.....	35
2.3. Programación en el Contexto Escolar	37
2.3.1. Herramientas empleadas en prácticas de programación en el contexto escolar	37
2.3.2. Experiencias de programación en el contexto escolar.....	39
2.4. Estado del Arte.....	43
2.4.1. Marcos de Referencia y Evaluación del Pensamiento Computacional ..	43
Marco de Referencia de CT y Evaluación del aprendizaje a través del diseño ...	43
Marco de referencia para el análisis de Patrones del Pensamiento Computacional	45

Marco de referencia para el diseño y evaluación de códigos de programación visual	46
2.4.2. Herramientas y otras aproximaciones para la evaluación de CT	47
2.5. Conclusiones del estado del arte	51
Capítulo 3	53
3. Caracterización de los criterios mínimos para el desarrollo del pensamiento computacional y los mecanismos para su evaluación	53
3.1. Metodología.....	53
3.2. Fase preparatoria	55
3.2.1. Aspectos formales	55
3.2.2. Asunto investigado.....	55
3.2.3. Delimitación contextual	55
3.2.4. Propósito.....	55
3.2.5. Enfoque	56
3.2.6. Metodología	56
3.3. Fase Descriptiva.....	58
3.3.1. Elementos del Pensamiento Computacional	58
3.3.2. Criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación	67
3.4. Fase de Interpretación por Núcleos Temáticos	82
3.4.1. Mapeo de términos de los elementos de CT	83
3.4.2. Mapeo de los criterios mínimos para estimar el desarrollo de CT	88
3.4.3. Mapeo de los mecanismos de evaluación	95
3.5. Fase de construcción teórica global	99
3.5.1. Elementos de CT	100
3.5.2. Criterios mínimos para la evaluación de CT y sus mecanismos de evaluación	105
3.6. Fase de extensión y publicación	108
3.7. Conclusiones.....	108
Capítulo 4	111
4. Modelo de evaluación basada en evidencia	111
4.1. Alcance.....	112
4.2. Proceso de evaluación	113
4.3. Construcción del modelo	113

4.3.1.	Diseño de evaluaciones basadas en evidencia	114
4.3.2.	Modelo de evaluación para estimar el desarrollo de CT	115
4.3.3.	Fases del modelo	117
4.4.	Clasificación	127
4.4.1.	Metodología	127
4.4.2.	Contexto de minería de datos.....	128
4.4.3.	Comprensión del negocio	129
4.4.4.	Comprensión de la información	130
4.4.5.	Preparación de la información	135
4.4.6.	Modelado.....	139
4.4.7.	Evaluación	144
4.4.8.	Despliegue.....	146
4.5.	Conclusiones.....	147
Capítulo 5.....		149
5.	HERA-CT: Herramienta de Evaluación y Realimentación para el Aprendizaje – CT 149	
5.1.	Análisis del prototipo	149
5.1.1.	Definición del prototipo	149
5.1.2.	Identificación de requisitos	150
5.1.3.	Casos de uso.....	156
5.2.	Diseño del prototipo	167
5.2.1.	Visión general de la solución	167
5.2.2.	Procesos principales de la solución.....	168
5.2.3.	Arquitectura de la solución	169
5.3.	Implementación del prototipo	172
5.3.1.	Interfaz gráfica de usuario	173
5.4.	Despliegue del prototipo.....	179
Capítulo 6.....		181
6.	Estudio de caso.....	181
6.1.	Diseño y planeación del estudio de caso	181
6.1.1.	Definición del caso.....	181
6.1.2.	Protocolo del estudio de caso.....	184
6.1.3.	Consideraciones éticas.....	185

6.2. Recolección de datos	186
6.2.1. Fuentes de datos	187
6.3. Análisis de los datos recolectados	188
6.4. Conclusiones.....	193
Capítulo 7	195
7. Conclusiones y trabajo futuro	195
7.1. Conclusiones.....	195
7.2. Trabajos futuros	196
Bibliografía.....	199
Anexos	209
Anexo A.....	211
A. Tablas de apoyo en el proceso de caracterización – capítulo 3.....	211
Anexo B.....	221
B. Tablas de apoyo de la definición del modelo– capítulo 4	221
Anexo C.....	233
C. Tablas de apoyo en la definición de la herramienta HERA-CT capítulo 5.....	233
Anexo D.....	235
D. Información de soporte del estudio de caso- capítulo 6.....	235

Lista de tablas

Tabla 1. Caracterización de elementos de CT considerados por autores.....	58
Tabla 2. Caracterización de elementos de CT considerados por asociaciones educativas y universidades	60
Tabla 3. Caracterización de elementos de CT considerados por compañías de la industria informática	62
Tabla 4. Caracterización de elementos de CT considerados por autores que describen Marcos de Referencia	63
Tabla 5. Caracterización de elementos de CT considerados por autores que desarrollaron herramientas de evaluación CT.....	63
Tabla 6. Caracterización de elementos de CT considerados por autores que realizaron implementaciones en el aula de clase.....	65
Tabla 7. Referencia 1. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	68
Tabla 8. Referencia 2. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	69
Tabla 9. Referencia 3. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	69
Tabla 10. Referencia 4. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	70
Tabla 11. Referencia 5. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	71
Tabla 12. Referencia 6. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	72
Tabla 13. Referencia 7. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	73
Tabla 14. Referencia 8. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	73
Tabla 15. Referencia 9. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	74

Tabla 16. Referencia 10. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	75
Tabla 17. Referencia 11. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	76
Tabla 18. Referencia 12. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	77
Tabla 19. Referencia 13. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	77
Tabla 20. Referencia 14. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	78
Tabla 21. Referencia 15. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	79
Tabla 22. Referencia 16. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	80
Tabla 23. Referencia 17. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	80
Tabla 24. Referencia 18. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación	81
Tabla 25. Mapeo de terminos a nivel de autores	84
Tabla 26. Mapeo de terminos a nivel de asociaciones educativas.....	84
Tabla 27. Mapeo de terminos a nivel de compañías de la industria informática	85
Tabla 28. Mapeo de terminos a nivel de marcos de referencia.....	86
Tabla 29. Mapeo de terminos a nivel de herramientas de evaluación	87
Tabla 30. Mapeo de terminos a nivel de implementaciones en el aula de clase.....	88
Tabla 31. Mapeo de criterios mínimos encontrados en Marcos de referencia	88
Tabla 32. Mapeo de criterios mínimos encontrados en herramientas.....	90
Tabla 33. Mapeo de criterios mínimos encontrados en implementaciones.....	92
Tabla 34. Mapeo de mecanismos encontrados en Marcos de referencia	96
Tabla 35. Mapeo de mecanismos encontrados en herramientas.....	97
Tabla 36. Mapeo de mecanismos encontrados en implementaciones.....	98
Tabla 37. Elementos de CT.....	100
Tabla 38. Definiciones de elementos de CT.....	101
Tabla 39. Elementos de CT, criterios mínimos y mecanismos de evaluación.....	105
Tabla 40. Mapeo de criterios mínimos y evidencia entregada por Scratch	120
Tabla 41. Encuesta para los estudiantes	121

Tabla 42. Variables seleccionadas tras el proceso de exploración de datos	132
Tabla 43. Variables seleccionadas tras el proceso de verificación de la calidad de los datos	134
Tabla 44. Variables de la plantilla de la data set.....	137
Tabla 45. Plantilla de identificación de requisitos funcionales y no funcionales.....	151
Tabla 46. Requisito funcional RQF01	151
Tabla 47. Requisito funcional RQF02	152
Tabla 48. Requisito funcional RQF03	152
Tabla 49. Requisito funcional RQF04	152
Tabla 50. Requisito funcional RQF05	153
Tabla 51. Requisito funcional RQF06	153
Tabla 52. Requisito funcional RQF07	153
Tabla 53. Requisito funcional RQF08	154
Tabla 54. Requisito funcional RQF09	154
Tabla 55. Requisito no funcional RQNF01	155
Tabla 56. Requisito no funcional RQNF02.....	155
Tabla 57. Requisito no funcional RQNF03.....	155
Tabla 58. Plantilla de descripción textual de los casos de uso	156
Tabla 59. Caso de uso CUE0.1	158
Tabla 60. Caso de uso CUE0.2	158
Tabla 61. Caso de uso CUE1.1	159
Tabla 62. Caso de uso CUE1.2	159
Tabla 63. Caso de uso CUE2.1	160
Tabla 64. Caso de uso CUE2.2	160
Tabla 65. Caso de uso CUE2.3	161
Tabla 66. Caso de uso CUE2.4	161
Tabla 67. Caso de uso CUE3.1	162
Tabla 68. Caso de uso CUE3.2	162
Tabla 69. Caso de uso CUE3.3	163
Tabla 70. Caso de uso CUE4.1	164
Tabla 71. Caso de uso CUE4.2	164
Tabla 72. Caso de uso CUE4.3	164

Tabla 73. Caso de uso CUE5.1.....	165
Tabla 74. Caso de uso CUE5.2.....	165
Tabla 75. Caso de uso CUE6.1.....	166
Tabla 76. Casos y unidades de análisis.....	182
Tabla 77. Métodos de recolección de datos.....	183
Tabla 78. Medidas estadísticas aplicadas.....	188
Tabla 79. Clasificación de estudiantes.....	192

Lista de figuras

Figura 1. Resumen del alcance del proyecto.....	112
Figura 2. Proceso de evaluación. Basado en [34].	113
Figura 3. Implementación ECD.....	115
Figura 4. Fases del modelo de evaluación	116
Figura 5. Diagrama detallado: Modelo de evaluación.....	116
Figura 6. Diagrama general: Fases del modelo de evaluación.....	117
Figura 7. Fase de iniciación.....	118
Figura 8. Fase de planeación	119
Figura 9. Fase de ejecución	123
Figura 10. Módulo del estudiante	125
Figura 11. Módulo del estudiante en el tiempo	126
Figura 12. Diagrama de casos de uso para el escenario 0	157
Figura 13. Diagrama de casos de uso para el escenario 1	159
Figura 14. Diagrama de casos de uso para el escenario 2	160
Figura 15. Diagrama de casos de uso para el escenario 3	162
Figura 16. Diagrama de casos de uso para el escenario 4	164
Figura 17. Diagrama de casos de uso para el escenario 5	165
Figura 18. Diagrama de casos de uso para el escenario 6	166
Figura 19. Procesos principales de HERA-CT.....	169
Figura 20. Esquema Cliente/Servidor.....	169
Figura 21. Diagrama de componentes del sistema	170
Figura 22. Modelo entidad relación de la base de datos	171
Figura 23. GUI información curso.....	173
Figura 24. GUI creación de desafío.....	174

Figura 25. GUI calificación individual.....	175
Figura 26. GUI calificación grupal.....	176
Figura 27. GUI consulta de desafíos	177
Figura 28. GUI inicio de desafío	177
Figura 29. GUI opciones ir al escenario o enviar el proyecto a calificar	178
Figura 30. GUI visualización de resultados.....	178
Figura 31. Vista de despliegue	179
Figura 32. Resultados de desempeño desafío 2	190
Figura 33. Resultados de desempeño desafío 6	191
Figura 34. Resultados de desempeño desafío 13	191

Lista de ecuaciones

Ecuación 1. Coeficiente de correlación lineal.....	132
Ecuación 2. Error cuadrático medio	137
Ecuación 3. Característica scaling	137
Ecuación 4. Teorema de Bayes	146

Capítulo 1

1. Introducción

En el actual capítulo se dará a conocer la naturaleza del problema que generó la investigación realizada, se inicia con el planteamiento del problema que da lugar a la pregunta de investigación, seguido de su justificación, aportes, objetivos e hipótesis y finaliza con la estructura de la solución propuesta.

1.1. Planteamiento del problema

El Pensamiento Computacional (CT: Computational Thinking) ha recibido considerable atención en los últimos años, ya que su desarrollo implica mucho más que aprender a codificar. Permite a los estudiantes realizar procesos de abstracción mediante la definición de patrones y generalización, a partir de casos concretos. Introduce formas de tratamiento de la información y su representación de diferentes maneras. Permite a los estudiantes trabajar sistemáticamente para identificar y eliminar errores, proporcionándoles una forma de enfrentar problemas y desarrollar soluciones vinculando diferentes áreas de conocimiento [1], [2].

Según Jeannette Wing, “el Pensamiento Computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano haciendo uso de los conceptos fundamentales de la informática”; siendo considerada una habilidad esencial que debe estar incluida en la formación de los estudiantes, ya que representa un elemento vital del aprendizaje de la ciencia, la tecnología, la ingeniería y las

matemáticas [1]. Además, el CT por sus características y dimensiones facilita el aprendizaje entre y a través de diferentes disciplinas, en las que puede estar incluido transversalmente para fomentar en los estudiantes diferentes habilidades que les permitan resolver problemas presentes y futuros [3]. El enfoque al que está dirigido el Pensamiento Computacional va más allá del conocimiento introductorio de la computación, se trata de una manera diferente de pensar que es esencial para la sociedad del conocimiento actual [4] [5].

La Sociedad Internacional para la Tecnología en Educación (ISTE), la Asociación de Docentes en Ciencias de la Computación (CSTA), líderes de educación superior, de educación escolar y de la industria desarrollaron una definición operativa del CT en la que incluyen características como: formular problemas, organizar datos, realizar modelos para representar datos, apoyadas en actitudes que son dimensiones esenciales del CT como: confianza en el manejo de la complejidad, persistencia al trabajar con problemas difíciles, habilidades de comunicación y trabajo en equipo, entre otras [6].

No obstante, no existe un acuerdo formal sobre las dimensiones que abarca el CT ni sobre las estrategias de su evaluación [7]. Adicionalmente, se presenta un déficit de herramientas que soporten al profesorado en la evaluación de los proyectos de los estudiantes para identificar el desarrollo del Pensamiento Computacional [8], [9].

Existen aproximaciones cuyo objetivo es fortalecer el Pensamiento Computacional [10], [11], orientadas a desarrollar ejercicios basados en prácticas computacionales, sin incorporar mecanismos explícitos de evaluación, imposibilitando evidenciar con claridad el fortalecimiento del CT. Algunos estudios se enfocan en procesos de evaluación del CT mediante pruebas de selección múltiple, orales o escritas [10], [12], utilizando preguntas explícitas para evaluar el desarrollo de la habilidad. Otros presentan procesos de evaluación semiautomática [13], [14], [15], cuyo objetivo es medir el desempeño de los estudiantes utilizando ejercicios de clase basados en prácticas de programación, enfocadas únicamente en tareas de medición, sin permitir al docente hacer un seguimiento continuo del desempeño de sus estudiantes, además de estar limitadas a versiones de software específicas [13]. Koh et Al. en [16] presentan un sistema de evaluación del CT en tiempo real que permite al docente llevar un seguimiento de los avances a nivel individual y grupal, sin embargo se presenta como

1. Introducción

una versión comercial no disponible para ser usada en un contexto académico de manera gratuita.

Considerando lo anteriormente expuesto, se evidencia que a pesar de la importancia que tiene para la sociedad actual fomentar el desarrollo del CT [17], tanto para estudiantes vinculados al campo de las Ciencias de la Computación como para quienes no lo están, aún se carece de mecanismos efectivos que permitan identificar el conjunto de factores a evaluar, así como su estimación y valoración; que sirvan de soporte y faciliten el proceso de evaluación por parte del profesor, que incorporen el desarrollo de prácticas computacionales mediante el uso de problemas contextualizados [18], de manera que la evaluación pueda convertirse en un estímulo para animar a los estudiantes a mejorar su desempeño y en una herramienta que permita al profesor evidenciar el verdadero progreso de los estudiantes en el fortalecimiento del CT.

Por lo anterior, a través del desarrollo de la presente propuesta se pretende dar respuesta a la pregunta de investigación que se plantea a continuación.

¿Cómo soportar el proceso de evaluación de criterios mínimos para el desarrollo del Pensamiento Computacional en contextos de Educación Básica?

1.2. Justificación

Según la UNESCO en [19], muchos países consideran importante apoyar a los estudiantes para alcanzar su máximo potencial de aprendizaje. Esto incluye el aprendizaje no solo de conocimiento basado en datos, sino también, el fortalecimiento de habilidades de alto nivel necesarias para identificar, analizar y comprender problemáticas del contexto real, plantear soluciones prácticas, efectivas y pertinentes que contribuyan a superar los desafíos de la sociedad actual.

El Ministerio de Educación Nacional (MEN) apoya las prácticas relacionadas con el fortalecimiento de habilidades y competencias, aunque no especifica la manera como deban realizarse estas prácticas, ni su integración en el plan académico o en los procesos de evaluación [20]. En el departamento del Cauca se llevan a cabo iniciativas

apoyadas por el Programa Cauca Vive Digital, que incluye contenidos educativos, capacitaciones básicas para docentes e infraestructura tecnológica, sin embargo no contemplan un plan que permita el fortalecimiento y evaluación del Pensamiento Computacional aprovechando los recursos disponibles en la actualidad [21].

Al plantear la propuesta de evaluación basada en evidencia para estimar el desarrollo de CT– Fase I se busca: Contribuir al fortalecimiento del CT en los estudiantes de Educación Básica que asisten al aula Smart School del Parque Informático Carlos Albán, e incentivar a los docentes a usar la programación como instrumento para la solución de problemas. Adicionalmente, en una segunda fase del proyecto se pretende extender el uso de la solución propuesta a una mayor población de estudiantes y profesores, con el fin de expandir el proyecto y proveer evidencia del impacto obtenido.

1.3. Aporte investigativo

Considerando lo expuesto anteriormente, se puede decir que en la actualidad dentro del aula de clase los docentes se enfrentan al reto de generar evaluaciones que consideren la adquisición del CT por parte de los estudiantes. Se puede notar una carencia de herramientas que permitan al docente hacer una evaluación formativa en un entorno de libre acceso para todos los estudiantes.

Con este proyecto se busca aportar a la comunidad científica y educativa un modelo de evaluación que estime el desarrollo del CT. Considerando los insumos entregados por el estudiante como la evidencia para valorar si ha adquirido las habilidades mínimas de CT. El docente al conocer los resultados de la evaluación grupal y las individuales puede tomar decisiones y realizar acciones para que los estudiantes fortalezcan la competencia. Para la elaboración de dicho modelo se ha considerado realizar:

- La identificación de los criterios mínimos del CT y sus mecanismos de evaluación.
- La definición de reglas de clasificación que determine la adquisición de la habilidad mediante la evidencia brindada por el estudiante.
- El diseño o adaptación de una herramienta para evaluar el aprendizaje de CT.

1. Introducción

- La validación del modelo propuesto mediante un estudio de caso.

Dentro del proyecto, para la evaluación del modelo, se ha considerado elaborar un repositorio de desafíos en algunas áreas del conocimiento. El objetivo es que los estudiantes resuelvan los desafíos; estos desafíos componen las actividades que se realizan durante las sesiones, razón por la cual cada desafío debe ser planteado considerando el grado de escolaridad en el que se encuentran los estudiantes. Para facilitar el uso del modelo, la herramienta permite al docente incluir sus propios desafíos y compartirlos con los demás docentes.

Como impacto se espera:

- Generar evidencia del aprendizaje mínimo de CT por medio de la resolución de los desafíos.
- Contribuir al fortalecimiento de CT en los estudiantes para que puedan brindar soluciones a problemas de diversas áreas a través de la programación.
- Motivar a los estudiantes a estudiar carreras científico tecnológicas y a los docentes a emplear la programación para fomentar en sus estudiantes habilidades de alto nivel como el CT.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar un modelo de evaluación basado en evidencia¹, que incorporando resolución de problemas mediante prácticas de programación y análisis automático permita

¹ "Evidence-centered Design (ECD) is a methodology for designing assessments that underscores the central role of evidentiary reasoning in assessment design. It is based on three premises: (i) An assessment must build around the important knowledge in the domain of interest and an understanding of how that knowledge is acquired and put to use; (ii) The chain of reasoning from what participants say and do in assessments to inferences about what they know, can do, or should do next, must be based on the principles of evidentiary reasoning; (iii) Purpose must be the driving force behind design decisions, which reflect constraints, resources, and conditions of use". [31]

"Diseño basado en evidencia (ECD) es una metodología para el diseño de las evaluaciones que pone de relieve el papel central del razonamiento probatorio en el diseño de la evaluación. Se basa en tres premisas : (i) Una evaluación debe construir todo el conocimiento importante en el dominio de interés y una comprensión de cómo se adquiere ese conocimiento y se pone en uso; (ii) La cadena de razonamiento de lo que dicen y hacen los participantes en las evaluaciones a inferir acerca de lo que saben, puede hacer, o debe hacer a continuación, debe estar basado en los principios del razonamiento probatorio; (iii) El propósito

estimar el desarrollo del Pensamiento Computacional en estudiantes de Educación Básica.

1.4.2. Objetivos específicos

- Caracterizar los criterios mínimos para el desarrollo del Pensamiento Computacional y los mecanismos para su evaluación.
- Definir un conjunto de reglas de clasificación que permitan determinar la correspondencia entre la evidencia entregada por el estudiante y los parámetros de evaluación definidos por el docente.
- Desarrollar o adaptar un prototipo de herramienta de evaluación del desarrollo del Pensamiento Computacional.
- Evaluar el prototipo mediante un estudio de caso al interior del aula Smart School del Parque Informático de Ciencia, Arte y Tecnología Carlos Albán de la ciudad de Popayán.

1.5.Hipótesis

Mediante la definición de un modelo de evaluación basado en evidencia es posible estimar el desarrollo de CT de los estudiantes de educación básica, a través de la valoración de los criterios mínimos y potenciando la incorporación de problemas, de diferentes áreas del conocimiento, en el desarrollo de la competencia mediante prácticas de programación.

1.6.Estructura del documento

En esta sección se describe la organización del documento que presenta el modelo de evaluación basada en evidencia para estimar el desarrollo del pensamiento computacional. El Capítulo 2 trata el marco teórico y estado de arte, en el que se

debe ser la fuerza impulsora detrás de las decisiones de diseño , que reflejan las limitaciones , los recursos y las condiciones de uso” . (Traducido de: [31]).

1. Introducción

contextualizan las temáticas que fundamentan la investigación y ejecución del proyecto, como lo son: el pensamiento computacional, la evaluación en entornos escolares, la evaluación basada en evidencia, los marcos de referencia, aproximaciones y herramientas para la evaluación del pensamiento computacional. El Capítulo 3 ofrece la caracterización de los criterios mínimos para estimar el desarrollo de CT y los mecanismos que han sido empleados para su evaluación. En el Capítulo 4 se describe el modelo de evaluación que es el objeto de la investigación, para lo cual se describen sus componentes y se explican las reglas de clasificación, que facilitan la automatización de la evaluación. En el Capítulo 5 se brinda la descripción del prototipo, se evidencia el proceso de diseño e implementación ejecutados para la construcción del prototipo. En el Capítulo 6 se describe el estudio de caso realizado para la evaluación del modelo planteado y sus resultados. Y finalmente en el Capítulo 7 se exponen las conclusiones obtenidas a raíz del trabajo realizado, se presentan las lecciones aprendidas y se proponen recomendaciones a considerar para trabajos futuros.

Capítulo 2

2. Marco teórico y estado del arte

2.1. Pensamiento Computacional

El término Pensamiento Computacional fue popularizado por Jeannette M. Wing en el 2006 [1], desde entonces se ha considerado como una habilidad que deben desarrollar todas las personas para lograr ser competitivos en la sociedad actual. A pesar de que no se ha llegado a un consenso acerca de la definición de Pensamiento Computacional, diferentes autores hacen referencia a éste.

Jeannette M. Wing define el Pensamiento Computacional como todos los procesos del pensamiento que se involucran en la formulación y resolución de problemas y que estas soluciones se representan en una forma que puede ser desarrollada por un agente de procesamiento de información [22]. Además, como una habilidad que debe añadirse a la capacidad de análisis de cada estudiante un componente fundamental de la ciencia, la tecnología, la ingeniería y las matemáticas (STEM), aprendizaje que incluye una serie de herramientas mentales que reflejan la amplitud del campo de la informática [17].

El Pensamiento Computacional describe la actividad mental en la formulación de un problema, que admite una solución computacional y que puede ser llevada a cabo por un ser humano, una máquina o por la combinación de ambos. Se superpone al

pensamiento lógico y el pensamiento sistémico, incluye pensamiento algorítmico, pensamiento en paralelo; involucrando así otros tipos de procesos de pensamiento como: razonamiento, composición, patrones, procedimiento, pensamiento recurrente, abstracción, generalización de instancias y parametrización [22].

La Sociedad Internacional para la Tecnología en la Educación (ISTE) y la Asociación de Maestros de Ciencias de la Computación (CSTA) establecieron en [6] una definición operativa para Pensamiento Computacional para Educación Básica y Media (K-12): “proceso de solución de problemas que incluye (pero no se limita a) (i) formulación de problemas de manera que permitan usar computadores y otras herramientas; (ii) organización y análisis de datos de manera lógica; (iii) representación de datos mediante abstracciones; (iii) automatización de soluciones mediante pensamiento algorítmico,…”

Google define el Pensamiento Computacional como “un proceso de resolución de problemas que incluye una serie de características, tales como ordenar de manera lógica y analizar los datos, creando soluciones que emplean una serie de pasos ordenados (o algoritmos), y disposiciones , tales como la capacidad de manejar con confianza complejidad y los problemas de composición abierta”, también lo describen como esencial para el desarrollo de aplicaciones informáticas y que es capaz de soportar la resolución de problemas en diferentes áreas como las ciencias y matemáticas [23].

La Royal Society define el Pensamiento Computacional como el proceso de reconocer los aspectos de la computación en un contexto real, aplicar herramientas y técnicas de la informática para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales [24].

Diferentes grupos profesionales como CSTA, ISTE, entre otros, han desarrollado estándares académicos y actividades con el fin de llevar el Pensamiento Computacional a la educación K-12 [25], [26], [27]; en los programas propuestos por los diferentes grupos se puede notar que la programación es muy usada para fortalecer el CT. La programación es compatible con el Pensamiento Computacional ya que es un proceso creativo que facilita la producción de artefactos computacionales, además proporciona beneficios que permiten aprender estrategias para la resolución de

2. Marco teórico y estado del arte

problemas y diseño, brinda la posibilidad de reflexionar acerca del propio pensamiento e incita a pensar acerca de lo que se está pensando [28], [29], [30].

2.2.Evaluación educativa

Según Scriven, la evaluación se refiere a estimar y verificar la enseñanza impartida, teniendo en cuenta tanto los resultados como el proceso de desarrollo [31]. En [32] el autor la define como una oportunidad para que los estudiantes pongan en práctica sus conocimientos, defiendan sus razones, pregunten acerca de lo que dudan, ignoran o creen; además, la evaluación proponen que debe ser un proceso continuo y procesual, para garantizar que se puede actuar a tiempo y adecuadamente, ante las situaciones que se presenten con los estudiantes. Otros autores añaden que la evaluación es un proceso inferencial ya que no se puede entender todo lo que los estudiantes tienen en su mente [33] y que se enfoca a todo el proceso educativo, ya que en la evaluación se espera analizar la información obtenida para mejorar proceso educativo [34]. Generalmente, la calidad de la evaluación educativa se describe en dos conceptos: validez y fiabilidad, la validez apunta al que y como de la evaluación, y la fiabilidad se refiere al que tan bien está planteada la evaluación para que en condiciones iguales, donde se entregue la misma evidencia el resultado se conserve [35].

En Colombia, el Ministerio de Educación Nacional define la evaluación como el elemento regulador que permite dar una valoración, a partir de evidencias, el progreso de los estudiantes con el fin de mejorar la calidad educativa [36]. El decreto 1290 de 2009 reglamenta la evaluación del aprendizaje y establece los siguientes propósitos de la misma: (a) Identificar las características personales de los estudiantes como intereses y estilos de aprendizaje. (b) Dar información para orientar el desarrollo de los estudiantes. (c) Obtener información para implementar estrategias pedagógicas de acuerdo a las necesidades de los estudiantes. (d) Determinar si un estudiante es promovido. (e) Brindar la información necesaria para el mejoramiento de la calidad institucional [37].

2.2.1. Evaluación formativa

La evaluación educativa es generalmente dividida en dos: evaluación sumativa y evaluación formativa. La evaluación sumativa es descrita como una estimación que se centra en el estudio de los resultados finales [31], cuyos propósitos incluyen: recolectar, interpretar y reportar evidencia del aprendizaje en contraste con los objetivos de aprendizaje [35], casi siempre es calificada y ocurre al final de segmentos de instrucción [38].

La evaluación formativa según Scriven es una estimación que tiene como objetivo servir al perfeccionamiento de la enseñanza y facilitar las decisiones del proceso didáctico[31]. Bennett define a la evaluación formativa como un proceso llevado a cabo durante toda la enseñanza-aprendizaje, que facilita la realimentación con el fin de ajustar la enseñanza para mejorar el aprendizaje de los estudiantes, para que ellos puedan alcanzar los objetivos planteados [33]. Esta evaluación puede ser planeada con anterioridad o se puede generar en cualquier momento durante las sesiones de clase, por ejemplo cuando un docente hace preguntas a sus estudiantes en medio de la instrucción o cuando pide que den ejemplos de los conceptos más recientes [38].

Para mejorar el desempeño de los estudiantes, se requiere que tanto el docente como ellos conozcan oportunamente su progreso. La realimentación se define como la información que se brinda al estudiante acerca de la calidad de su desempeño, considerando la brecha que hay entre sus resultados y los objetivos planteados [39]. Se debe considerar que la realimentación incluye la transferencia de información, regula y está regulada por creencias motivacionales. Según investigaciones la realimentación externa influye en cómo se sienten los estudiantes y cómo aprenden [40]. Por eso es importante, que se brinde desde los docentes hacia los estudiantes, entre estudiantes y desde los estudiantes hacia los docentes [35].

Los autores en [40] definen 7 buenas prácticas de realimentación:

- (i) Ayuda a explicar que es un buen desempeño (objetivos y criterios definidos).
- (ii) Facilita la reflexión de los estudiantes en cuanto a su desempeño.
- (iii) Brinda información relevante a los estudiantes acerca de su aprendizaje.
- (iv) Fomenta el dialogo acerca del aprendizaje entre docentes.
- (v) Estimula la motivación y autoestima.

2. Marco teórico y estado del arte

- (vi) Aporta oportunidades para que el desempeño actual del estudiante se aproxime al deseado.
- (vii) Proporciona información pertinente para que los docentes realicen acciones para mejorar el aprendizaje.

2.2.2. Evaluación Basada en Evidencia

El diseño de evaluaciones basadas en evidencia (ECD: Evidence Centered Design) es un marco de referencia para construir evaluaciones educativas teniendo en cuenta la evidencia que proporciona el estudiante, para brindar una estimación precisa del desempeño dinámicamente y otra información que es difícil de recolectar y analizar con procesos convencionales [41], [42].

Según Bienkowski et Al. [43], ECD es útil cuando los conocimientos y habilidades a medir implican actuaciones complejas que implican múltiples pasos, tales como los requeridos en CT. ECD se basa en la suposición de que la evaluación es esencialmente un argumento probatorio, para lo que se requiere la relación de los factores a evaluar con las acciones desempeñadas por los estudiantes [44].

ECD está integrado por varias capas cuya función es brindar información relevante que permite a los diseñadores estructurar conceptual y operativamente sus evaluaciones. Las capas de ECD son: (1) Análisis del dominio; (2) Modelado del dominio; (3) Marco de evaluación conceptual CAF (por sus siglas en inglés: Conceptual Assessment Framework); (4) Implementación de la evaluación y (5) Entrega de la evaluación.

Análisis del dominio: en esta capa se realiza la categorización de la información más relevante acerca del dominio, se tiene en consideración qué es importante acerca del dominio, qué trabajos y situaciones son importantes en este dominio y qué representaciones del conocimiento son centrales.

Modelado del dominio: esta capa toma la información identificada en la capa de Análisis de dominios para organizarla de acuerdo con los argumentos de evaluación.

Marco de evaluación conceptual (CAF): es la capa encargada de las especificaciones técnicas de los elementos operativos, el CAF se compone de los modelos descritos a continuación:

- Modelo de competencia (CM): se refiere al conjunto de atributos, conocimientos y/o habilidades que se van a evaluar, cada uno de estos se representa por un nodo, la competencia puede ser específica o general, esto depende de quién diseñe la evaluación.
- Modelo de evidencia (EM): expresa como las interacciones de los estudiantes y sus respuestas a un problema, constituyen evidencia sobre la adquisición de competencias especificadas en los nodos de CM, es la unión entre los otros modelos que determinan las competencias (lo que se va a evaluar) y las tareas (de las que se obtienen las pruebas); se compone de dos partes: el modelo estadístico y las reglas de evidencia.
- Modelo de tareas (TM): las variables de este modelo describen las situaciones con las que un estudiante interactúa para proporcionar evidencia sobre aspectos de la competencia.

Implementación de la evaluación: esta capa se encarga de las actividades que se preparan para la administración operativa.

Entrega de la evaluación: incluye los procesos de presentación de tareas a los estudiantes, se evalúa el desempeño para asignar puntajes e informar resultados oportunamente, con el objetivo de brindar realimentación a los estudiantes y soportar la toma de decisiones de los docentes [41].

A través del reporte [45] SRI (centro de investigación independiente) que se encarga de diseñar y validar evaluaciones, soportados en el proyecto PACT (Principled Assessment of CT: Evaluación de principios del pensamiento computacional) exponen la importancia de emplear ECD y enfatizan en su utilidad cuando las habilidades a evaluar implican gestiones complejas como es el caso de CT.

2.3.Programación en el Contexto Escolar

Desde que Seymour Papert desarrollo Logo [46] como un lenguaje de programación para niños, se ha hablado de la inclusión de la programación en el contexto escolar. Concibiendo la programación como un instrumento que brinda la posibilidad de que los niños puedan expresar sus ideas, que requiere constantemente pensar y aprender, además de provocar un cambio para alterar y mejorar los procesos de pensamiento y aprendizaje. Algunas herramientas de programación usadas en contextos escolares y algunas experiencias se describen a continuación.

2.3.1. Herramientas empleadas en prácticas de programación en el contexto escolar

Logo es un lenguaje de programación creado en la década de los 60's introducido inicialmente para enseñar matemáticas, es uno de los lenguajes de programación educativos para los niños más antiguos, permite el empleo del paradigma de programación funcional. Es compatible con los gráficos de tortuga por lo que los usuarios obtienen inmediatamente salidas gráficas, su código es textual, lo que dificulta su uso, por eso se cree que esta fue una de las razones por la cual no se pudo popularizar [46].

Scratch es un entorno gráfico de programación creado para diseñar y desarrollar soluciones con un diseño en bloques gráficos que se puede utilizar por principiantes muy fácilmente, permite crear juegos, simulaciones e historias interactivas, además incluye una comunidad online con millones de jóvenes programadores, que comparten sus creaciones y experiencias alrededor del mundo [29]. Scratch permite usar el paradigma de programación dirigida por eventos e imperativa, ya que la ejecución de los programas lo determina la ejecución de sucesos, ya hayan sido definidos por el usuario o provocados como consecuencia. Este entorno puede leer datos arrojados por diferentes dispositivos externos dotados de sensores que se conectan a él, tales como: Makey Makey, Lego WeDo, teléfonos inteligentes y tabletas (Android) y la tarjeta Arduino. Además, esta herramienta puede ser una manera eficaz de introducir a los estudiantes a los conceptos de programación [47]; por estas razones está siendo usada alrededor del mundo; más de treinta millones de personas de diferentes países

que comparten sus experiencias a través de la comunidad y más de treinta millones de proyectos han sido compartidos hasta la fecha [48].

Stagecast Creator [49] es un lenguaje de programación creado para que los niños desarrollen en sobre una herramienta visual. Se basa en la programación por demostración del concepto, donde las reglas son creadas para dar ejemplos de las acciones que deben realizarse en cada situación dada. Cada acción o grupo de acciones se ejecutan en Java y pueden ser empleadas para construir simulaciones, animaciones y juegos. Creator se basa en la idea de caracteres independientes con una apariencia gráfica y que además poseen propiedades no gráficas. Para cada personaje se define una lista de reglas que determinan su comportamiento. Las reglas son creadas demostrando que puede hacer un personaje en una situación específica.

Alice [50] es una herramienta para la enseñanza – aprendizaje de programación, creado por la Universidad Carnegie Mellon, soporta varios paradigmas de programación; imperativo, funcional, distribuido y concurrente. Proporciona una interfaz visual que facilita su uso, ya que no considera la sintaxis ni codificación, lo que reduce el tiempo de desarrollo, además permite ver los resultados durante la programación y al finalizar la solución.

Stencyl es una herramienta de diseño de videojuegos que ha sido usada por un número de usuarios significativo, gracias a que permite exportar los proyectos para Mac, Windows y Android, y por ofrecer un conjunto de herramientas intuitivas que reducen el tiempo del aprendizaje de la herramienta. Ésta divide su creación en escenas y actores, los cuales pueden ser diferentes tipos de elementos que cuenten con una representación gráfica dentro del juego. Al igual que Stencyl, GameSalad permite crear juegos para diferentes plataformas como: Android, HTML5 y Mac, su funcionalidad es muy similar a la de Stencyl; para cada actor se configuran las reglas que definen su comportamiento, estas pueden ser tomadas desde la biblioteca o personalizadas [51].

Pencil Code [52] es una herramienta de codificación basada en bloques, su diseño permite alternar entre códigos de texto y bloques, lo que facilita el aprendizaje de lenguajes de programación basados en texto, ya que esta manera es más cómoda para realizar la transición a medida de que los estudiantes van conociendo la sintaxis

2. Marco teórico y estado del arte

de lenguajes como JavaScript, CoffeeScript, HTML y CSS. Asimismo permite a los docentes generar vocabularios de bloques, facilitando emplear lecciones específicas sin tener que cambiar a un lenguaje de programación.

Kodu [53] permite a los niños crear juegos en el PC y Xbox a través de un sencillo lenguaje de programación visual, que se puede utilizar para enseñar creatividad, resolución de problemas, narración, y programación, el usuario programa los comportamientos de cada personaje en un mundo 3D y estos programas se expresan en un alto nivel, paradigma sensorial que consiste en un sistema o lenguaje basado en reglas.

Agentsheets [54] es una herramienta que permite la creación de juegos y simulaciones por usuarios de diferentes edades, permitiendo compartirlas en línea, empleando los paradigmas de orientación a objetos, educacional y programación conversacional. Flip [55] es otro lenguaje visual que facilita a las personas sin conocimientos de programación crear sus propios juegos de ordenador utilizando el conjunto de herramientas Neverwinter Nights 2 Electron. Lego [56] ofrece kits de robótica programable de uso comercial, orientado a la programación de robots de forma inteligente y divertida.

App Inventor [57], [58] es un entorno de programación diseñado para crear aplicaciones móviles para teléfonos inteligentes y tabletas Android, debido a su fácil uso es muy recomendado para principiantes, facilita crear aplicaciones transformando el complejo lenguaje de código en texto por bloques de construcción visual fáciles de arrastrar y soltar.

Code.org [59] es una iniciativa que busca la inclusión de la informática en el aula de clase y la equidad de género en las ciencias de la computación, dentro de sus programas estimulan a los niños a realizar diferentes actividades animadas con caracteres populares, dispone de cursos básicos de libre acceso para niños desde los 4 años hasta los 18.

2.3.2. Experiencias de programación en el contexto escolar

En los últimos años, alrededor del mundo se han desarrollado varias iniciativas que incorporan la programación en el contexto escolar [60], [61], [28], debido a la variedad de herramientas disponibles en la actualidad, cada docente puede seleccionar la que más se adapta a las necesidades de su aula de clase; en [62] el autor realiza una comparación de actitud y resultados de aprendizaje entre estudiantes de sexto grado asistentes a un campamento de verano, quienes programaron en Logo y Scratch, en el estudio se concluyó que quienes programaron en Logo primero se encontraron más confiados y desarrollaron más su interés por la programación y quienes programaron primero con Scratch entendieron mejor los condicionales.

La inclusión de CT en la clase de ciencias fue llevada a cabo en [63] a través de la incorporación de prácticas en NetLogo, un entorno programable de agentes múltiples que ha sido ampliamente utilizado [64] tanto en su versión descargable como en la página web, 133 estudiantes de grado noveno participaron en el estudio, primero respondieron a un pre-test, después participaron en sesiones y finalmente respondieron un pos-test, después de analizar los datos pudieron identificar la ayuda de las sesiones en la capacidad de los estudiantes para explorar modelos computacionales.

Los autores en [65] emplearon Stagecast Creator como herramienta de programación con niñas de 12 a 14 años asistentes a una clase extracurricular, revelando que mediante la creación de juegos las estudiantes pudieron entender los conceptos de ciencias de la computación.

Alice fue usado como herramienta de programación en [66] con la participación de 138 adolescentes quienes desarrollaron tres tareas, cada una requería que el estudiante identificara e integrara varias piezas de solución con el fin de que los programas funcionaran correctamente. También en [67] donde los autores realizaron un campamento de verano con 31 estudiantes entre 10 y 17 años, quienes trabajaron con las herramientas Kodu, Alice y Lego, la interfaz de Kodu permitió que rápidamente conocieran la herramienta y pudieran trabajar con ella, lo que permitió una transición suave a Alice y posteriormente a Lego, según sus hallazgos hubo más aceptación al trabajar con robots de Lego. Y en [68] las autoras emplearon Alice y Chatbot (un programa informático que puede ser programado para sostener conversaciones con humanos u otros bots) Chatbot en dos estudios observacionales una competencia en

2. Marco teórico y estado del arte

línea y en una clase de 15 lecciones en dos escuelas secundarias, sus resultados revelaron que hubo mayor aceptación empleando Chatbot.

Pencil Code se ha usado en [52] en una prueba a un grupo de estudiantes de secundaria, ninguno de ellos tenía experiencia previa en programación, sin embargo el estudio reveló que en un 95% hicieron transición entre bloques y texto incluso cuando no se requería trabajar en texto.

El proyecto iDREAMS descrito en [69] busca incorporar la informática en los colegios mediante el diseño de juegos, su objetivo es extender la participación de estudiantes en las ciencias de la computación, teniendo en cuenta diferentes factores como el Diseño de Juegos Escalable como principal enfoque y el análisis y evaluación a través de los patrones de CT, que se describen más adelante. Más de 500 estudiantes de secundaria emplearon AgentSheets para la creación de sus juegos durante el estudio los autores encontraron en las respuestas de los estudiantes nuevos patrones que no habían considerado [70].

Cincuenta y cinco estudiantes entre los 12 y 13 años participaron en un curso de creación de juego empleando Flip, entre los hallazgos los docentes sugirieron que la herramienta permitió a los estudiantes crear juegos complejos e iniciaron a dominar habilidades computacionales básicas desarrollando habilidades computacionales a través de actividades que disfrutaron y fueron motivadoras [71].

En [72] los autores cuentan su experiencia empleando robots Lego en un campamento en la India con niños de 7 a 13 años, donde concluyeron que al integrar programación con robots los niños se divirtieron, aprendieron y aumentaron su nivel de compromiso y motivación.

Después de desarrollar un currículo de robótica e implementarlo en un colegio público los autores evaluaron a 112 estudiantes acerca de su experiencia a través de la programación de un robot con un instrumento diseñado por los investigadores, quienes se aseguraron se incluyera los factores más importantes de CT mediante respuestas múltiples, preguntas abiertas y programación robótica y razonamiento cotidiano [73].

App Inventor fue empleado en [74] con 29 estudiantes de 8vo y 9no con la finalidad era fomentar las habilidades CT en los estudiantes a través de la programación de aplicaciones. En [75] los autores describen una experiencia en que 111 estudiantes participaron en un campamento donde desarrollaron aplicación con la herramienta App Inventor y se analizaron sus desarrollos con una herramienta creada por los autores.

A través de un estudio comparativo entre tres comportamientos asumidos por estudiantes de cuarto grado mientras programan Scratch, los autores de [76] exhiben como los niños asumen diferentes formas para programar, considerando que algunos estudiantes pueden tardar más tiempo que otros en el desarrollo de habilidades CT, los hallazgos de los investigadores ponen en evidencia las diferencias que deben ser consideradas por los docentes de K-12.

La participación de 44 estudiantes entre los 8 y 17 años a un curso de 2 semanas se detalla en [77], durante la primera semana los participantes interactuaron con robots y en la segunda desarrollaron juegos usando Scratch, los investigadores analizaron la relación de los conceptos CT y encontraron relaciones de cómo los niños codifican.

A través de un estudio cuasi experimental el autor midió la incidencia de CT y la resolución de problemas en los estudiantes de secundaria, los grupos fueron formados así: experimental con 27 estudiantes y control con 25 estudiantes, ambos de grado octavo, quienes en su curso de programación presentaron un pre y pos-test que incorporaba conceptos computacionales y resolución de problemas por medio de la programación en Scratch, entre los hallazgos reafirmaron la importancia de brindar espacios donde los estudiantes puedan desarrollar estas habilidades, ya que los resultados indicaron una mejora [78].

Siguiendo dos enfoques de enseñanza: construccionista y 4C/ID (cuatro componentes / diseño instruccional) , los autores de [79] realizaron un estudio comparativo basado en diseño, para lo cual enseñaron en dos escuelas dos conjuntos de lecciones teniendo en cuenta los enfoques de enseñanza propuestos; emplearon la herramienta Scratch con 87 estudiantes entre 9 y 12 años, los resultados mostraron una ganancia de aprendizaje considerable comprobada en los resultados del pre-test y post-test realizados.

2. Marco teórico y estado del arte

En [80] se describe un estudio piloto para soportar el diseño de juegos de cambio climático, para lo cual realizaron un taller de 4 días, donde además de discutir la temática, los estudiantes fueron introducidos a la programación visual de Scratch, los resultados fueron exitosos, llevando a los investigadores a desarrollar un plan de estudios que fue implementado por dos docentes, dejando como trabajo futuro la exploración de la medición de diferentes formas de evaluar CT y su automatización.

En [81] los autores realizan una comparación de los resultados del impacto de la programación con Scratch en tres colegios diferentes con 129 niños de grados 2do y 6to, los resultados arrojaron que la curva de aprendizaje se acelera en los grupos experimentales, tanto en factores cognitivos como de motivación.

2.4.Estado del Arte

2.4.1. Marcos de Referencia y Evaluación del Pensamiento Computacional

Marco de Referencia de CT y Evaluación del aprendizaje a través del diseño

Según Brennan y Resnick [82], el Pensamiento Computacional involucra tres dimensiones claves: conceptos, prácticas y perspectivas computacionales.

Conceptos computacionales: son comunes en los diferentes lenguajes de programación, los definidos en este marco de referencia tienen una correspondencia directa con los bloques de Scratch, los conceptos son: secuencias (expresar una tarea en una serie de pasos), bucles (repetición de pasos), paralelismo (secuencia de instrucciones ejecutándose al mismo tiempo), eventos (una tarea se efectúa como consecuencia de un acontecimiento), condicionales (ejecutar determinada acción basando en una condición), operadores (permite manipular datos numéricos y String) y datos (almacenamiento, recuperación y actualización de información).

Prácticas computacionales: las prácticas corresponden a la manera como el estudiante aborda el problema y programa, son: incremental e iterativo (construir la solución por pequeños pasos), pruebas y depuración (esta práctica se refiere a la

facultad de ir probando cada una de las partes a través del desarrollo), reutilización y remezcla (la habilidad de tomar un código existente y emplearlo en un nuevo proyecto), y abstracción y modularización (construir algo grande y complejo, partiendo de pequeñas partes).

Perspectivas computacionales: se refieren a cómo perciben los estudiantes el proceso de resolución de problemas a través de la programación, las perspectivas son: expresar (comunicarse con diferentes personas interesadas para intercambiar información de lo que se está desarrollando y la manera como se hace), conectarse (implica compartir con otros desarrolladores y buscar entre los diferentes proyectos) y cuestionarse (incluye preguntarse por todo aquello que los rodea, y en algunos casos responder a través del diseño).

Para evaluar las dimensiones en contextos donde los estudiantes empleen Scratch, los autores proponen tres aproximaciones [82]:

Aproximación 1- Análisis de proyectos: corresponde al registro de los bloques usados en los proyectos y la frecuencia con la que se han aprovechado. Como se trata del análisis del producto final, esta aproximación deja de lado las prácticas y las perspectivas centrándose en los conceptos computacionales que se encuentran en correspondencia directa con los bloques de Scratch.

Aproximación 2- Entrevistas basadas en artefactos-objetos: los estudiantes seleccionan un par de proyectos y sobre estos se realiza una entrevista que incluye preguntas para evaluar tanto los conceptos como las prácticas computacionales. Se inicia indagando acerca de la motivación inicial, después de la evolución del proceso de desarrollo del proyecto y finalmente acerca del resultado y la satisfacción con el producto final. Al ser entrevistas individuales, se requiere de mucho tiempo para el evaluador, además de las limitaciones propias de las entrevistas como las distorsiones de la memoria al recordar algo sucedido.

Aproximación 3- Diseño de escenarios: los evaluadores proporcionan los proyectos Scratch a los estudiantes, brindan escenarios y un código inicial que en un principio son de baja complejidad, la que se van incrementando a medida que el estudiante avanza en la evaluación; los evaluadores piden a los estudiantes que: a) expliquen

2. Marco teórico y estado del arte

qué es lo que hace el código del proyecto; b) describan cómo el proyecto podría ser ampliado; c) arreglen los errores de código del proyecto; y d) re mezclar el proyecto. A pesar de que requiere gran consumo de tiempo por parte de los evaluadores, cuenta con la ventaja de evaluar en tiempo real los conceptos y prácticas computacionales relacionadas con el proyecto.

Marco de referencia para el análisis de Patrones del Pensamiento Computacional

El origen de los Patrones de Pensamiento Computacional (Computational Thinking Patterns: CTP) viene de la exploración de desarrollar el CT entre diferentes dominios de aplicación, específicamente entre el diseño del juego y las simulaciones de ciencias. Los autores en [83] describen los CTP que surgieron como resultado de la investigación realizada:

- Collision: Un evento donde dos agentes chocan.
- Push: Un objeto empuja otro.
- Pull: Un objeto hala uno o varios objetos.
- Transport: Un objeto lleva a otro.
- Generation: Se generan muchos objetos de una fuente.
- Absorption: Un objeto absorbe una corriente de objetos.
- Choreography: Coordinación de movimientos.
- Diffusion: Propagación de partículas físicas o elementos conceptuales.
- Path Finding / Seeking: Los objetos se mueven hacia algún objetivo en un entorno complejo como un laberinto.
- Collaborative Diffusion: Varios objetos colaboran o compiten entre sí.
- Multiple Needs: Objetos o caracteres que siguen diferentes necesidades.

Los CTP se pensaron inicialmente como un marco teórico que conceptualiza la interacción entre los objetos, permitiendo a los estudiantes seccionar su desarrollo y describir los CTP encontrados. Cada CTP incluye una descripción del fenómeno e instrucciones teóricas para hacer operativo el patrón, basado en la representación de las interacciones de los objetos, independiente del lenguaje de programación.

El análisis de CTP fue diseñado para valorar el significado semántico de un desarrollo, emplea una técnica basada en LSA (Latent Semantic Analysis), comparando el juego o simulación con cada CTP para generar un gráfica que muestra la evaluación que corresponde en cada patrón.

Con el fin de recolectar y analizar los juegos y simulaciones creadas por los estudiantes, Ioannidou et Al. [83] presentan la construcción de una infraestructura para el aprendizaje, llamada Scalable Game Design Arcade, que analiza la solución en términos de los CTP, muestra los resultados en gráfica y también las similitudes de la programación con otros proyectos. El uso de esta herramienta está limitado a los estudiantes que usan AgentSheets en las escuelas participantes en el proyecto.

Marco de referencia para el diseño y evaluación de códigos de programación visual

Los autores proponen un marco de referencia para la evaluación de lenguajes de programación visual (VPLEF: Visual Programming Languages Evaluation Framework) [84].

Este marco fue desarrollado teniendo en cuenta algunos elementos divididos en dos grupos: funcionalidad y mantenimiento.

Funcionalidad: se relaciona con el proceso de desarrollo.

- **Efectividad:** Está relacionado con el correcto funcionamiento del programa, si cumple con las necesidades del usuario y si cumple con las metas que se establecieron desde el inicio.
- **Seguridad:** Se refiere a si las variables se inicializan adecuadamente y las iteraciones terminan después de un cierto número de pasos.
- **Interactividad del usuario:** Valora si hay una interacción adecuada con el usuario.
- **Abstracción:** Estima si se puede descomponer el problema en piezas más pequeñas y simples.
- **Sincronización:** Determina si la comunicación entre los objetos es correcta, si los comandos son apropiados para ayudar al flujo del programa y si están correctamente definidos.

2. Marco teórico y estado del arte

- Representación de los datos: Las variables están configuradas correctamente y se usan de manera apropiada, se emplean estructuras de datos (listas).
- Paralelismo-linealidad: hay una correcta diferencia entre procesos paralelos y lineales.

Mantenimiento: se refiere al código terminado y las modificaciones que se permiten.

- Legibilidad: Se refiere a si el código es fácilmente comprensible, las variables tienen nombre apropiados y si hay coherencia en la ubicación de los bloques.
- Capacidad de prueba: Se relaciona con qué facilidad se puede probar la corrección del programa.
- Posibilidad de cambiar: Es la facilidad con la que se puede extender o modificar el código.
- Optimización: Comprueba si se pueden expresar los algoritmos en menos pasos y llegar al mismo resultado, hay código repetido, se emplean todos los bloques o hay código muerto, son necesarias todos los bloques para que el código funcione correctamente.

Karvounidis et Al. [84], brindan un marco de evaluación de lenguajes de programación visual, los criterios seleccionados cumplen una correspondencia con el uso de bloques y la facilidad de probar, optimizar y realizar pruebas sobre el desarrollo, para la evaluación emplean la herramienta codOrama, aunque esta cuenta con un limitado número de bloques Scratch para su análisis. Los autores esperan con el tiempo enriquecer el sistema con más evaluaciones para hacerlo más robusto y compararlas evaluaciones futuras con los resultados obtenidos anteriormente.

2.4.2. Herramientas y otras aproximaciones para la evaluación de CT

Para la evaluación de CT, diferentes autores proponen aproximaciones como herramientas automáticas y semi-automáticas, pruebas escritas, definiciones de sistemas, etc.

El estudio expuesto por Kasimoglu en [85] analiza la relación entre el Pensamiento Computacional y el aprendizaje de la programación dentro de un contexto de juegos

serios. El documento sostiene que los juegos serios desarrollados en la actualidad específicamente con fines de aprendizaje de programación, no consideran un juego profundo para el desarrollo de habilidades de CT. Para hacer frente a este problema, el documento propone ejercicios que integran la programación de un robot, evidenciando resultados satisfactorios relacionados con los niveles de motivación de los estudiantes. En la implementación, presentan un conjunto limitado de ejercicios con un nivel bajo de complejidad donde la programación es el fin, no considerar los diferentes conceptos y dimensiones que abarca el CT.

En [11], el artículo presenta un conjunto de evaluaciones en línea interactivas diseñadas para el fortalecimiento del Pensamiento Computacional en aulas de secundaria que trabajan bajo un enfoque STEM. Cada evaluación incluye herramientas interactivas que evidencian el poder de la computación en la práctica de la investigación científica y matemática. Las herramientas computacionales utilizadas en dichas evaluaciones permiten a los estudiantes analizar los datos con visualizaciones dinámicas y explorar conceptos con los modelos computacionales. El trabajo se enfoca a fortalecer CT, no obstante, no hay una evidencia concreta del desarrollo de la habilidad, ni una especificación clara del proceso que debe llevar a cabo el docente, dificultando la inclusión de ejercicios propios para evaluar a los estudiantes.

Fairy Assessment es un programa para analizar el pensamiento algorítmico, la abstracción y el modelado. Los autores describen los primeros resultados de la herramienta de evaluación para la medición del Pensamiento Computacional en estudiantes de Educación Media bajo un diseño de investigación experimental. Los resultados evidencian la necesidad de contar con un conjunto de problemas contextualizados, no predefinidos. Dentro de lo expuesto en el artículo, los autores consideran la importancia que tiene la evaluación del Pensamiento Computacional usando herramientas de programación como Scratch [18].

El trabajo expuesto en [10], investiga el desarrollo de habilidades de Pensamiento Computacional de los estudiantes en el contexto de la Robótica Educativa. El estudio emplea un modelo para Pensamiento Computacional y explora las habilidades de los estudiantes en el desarrollo de las actividades de aprendizaje, evaluándolas de manera oral y escrita. El trabajo se realizó con 164 estudiantes de diferentes edades

2. Marco teórico y estado del arte

y niveles educativos. Este trabajo se centra en evaluar las habilidades del Pensamiento Computacional en el contexto de la Robótica Educativa dejando de lado la programación, sin hacer referencia a casos problema ni su integración con los procesos de enseñanza-aprendizaje.

En [86] se expone el marco de trabajo del Pensamiento Computacional (CTF: Computational Thinking Framework) que sirve como referencia para diseñar recursos de aprendizaje del Pensamiento Computacional y evaluar a los estudiantes, el objetivo se centra en unificar las opiniones de lo que debe involucrar el pensamiento computacional. En su estudio los investigadores aplicaron el marco en Light-Bot, un juego educativo dirigido a la programación, y encontraron que el marco provee una manera para reforzar los conceptos de la informática en el juego. Los autores en [87] probaron también el marco de trabajo con estudiantes del primer año universitario, a los que les aplicaron preguntas propuestas por las olimpiadas de informática en busca de talentos, que busca encaminar a los estudiantes con aptitudes hacia carreras tecnológico-científicas.

Román-González propone en [12] una prueba para evaluar el Pensamiento Computacional en la que considera cinco dimensiones, emplea un instrumento de respuesta múltiple con 28 preguntas y cuatro opciones de respuesta por pregunta, cada una las alternativas de respuesta se pueden representar con flechas o bloques visuales, o de manera textual. El autor expone en [88] que al tratarse de preguntas de opción múltiple, la prueba tiene limitaciones, además únicamente considera medir los niveles más bajos de complejidad (reconocer y entender, según los referentes del autor) de manera sumativa, dejando a un lado los problemas cuya resolución requiere que los estudiantes empleen de forma creativa diferentes conceptos computacionales.

La iniciativa propuesta en [89] responde a una necesidad planteada desde el primer ministro de Malasia, quien anuncio que se debe incluir CT en el currículo iniciando con los niños de primaria, los autores proponen unas rúbricas para las pruebas escritas en matemáticas, ciencias y lengua, dónde se indaga acerca de la temática y se evalúa tanto la temática como los elementos de CT que han definido. La propuesta se encuentra ligada al currículo del país.

En [16] proponen REACT (Real Time Evaluation and Assessment of Computational Thinking), una herramienta de evaluación formativa gráfica en tiempo real basada en el marco de referencia para el análisis de los CTP [83], ésta permite a los docentes ver rápidamente el nivel de sus estudiantes de manera individual o grupal, de acuerdo a los patrones establecidos en el marco de referencia. La mayor limitación de esta herramienta es que se encuentra disponible únicamente para usuarios de AgentSheets o AgentCubes inscritos en Scalable Game Design Arcade.

Algunas herramientas relacionadas con la evaluación de CT para proyectos Scratch corresponden a Scrape, Hairball, Dr. Scratch, Ninja Code, codOrama y SAT, descritas a continuación:

Scrape [13] es una herramienta de visualización que analiza bloques de programación al interior de proyectos Scratch, sin embargo, su uso está limitado a los archivos de la versión 1.4, lo que impide su uso en la actualidad.

Hairball [14] proporciona un marco de referencia y un conjunto de plugins para analizar la aplicación de programas realizados en Scratch.

Dr. Scratch [15] es una herramienta web gratuita de código abierto que analiza proyectos de Scratch en términos de abstracción, descomposición del problema, paralelismo, pensamiento lógico, sincronización, control de flujo, interactividad del usuario y representación de los datos, asignando de forma automática una calificación a cada factor. Sin embargo en [80] los autores mencionan que actualmente no contemplan factores como la longitud del desarrollo, número de variables, y algunas características cualitativas que son elementos fundamentales de CT y van a ser incluidas.

Ninja Code [90] es una página web que facilita el análisis automático de proyectos Scratch en términos de: condicionales, bucles, procedimientos, procedimientos comunes, datos, eventos, paralelismo e interfaz con el usuario, divididos en cuatro niveles de complejidad; aunque tienen en cuenta más niveles de competencia, únicamente consideran tres tipos de proyectos para dar el resultado, otra de las debilidades es que se encuentra muy poca información disponible en inglés.

2. Marco teórico y estado del arte

La herramienta visual codOrama [84] facilita el análisis de proyectos realizados en lenguajes de programación visual, particularmente Scratch. A través de una tabla de dos dimensiones simplifica el complejo escenario de análisis; organizando los bloques empleados de acuerdo a los elementos CT establecidos en el marco de evaluación anteriormente expuesto, sin embargo como se mencionó los autores solo han considerado un número limitado de bloques para su análisis.

SAT [91] es una herramienta web que permite analizar proyectos Scratch, es basada en Dr. Scratch y considera las mismas dimensiones y niveles, pero según sus pruebas los resultados muestran que tienen mayor tasa de éxito y en otras métricas de rendimiento superan los valores. Según sus autores la mayor limitación corresponde a que algunos elementos de CT no se pueden medir únicamente analizando el código estático, se requiere recibir los comentarios de los estudiantes para conocer sus habilidades de diseño, depuración y remezcla.

FCS1 Assessment Instrument [92] es un instrumento que considera los conceptos fundamentales de las ciencias de la computación como: operadores lógicos, condicionales simples y compuestos, bucles definidos e indefinidos, matrices, funciones simples y con parámetros, recursividad, variables, y orientación a objetos, para evaluar la adquisición de los conceptos fundamentales de la computación en los estudiantes.

2.5. Conclusiones del estado del arte

Conforme a lo expuesto anteriormente, se puede decir que hay carencia de estrategias y herramientas que proporcionen a los docentes información acerca del desarrollo de CT de sus estudiantes y les facilite el proceso de evaluación. Aunque existen algunas propuestas como: marcos de referencia, herramientas, pruebas sumativas y otras aproximaciones, la evaluación de CT tiene un amplio camino por recorrer, ya que se considera pertinente que ésta incluya todos los aspectos que definen la evaluación, particularmente la evaluación formativa. Abordar este desafío exige de nuevas herramientas y actividades instruccionales, pero también requiere de nuevas formas de evaluar que consideren los aspectos más relevantes de CT [93].

Es importante destacar que es cotidiano verificar en distintas fuentes que la estimación del desarrollo de CT se realiza de manera sumativa, limitándose a una prueba final al terminar un proyecto o test, pero esto debe ser una práctica complementaria, ya que el proceso de evaluación debe comenzar desde la instrucción inicial y permanecer siendo continuo. Debe abarcar todo el proceso de enseñanza-aprendizaje. Para permitir al docente adquirir información en todo momento, para tomar decisiones en el proceso de enseñanza y lograr mejoras. También para facilitar al estudiante la realimentación continua, necesaria para mejorar su aprendizaje de manera oportuna.

Capítulo 3

3. Caracterización de los criterios mínimos para el desarrollo del pensamiento computacional y los mecanismos para su evaluación

En este capítulo se describen los procedimientos realizados para identificar y caracterizar los criterios mínimos considerados para estimar el desarrollo de CT y los mecanismos que se emplean para evaluarlos, el capítulo se divide entre la metodología empleada en las dos caracterizaciones a ser abordadas y el desarrollo de las fases que la componen.

A continuación, se brinda una descripción del marco metodológico.

3.1. Metodología

La caracterización de los criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación se realizó a través de una revisión guiada, siguiendo la metodología planteada en "Modelo integral para el profesional en ingeniería" [94], la cual propone las siguientes fases:

1. **Fase Preparatoria.** Consiste en la definición del método, la temática y los núcleos temáticos.
2. **Fase Descriptiva.** Se compone de la identificación y revisión documental de los diferentes estudios acerca del tema.
3. **Fase de Interpretación por Núcleos Temáticos.** Se realiza el análisis basado en cada núcleo temático, para conseguir las evidencias y derivar resultados.
4. **Fase de Construcción Teórica Global.** Se establecen el conjunto de resultados del estudio.
5. **Fase de Extensión y Publicación.** Se trata de la divulgación de los resultados obtenidos.

La metodología se soporta en los factores para la realización del estudio:

(1) Aspectos formales, (2) Asunto investigado, (3) Delimitación contextual, (4) Propósito, (5) Enfoque, (6) Metodología, (7) Resultados, (8) Observaciones.

La fase preparatoria abarca los primeros seis factores, los resultados y observaciones serán incluidos en las fases: descriptiva, de interpretación y de construcción teórica global.

En seguida, se da una descripción del marco metodológico, seguido del desarrollo de cada una de las fases propuestas. Se inicia con la consulta a fuentes de información primarias como los planteamientos teóricos definidos por la academia y la industria, aproximaciones y herramientas de evaluación, después se procede a realizar un mapeo entre las fuentes para concluir con una relación de los criterios mínimos de CT.

Se ha podido notar que además de la falta de consenso en la definición de CT [95], los elementos que la componen se encuentran en diversos términos y diferente grado de especificidad, por este motivo ha sido importante realizar un mapeo para unificar términos semánticamente y evitar ambigüedades.

Más adelante, se realiza la identificación y caracterización de los criterios y mecanismos de evaluación de CT.

3. Caracterización de criterios mínimos y mecanismos de evaluación

3.2.Fase preparatoria

3.2.1. Aspectos formales

La caracterización se realiza en el marco del proceso investigativo para el proyecto "Modelo de evaluación basada evidencia para estimar el desarrollo del Pensamiento Computacional", para el cual se busca identificar el conjunto de criterios que son tenidos en cuenta en los planteamientos de teóricos y estudios prácticos para estimar el desarrollo de CT, de igual manera caracterizar los mecanismos que se emplean en la evaluación de dichos criterios.

3.2.2. Asunto investigado

El siguiente estudio se enfoca en identificar los criterios mínimos que permiten determinar el desarrollo de CT y los mecanismos empleados con el fin de evaluarlos.

3.2.3. Delimitación contextual

La temática corresponde a los criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación, se consideran las propuestas de autores reconocidos en el área y otros autores que han presentado implementaciones y aproximaciones de evaluación CT, se consideran también los criterios mínimos y mecanismos definidos a partir de prácticas de programación en el contexto escolar.

Asimismo, se tienen en consideración los elementos que componen CT, según diversas definiciones teóricas planteadas por la academia, por la industria, en algunas aproximaciones, y herramientas.

3.2.4. Propósito

El estudio tiene por objeto "caracterizar los criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación", considerando los elementos necesarios para aportar en la evaluación requerida por el docente. En este sentido, con la identificación de criterios de CT y sus mecanismos de evaluación, se constituirá un importante hito a partir del cual será posible establecer una guía para la estimación del desarrollo de CT, especialmente en el contexto escolar, aportando mayor orden y completitud al proceso.

Por lo anterior, las preguntas guía son: ¿Cuáles son los criterios mínimos para estimar el desarrollo de CT mediante prácticas de programación en el contexto escolar? ¿Qué mecanismos se pueden emplear para evaluar los criterios mínimos?

3.2.5. Enfoque

Con el fin de enmarcar el concerniente estudio, se hace necesario identificar los conceptos relevantes a emplear, con el propósito de reducir la posible ambigüedad en la interpretación, de la siguiente manera: elemento, aproximación, criterio mínimo y mecanismo de evaluación.

- **Elemento:** parte constitutiva o integrante de algo.
- **Aproximación:** toda definición de modelo, marco de referencia, o recomendación que brinde información acerca del desarrollo de CT y la estimación de su desarrollo.
- **Criterio mínimo:** regla o norma que alcanzado su menor grado posible de valor o medida acorde a la cual se busca implantar juicios y tomar decisiones.
- **Mecanismo de evaluación:** son los medios a través de los cuales se recoge la información acerca del proceso de enseñanza-aprendizaje.

3.2.6. Metodología

El presente estudio será cumplido con una investigación de tipo exploratoria sobre diferentes especificaciones teóricas, aproximaciones, herramientas, estudios prácticos e investigaciones, de manera que sobre los criterios mínimos para estimar del

3. Caracterización de criterios mínimos y mecanismos de evaluación

desarrollo CT y sus mecanismos de evaluación se consideren los datos con mayor pertinencia.

El estudio se cumplió por medio de las siguientes actividades:

a) Identificación de aportes teóricos, aproximaciones, herramientas e implementaciones de estudios relacionados con el desarrollo de CT y su proceso de estimación, fundamentados en la consecución de documentación técnica- científica que las respalda, a fin de hallar con mayor precisión los componentes de caracterización en relación con el objetivo propuesto.

Teniendo en cuenta la diversidad de aportes existentes, se procede a acotar el estudio seleccionando una muestra, según la aplicación de los siguientes criterios:

- Consideración de los documentos más relevantes acerca del tema.
- Actualidad en su desarrollo, preferiblemente en el periodo de los últimos siete (7) años.
- Documentación soportada en artículos relacionados en índices bibliográficos relevantes.

Los resultados de la ejecución de esta actividad se exponen en la Fase Descriptiva.

b) Mapeo de términos. Dada la heterogeneidad de términos, se hace necesario identificar el contexto de cada elemento brindado y relacionarlo con sus complementarios, a fin de captar correspondencias semánticas (proceso de mapeo); cuando esto ocurre, se procede a generar un término genérico. El agregado de términos genéricos puede apreciarse en el apartado de Fase de Interpretación por Núcleos Temáticos.

c) Análisis de Elementos. Como ya se han identificado los elementos, criterios mínimos y mecanismos, en este punto se realiza la interpretación de componentes entre todos los objetos de estudio. El resultado de esta actividad puede observarse en la Fase de Construcción Teórica Global.

3.3. Fase Descriptiva.

En esta fase se identifican los elementos que componen CT, los criterios mínimos para la estimación del desarrollo de CT y sus mecanismos de evaluación propuestos en la literatura y que fueron considerados los más relevantes.

Para la caracterización de los elementos identificados, se relacionaron los aportes que proponen elementos para el desarrollo CT. Los criterios mínimos y mecanismos fueron caracterizados teniendo en cuenta aportes teóricos, herramientas e implementaciones en el aula que estiman el desarrollo de CT.

3.3.1. Elementos del Pensamiento Computacional

Los elementos de CT son las partes que componen el conjunto de habilidades del pensamiento computacional. En la revisión documental, fue posible establecer con respecto a los elementos de CT, que existen especificaciones teóricas que sugieren los elementos "que deberían" tenerse en cuenta según las definiciones de: (1) autores expertos del área; (2) asociaciones educativas y universidades; (3) compañías dentro de la industria informática y (4) aproximaciones como marcos de referencia y modelos; (5) herramientas diseñadas para la evaluación de CT o para evaluar prácticas de programación y (6) propuestas implementadas en el aula de clase.

- Caracterización de elementos de CT considerados por autores

Se tuvieron en cuenta dentro de esta caracterización los estudios en los que los autores han considerado diferentes trabajos para llegar a sus conclusiones, además se revisó la documentación en la que han soportado con el fin de garantizar confiabilidad en el proceso de caracterización, los resultados se encuentran consignados en la Tabla 1.

Tabla 1. Caracterización de elementos de CT considerados por autores

Referencia	Elementos CT
------------	--------------

3. Caracterización de criterios mínimos y mecanismos de evaluación

Valerie J. Shute, Chen Sun, y Jodi Asbell-Clarke [96]	<ul style="list-style-type: none"> -Descomposición -Abstracción -Algoritmos -Depuración -Iteración -Generalización
Sushi Grover y Roy Pea [2]	<ul style="list-style-type: none"> -Abstracción y generalización de patrones -Tratamiento sistemático de la información -Sistemas de símbolos y representaciones (información) -Nociones algorítmicas de flujo de control. -Descomposición ç de problemas estructurados (modularización) -Pensamiento iterativo, recursivo y paralelo -Lógica condicional -Limitaciones de eficiencia y rendimiento -Depuración y detección sistemática de errores
Marina Umaschi Bers [97]	<ul style="list-style-type: none"> -Algoritmos -Modularidad -Estructuras de Control -Representación -Hardware y software -Proceso de diseño -Depuración
Ting-Chia Hsu, Shao-Chen Chang y Yu-Ting Hung [98]	<ul style="list-style-type: none"> -Abstracción -Diseño de algoritmos -Automatización -Análisis de información -Recolección de información -Representación de información -Descomposición -Paralelismo -Reconocimiento de patrones -Generalización de patrones -Simulación -Transformación -Logica condicional -Conección con otras áreas -Visualización

	<ul style="list-style-type: none"> -Depuración y detección de errores -Eficiencia y desempeño -Modelado -Solución de problemas
--	--

- **Caracterización de elementos de CT considerados por asociaciones educativas y universidades**

Algunas asociaciones educativas y universidades han emitido reportes y recomendaciones acerca de CT, con el objetivo de facilitar su incorporación en el proceso de enseñanza-aprendizaje, en estos reportes los involucrados en educación han incluido los elementos que consideran componen CT, estos se puntualizan a continuación en la Tabla 2.

Tabla 2. Caracterización de elementos de CT considerados por asociaciones educativas y universidades

Referencia	Elementos
Computer Science Teachers Association CSTA – International Society for Technology in Education ISTE [26], [26], [99]	<ul style="list-style-type: none"> -Recolección de información -Análisis de información -Representación de información -Descomposición de problemas -Abstracción -Algoritmos y procedimientos -Automatización -Simulación -Paralelismo
Computing At School CAS [100], [101]	<p>Conceptos:</p> <ul style="list-style-type: none"> -Razonamiento lógico -Algoritmos -Descomposición -Patrones y generalización -Abstracción: -Evaluación: <p>Aproximaciones:</p> <ul style="list-style-type: none"> -Experimentar -Crear

3. Caracterización de criterios mínimos y mecanismos de evaluación

	<ul style="list-style-type: none"> -Depurar -Perseverar -Colaborar <p>Técnicas:</p> <ul style="list-style-type: none"> -Reflexionar -Codificar -Diseñar -Analizar -Aplicar
College Board [102]	<p>Prácticas:</p> <ul style="list-style-type: none"> -Conexión de la computación -Crear artefactos computacionales -Abstracción -Analizar problemas y artefactos -Comunicar -Colaborar
Queen Mary – University of London [103]	<p>Elementos:</p> <ul style="list-style-type: none"> -Pensamiento lógico -Pensamiento algorítmico -Soluciones eficientes -Pensamiento científico -Pensamientos innovadores
Brigham Young University [104]	<p>Componentes:</p> <p>Habilidades:</p> <ul style="list-style-type: none"> -Descomposición -Reconocimiento de patrones -Abstracción -Diseño de algoritmos -Evaluación <p>Actitudes:</p> <ul style="list-style-type: none"> -Seguro, confiado -Comunicativo -Flexible <p>Enfoques:</p> <ul style="list-style-type: none"> -Experimentar -Crear -Depurar -Ser Perseverante -Colaborar

- **Caracterización de elementos de CT considerados por compañías de la industria informática**

Google y Microsoft han dispuesto de documentación referente a CT en sus páginas web, dentro de la información allí incluida, en la Tabla 3 se pueden apreciar los elementos CT propuestos.

Tabla 3. Caracterización de elementos de CT considerados por compañías de la industria informática

Referencia	Elementos
Google [23]	<ul style="list-style-type: none"> -Abstracción -Diseño de algoritmos -Automatización -Recolección de información -Análisis de información -Representación de información -Descomposición -Paralelismo -Generalización de patrones -Reconocimiento de patrones -Simulación
Microsoft [105]	<ul style="list-style-type: none"> -Descomposicion -Reconocimiento de patrones -Abstracción -Pensamiento algorítmico

- **Caracterización de elementos de CT considerados por autores que describen marcos de referencia**

La Tabla 4 relaciona la denominación de los marcos de referencia considerados para el estudio y los elementos que internamente son considerados en cada uno de los marcos.

3. Caracterización de criterios mínimos y mecanismos de evaluación

Tabla 4. Caracterización de elementos de CT considerados por autores que describen Marcos de Referencia

Referencia	Elementos CT
“Developing computational thinking in the classroom: A framework” [106] y “Evidence of assessing computational thinking” [107] por Cynthia Selby, Mark Dorling y John Woollard	Conceptos: -Abstracción -Descomposición -Pensamiento algorítmico -Evaluación -Generalización
“New frameworks for studying and assessing the development of computational thinking” [82] por Karen Brennan y Mitchel Resnick	Conceptos: -Secuencias -Bucles -Paralelismo -Eventos -Condicionales -Operadores -Información Prácticas: -Incremental e iterativo -Pruebas y depuración -Reutilización y remezclado -Abstracción y modularización Perspectivas: -Expresar -Conectar -Cuestionar
“Defining Computational Thinking for Science, Technology, Engineering, and Math” [108] por David Weintrop et Al.	-Datos e información -Modelado y simulación -Solución de problemas computacionales -Pensamiento sistémico

- **Caracterización de elementos de CT considerados por autores que desarrollaron herramientas de evaluación CT**

Los autores que desarrollaron las herramientas consideradas tuvieron en cuenta los siguientes elementos relacionados en la Tabla 5, presentada a continuación:

Tabla 5. Caracterización de elementos de CT considerados por autores que desarrollaron herramientas de evaluación CT

Referencia	Elementos CT
Dr. Scratch [109]	<ul style="list-style-type: none"> -Control de flujo -Representación de la información -Abstracción -Interactividad del usuario -Sincronización -Paralelismo -Lógica
Ninja Code [90]	<ul style="list-style-type: none"> -Declaraciones condicionales -Ciclos -Procedimiento -Procedimiento común -Información -Eventos -Paralelismo -Interfaz de usuario
SAT [91]	<ul style="list-style-type: none"> -Abstracción y descomposición de problemas -Paralelismo -Pensamiento lógico -Sincronización - Control de flujo -Interactividad del usuario -Representación de información
Bebras [110], [111]	<ul style="list-style-type: none"> -Descomposición -Diseño de algoritmos -Reconocimiento de patrones -Abstracción y generalización de patrones
CTArcade [112]	<ul style="list-style-type: none"> -Pensamiento algorítmico -Descomposición -Reconocimiento de patrones -Generalización de patrones y abstracción - Instancias no articuladas
codOrama [84]	<ul style="list-style-type: none"> -Seguridad -Efectividad -Abstracción -Interactividad con el usuario -Representación de la información -Sincronización

3. Caracterización de criterios mínimos y mecanismos de evaluación

- Caracterización de elementos de CT considerados por autores que realizaron implementaciones en el aula de clase

Algunos autores que han tenido experiencias en el aula de clase, donde incorporan dentro de sus objetivos brindar herramientas a los estudiantes para fomentar el desarrollo de CT, han definido los elementos empleados en sus pruebas en el aula. En la Tabla 6 se especifican los estudios incorporados en esta caracterización y los elementos determinados en estos.

Tabla 6. Caracterización de elementos de CT considerados por autores que realizaron implementaciones en el aula de clase

Referencia	Elementos CT
Assessing elementary students' computational thinking in everyday reasoning and robotics programming [73]	-Formulación y solución de problemas -Información -Representación y solución de problemas -Generar, revisar y evaluar soluciones
How to support students' computational thinking skills in educational robotics activities [113]	-Abstracción -Generalización -Algoritmos -Modularidad -Descomposición
Learning programming at the computational thinking level via digital game-play [114]	-Resolución de problemas -Construcción de algoritmos -Depuración -Simulación -Socializar
A serious game for developing computational thinking and learning introductory computer programming [115]	-Lógica condicional -Construcción de algoritmos -Depuración -Simulación -Computación distribuida
Relationship between computational thinking and a measure of intelligence as a general problem-solving ability [116]	-Descomposición -Abstracción y generalización de patrones -Organizar y analizar información -Diseño algorítmico
Educational game design as gateway for operationalizing computational thinking skills among middle school students [117]	-Descomposición -Reconocimiento de patrones

	<ul style="list-style-type: none"> -Generalización de patrones y abstracción -Diseño de algoritmos -Análisis y visualización de información
STEM outreach: Assessing computational thinking and problem solving [118]	<ul style="list-style-type: none"> -Pensamiento algorítmico -Abstracción -Descomposición de problemas -Información -Paralelismo -Control de flujo -Incremental e iterativo -Pruebas y depuración -Cuestionar
Assessing implicit computational thinking in zombinis gameplay [119]	<ul style="list-style-type: none"> -Descomposición de problemas -Reconocimiento de patrones -Abstracción -Diseño de algoritmos -Pruebas sistemáticas -Implementación de una solución
Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability [120]	<ul style="list-style-type: none"> -Razonamiento [lógico] -Abstracción -Descomposición -Generalización
Assessing young children's computational thinking abilities [121]	<ul style="list-style-type: none"> -Proceso de diseño -Hardware -Software -Estructura de control -Representación de la información -Algoritmos -Depuración
Task-based assessment of students' computational thinking skills developed through visual programming or tangible coding environments [122]	<ul style="list-style-type: none"> -Abstracción -Pensamiento algorítmico -Reconocimiento de patrones -Descomposición
A principled approach to designing assessments that integrate science and computational thinking evidence-centered design [123]	<ul style="list-style-type: none"> -Algoritmos -Inicialización y actualización de variables -Operadores y expresiones -Estructuras de control -Recolección de información y visualización de gráficas -Capacidad de hacer inferencias y predicciones -Modelado

3. Caracterización de criterios mínimos y mecanismos de evaluación

	-Evaluación y depuración
Assessing computational thinking process using a multiple evaluation approach [124]	Conceptos computacionales: -Secuencias -Bucles -Eventos -Paralelismo -Condicionales -Operadores -Variables -Abstracción Prácticas meta cognitivas: -Planeación -Evaluación y monitoreo Comportamientos de aprendizaje: -Reflexionar (resiliencia, inventiva, reflexividad y reciprocidad)

3.3.2. Criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación

Con el propósito de conocer cómo estimar el desarrollo de CT se han examinado algunos trabajos, estos se han dividido en tres grupos de acuerdo a lo que sus autores plantean: marcos de referencia para guiar el proceso, diseño de herramientas que facilitan la ejecución de la evaluación e implementación de evaluaciones en el aula de clase. Para los trabajos analizados se consigna el nombre del documento y su referencia, una breve descripción, los elementos de CT, los criterios mínimos y mecanismos considerados y expuestos en el documento.

- **Caracterización de elementos de CT, criterios y mecanismos de evaluación considerados por autores que realizaron marcos de referencia para la evaluación**

Los marcos de referencia propuestos guían el proceso de evaluación de CT, se tuvieron en consideración aquellos que especifican los criterios y presentan la manera como se pueden evaluar. Para cada referencia se detallan su nombre, descripción, criterios mínimos relacionados con los elementos y mecanismos de evaluación. Los resultados pueden ser vistos en las Tablas 7, 8 y 9.

Tabla 7. Referencia 1. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

<p>“New frameworks for studying and assessing the development of computational thinking” [82]</p>	
<p>Descripción: Los autores definen un marco de referencia para el estudio y evaluación de CT, en el que consideran los elementos que componen CT y recomiendan aproximaciones para su evaluación.</p>	
<p>Elementos: Conceptos: Secuencias Bucles Paralelismo Eventos</p> <p>Condicionales Operadores</p> <p>Información</p> <p>Prácticas: Incremental e iterativo Pruebas y depuración</p> <p>Reutilización y remezcla Abstracción y modularización</p> <p>Perspectivas: Expresar</p> <p>Conectarse</p> <p>Cuestionarse</p>	<p>Criterios mínimos:</p> <p>Representa las soluciones como una serie de pasos Repite pasos mediante el uso de bloques Usa secuencias ejecutándose al mismo tiempo Muestra como una tarea se efectúa como una consecuencia de un acontecimiento Ejecuta acciones como consecuencia de un acontecimiento Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres Almacenamiento, recuperación y actualización de la información</p> <p>Construye la solución por pequeños pasos Prueba y reconoce errores en las secuencias de programas y es capaz de arreglarlos Emplea código existente en un nuevo proyecto Puede construir algo grande y complejo, partiendo de pequeñas partes</p> <p>Comunica a sus compañeros o docente información de lo que está desarrollando y como lo hace Comparte con otros desarrolladores y buscar entre los diferentes proyectos Pregunta por lo que lo rodea y en algunos casos responder a través del diseño</p>
<p>Mecanismos: Analizar los proyectos, es el registro de los bloques de los proyectos y su frecuencia de uso. Entrevistar a los estudiantes a través de preguntas basadas en artefactos-objetos, donde los estudiantes responden diferentes preguntas acerca de su motivación, evolución y resultado del proceso de desarrollo. Preguntar acerca del diseño de escenarios, los evaluadores piden explicaciones de qué es lo que hace el código del proyecto, cómo el proyecto podría ser</p>	

3. Caracterización de criterios mínimos y mecanismos de evaluación

ampliado, si consideraron el control de errores y la depuración y si el proyecto fue remezclado.
--

Tabla 8. Referencia 2. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Evidence of assessing computational thinking” [107]	
Descripción: Los autores tuvieron en cuenta los requisitos de acuerdo al currículo y las definiciones de CT para sugerir un mecanismo para la evaluación de los objetivos y el progreso en la computación y desarrollo del pensamiento computacional.	
Elementos: Abstracción Descomposición Pensamiento algorítmico Generalización Evaluación	Criterios mínimos: Reconoce que existen diferentes soluciones para el mismo problema Sabe que se puede utilizar un procedimiento para ocultar los detalles con una sub-solución Emplea correctamente las declaraciones condicionales Utiliza una variable y operadores relacionales dentro de un bucle para manejarlo Realiza búsquedas de información Diseña soluciones descomponiendo problemas y soluciona cada parte Diseña, escribe y depura programas modulares utilizando procedimientos Analiza y evalúa la información Hace juicios sobre el contenido digital cuando lo evalúa y lo reutiliza Reconoce a la audiencia al diseñar y crear contenido digital Comprende el potencial de la tecnología de la información para la colaboración Utiliza criterios para evaluar la calidad de las soluciones
Mecanismos: Revisar el cumplimiento de cada criterio de acuerdo a tareas realizadas por sus estudiantes.	

Tabla 9. Referencia 3. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“An Exploration of Three-Dimensional Integrated Assessment (TDIA) for Computational Thinking” [125]

Descripción: El TDIA tiene el objetivo de integrar la direccionalidad, apertura y proceso en el diseño de tareas de evaluación efectivas y evaluar las tres dimensiones de CT. Proponen un marco de referencia que facilita la creación de tareas para evaluar los elementos de CT.	
Elementos: Los conceptos Las prácticas Las perspectivas [82]	Criterios mínimos: Mencionan que definen los elementos específicos que se deberían cumplir para cada tarea, abarcando todos los elementos en el desarrollo de las 6 tareas.
Mecanismos: Revisar el desarrollo de tareas diseñadas de acuerdo al marco conceptual definido. Inspeccionar el informe de reflexión. Examinar el informe de diseño creativo. Evaluación sumativa: evaluación de proyectos.	

- Caracterización de elementos de CT, criterios y mecanismos de evaluación considerados por autores que diseñaron herramientas

Las herramientas que se incluyeron en este estudio, fueron las diseñadas con el objetivo de facilitar el proceso de evaluación de los elementos de CT. Se consideraron seis (6) herramientas que definían los criterios mínimos y describían sus mecanismos de evaluación. Las Tablas de la 10 a la 15 exponen los detalles (descripción, criterios mínimos en relación con los elementos y mecanismos).

Tabla 10. Referencia 4. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

"Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking" [15]	
Descripción: Esta herramienta se encuentra disponible en la web de manera gratuita, facilita el análisis de proyectos Scratch teniendo en cuenta los elementos definidos en tres niveles de complejidad. El trabajo realizado en [91] considera los mismo elementos, criterios y mecanismos en su evaluación.	
Elementos:	Criterios mínimos:

3. Caracterización de criterios mínimos y mecanismos de evaluación

Abstracción y descomposición del problema	Emplea más de un hilo y más de un sprite, define bloques, usa clones
Paralelismo	Dispone de dos o más hilos inician con el mismo bloque
Pensamiento lógico	Hace uso de condicionales y operadores lógicos
Sincronización	Utiliza bloques como esperar y parar, maneja mensajes correctamente
Control de flujo	Emplea bloques como repetir, por siempre y otros bucles.
Interactividad del usuario	Brinda herramientas al usuario para interactuar con el sistema, a través del teclado y ratón.
Representación de los datos	Modifica las propiedades de los sprites, usa operaciones con variables o listas
<p>Mecanismos:</p> <p>Analizar automáticamente del código Scratch.</p> <p>Entregar realimentación a los estudiantes, de acuerdo a los niveles de experticia planteados: básico, medio y alto, también brinda algunas recomendaciones para mejorar el código evaluado.</p>	

Tabla 11. Referencia 5. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

"Real time assessment of computational thinking" [16]	
Descripción: Esta herramienta facilita a los docentes ver de manera rápida el nivel CT de sus estudiantes de acuerdo al marco establecido en [83].	
Elementos: Computational Thinking Patterns (CTP) definidos en el marco [83].	Criterios mínimos: Cumple con los CTP definidos para cada ejercicio.
<p>Mecanismos:</p> <p>Analizar automáticamente y en tiempo real de los ejercicios entregados por los estudiantes acerca de sus proyectos de acuerdo a los patrones definidos en su marco de referencia propio, brindando al docente un mejor entendimiento del grado de experticia de sus estudiantes.</p>	

Tabla 12. Referencia 6. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“A Design and Evaluation Framework for Visual Programming Codes” [84]	
Descripción: codOrama es una herramienta visual que facilita el análisis de proyectos realizados en Scratch. Emplea una tabla para simplificar el escenario de análisis; organizando los bloques empleados de acuerdo a los elementos CT definidos.	
Elementos: Elementos de funcionamiento: Seguridad Efectividad Abstracción Interactividad con el usuario Representación de la información Sincronización Elementos de mantenimiento: Paralelismo Legibilidad Optimización Adaptación al cambio Facilidad de pruebas	Criterios mínimos: Inicializa las variables, termina la interacción después de un tiempo Cumple con lo requerido para que el programa opera apropiadamente Descompone el problema en partes pequeñas Realiza una adecuada interacción con el usuario Emplea correctamente las variables, emplea listas Emplea una comunicación entre los objetos correcta, usa comandos disponibles para ayudar al flujo del programa Reconoce la diferencia entre procesos paralelos y seriales Programa de tal manera que el código se entiende con facilidad. Considero cambiar los nombres de los objetos con nombres significativos. Dispone de los recursos correctamente, por ejemplo no hay redundancia o no hay presencia de código muerto. Facilita la realización cambios. Permite realizar corrección de los errores.
Mecanismos: Analizar del cuadro entregado por codOrama, que relaciona la solución con los elementos definidos	

3. Caracterización de criterios mínimos y mecanismos de evaluación

Tabla 13. Referencia 7. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Ninja Code Village for Scratch: Function Samples/Function Analyser and Automatic Assessment of Computational Thinking Concepts” [90]	
Descripción: La página web Ninja Code permite el análisis automático de proyectos Scratch de manera gratuita.	
Elementos: Condicionales Bucles Procedimientos Procedimientos comunes Información Eventos Paralelismo Interfaz con el usuario	Criterios mínimos: Usa condicionales y operadores lógicos Emplea secuencias de repetición Codifica más de un script, define bloques, usa clones Define scripts que comparten bloques, recursividad del uso de bloques Define variables o listas Determina eventos a través del uso del teclado, ratón y recursos multimedia. Maneja correctamente los mensajes, sincroniza el fondo Permite la interacción con el usuario a través del uso del teclado, ratón y recursos multimedia
Mecanismos: Analizar automáticamente los proyectos entregados por los estudiantes De acuerdo a 4 niveles de experticia planteados por los autores, considera 3 tipos de proyectos para realizar su clasificación: animaciones, juegos y música.	

Tabla 14. Referencia 8. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Interactive assessment tools for computational thinking in high school STEM classrooms” [126] [93], [108]	
Descripción: La evaluación propuesta se compone de dos herramientas interactivas que facilitan al estudiante analizar la información con visualizaciones dinámicas y explorar conceptos con modelos computacionales bajo un enfoque STEM.	
Elementos:	Criterios mínimos:

<p>Habilidades:</p> <p>Análisis de información</p> <p>Modelado y simulación</p> <p>Solución de problemas computacionales</p> <p>Pensamiento sistémico</p>	<p>Toma la información para manipularla, analizarla y visualizarla</p> <p>Emplea modelos computacionales para entender conceptos y/o solucionar problemas, entiende el funcionamiento de los modelos, construye modelos o los modifica</p> <p>Selecciona herramientas computacionales efectivas, evalúa diferentes enfoques de un problema, brinda soluciones informáticas modulares a los problemas, emplea estrategias de resolución de problemas y creación de abstracciones</p> <p>Comprende las relaciones entre los sistemas, identifica, entiende y maneja la complejidad de los problemas, investiga un sistema como una unidad</p>
<p>Mecanismos:</p> <p>Analizar automáticamente las actividades realizadas a través de herramientas computacionales que permiten a los estudiantes explorar y aprender conceptos dinámicamente, los autores han definido un marco de referencia donde los estudiantes a través de la interacción con dichas herramientas son evaluados.</p>	

Tabla 15. Referencia 9. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

<p>“Smart assessment of and tutoring for computational thinking MOOC assignments using MindReader”[127]</p>	
<p>Descripción: MindReader facilita la evaluación de las tareas de los estudiantes y los guía para que mejoren sus desarrollos sin ayuda humana y en tiempo real.</p>	
<p>Elementos:</p> <p>No definidos en el documento</p>	<p>Criterios mínimos:</p> <p>No definidos en el documento</p>
<p>Mecanismos:</p> <p>Analizar automáticamente el código escrito por los estudiantes, la evaluación se realiza por medio del prototipo online MindReader, este sistema le brinda a los</p>	

3. Caracterización de criterios mínimos y mecanismos de evaluación

estudiantes de manera masiva la realimentación acerca de su desarrollo a manera de guía y automáticamente.
--

- **Caracterización de elementos de CT, criterios y mecanismos de evaluación considerados por autores que realizaron implementaciones en el aula**

Diversos autores han implementado evaluaciones de CT en el aula de clase, a continuación, se describen algunas, que fueron consideradas las más relevantes para este estudio, en las Tablas de la 16 a la 24 se enumeran los criterios mínimos definidos en relación con los elementos CT y mecanismos de evaluación.

Tabla 16. Referencia 10. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

"Assessing elementary students' computational thinking in everyday reasoning and robotics programming" [73]	
Descripción: Considerando los estándares asociados con CT, los autores elaboraron un instrumento para evaluar CT en los estudiantes, el instrumento esta contextualizado en dos tipos de aplicaciones de CT que son codificación con robots y razonamiento de los eventos cotidianos.	
Elementos: Formulación y solución de problemas utilizando una sintaxis comprensible para una maquina Información Representación y solución de problemas Generar, revisar y evaluar soluciones	Criterios mínimos: Formula problemas de forma que el computador puede ayudar a resolverlos Procesa información de una manera lógica, representa la información de manera abstracta Resuelve problemas de una manera eficiente Transfiere conocimiento en la solución de otros problemas

Mecanismos:

Calificar un instrumento de 23 ítems donde 15 son de respuesta múltiple, calificadas como correcto o incorrecto, y 8 con respuestas abiertas, verificadas por un grupo de personas que diseñaron rubricas donde se identificaron elementos CT para cada pregunta.

Los 23 ítems son referentes a la programación con robots (lenguajes de programación basados en texto y visuales) y al razonamiento de eventos cotidianos.

Tabla 17. Referencia 11. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“STEM outreach: Assessing computational thinking and problem solving” [118]	
<p>Descripción: El trabajo describe el proceso de cuatro semanas de trabajo con estudiantes, donde se presentó un programa STEM centrado en la programación de computadores y micro controladores, para la evaluación los autores sugieren un instrumento de autoeficacia para determinar el mejoramiento de las habilidades de resolución de problemas y CT.</p>	
<p>Elementos:</p> <p>Pensamiento algorítmico</p> <p>Abstracción</p> <p>Descomposición de problemas</p> <p>Información</p> <p>Paralelismo</p> <p>Control de flujo</p> <p>Incremental e iterativo</p> <p>Pruebas y depuración</p> <p>Reutilización, conexión y cuestionamiento</p>	<p>Criterios mínimos:</p> <p>Crea una lista de pasos para resolver los problemas, usa las matemáticas</p> <p>Intenta simplificar un problema ignorando detalles innecesarios, busca patrones</p> <p>Fracciona un problema en varias partes y resuelve una a una</p> <p>Puede guardar, actualizar y recuperar información</p> <p>Trabaja con otros para resolver diferentes partes del problema al mismo tiempo, realiza más de una acción al mismo tiempo</p> <p>Emplea bucles correctamente, responde a eventos, los comandos se ejecutan al cumplir con las condiciones esperadas</p> <p>Realiza mejoras paulatinas ampliando la solución</p> <p>Corre el programa para verificar que la solución sea correcta y soluciona los errores</p>

3. Caracterización de criterios mínimos y mecanismos de evaluación

	Comparte sus programas y revisa los de sus compañeros para mejorar el propio
<p>Mecanismos:</p> <p>Valorar una encuesta de autoeficacia, aplicada al inicio y final de la instrucción para realizar una comparación (evaluación sumativa).</p>	

Tabla 18. Referencia 12. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Assessing implicit computational thinking in zoombinis gameplay” [119]	
<p>Descripción: Zoombinis es un juego de aprendizaje que se enfoca en CT. El TDIA tiene el objetivo de integrar la direccionalidad, apertura y proceso en el diseño de tareas de evaluación efectivas y evaluar las tres dimensiones de CT. Proponen un marco de referencia que facilita la creación de tareas para evaluar los elementos de CT.</p>	
<p>Elementos:</p> <p>Descomposición de problemas</p> <p>Reconocimiento de patrones</p> <p>Abstracción</p> <p>Diseño de algoritmos</p> <p>Pruebas sistemáticas</p> <p>Implementación de una solución</p>	<p>Criterios mínimos:</p> <p>Aísla los atributos</p> <p>Observa los elementos comunes y los reconoce</p> <p>Es capaz de generalizar por atributos</p> <p>Puede generar secuencias de comportamientos que se repiten</p> <p>Prueba un atributo al tiempo</p> <p>Ha completado la solución</p>
<p>Mecanismos:</p> <p>Verificar un video grabación de los estudiantes mientras ellos desarrollan las tareas diseñadas.</p> <p>Estimar el grado de cumplimiento en el desarrollo de los retos propuestos en el juego.</p>	

Tabla 19. Referencia 13. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Assessing young children’s computational thinking abilities” [121]

<p>Descripción: En la tesis describen los fundamentos conceptuales para la creación del instrumento de evaluación, introducen la plataforma robótica KIBO y definen la evaluación mediante TACTIC.</p>	
<p>Elementos:</p> <p>Proceso de diseño</p> <p>Hardware</p> <p>Software</p> <p>Estructura de control</p> <p>Representación de la información</p> <p>Algoritmos</p> <p>Depuración</p>	<p>Criterios mínimos:</p> <p>Evidencia un proceso iterativo</p> <p>Hay conocimiento de las partes que componen el robot y sus funciones.</p> <p>Programa secuencias sintácticamente correctas.</p> <p>Integra parámetros y bloques que controlan comportamientos complejos con realimentación de sensores.</p> <p>Identifica los símbolos de los bloques y cartas de acción.</p> <p>Descompone un problema en partes más pequeñas.</p> <p>Utiliza grupos de bloques como unidades en programas más grandes.</p> <p>Reconoce errores en las secuencias de programas y es capaz de arreglarlos.</p>
<p>Mecanismos:</p> <p>Indagar los resultados del desarrollo de actividades interactuando con KIBO, para las que TACTIC determina algunas preguntas en relación con los criterios de evaluación de acuerdo a 4 niveles de experticia en la programación: proto, novatos, desarrollador y experto.</p> <p>Verificar un video grabación de los estudiantes interactuando con KIBO.</p>	

Tabla 20. Referencia 14. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

<p>“Task-based assessment of students’ computational thinking skills developed through visual programming or tangible coding environments” [122]</p>
<p>Descripción: Buscando encontrar la relación entre las tareas de resolución de problemas en un computador, los ambientes basados en robótica y su relación con CT, los autores validaron las tareas de sus estudiantes basadas en el concurso internacional Bebras [111], realizaron un pre test con un conjunto de</p>

3. Caracterización de criterios mínimos y mecanismos de evaluación

tareas y un post test con las tareas que encontraron dentro del conjunto y que son de mayor ayuda para estimar el desarrollo de CT.	
Elementos: Abstracción Pensamiento algorítmico Reconocimiento de patrones Descomposición	Criterios mínimos: Cada tarea tiene asociados los elementos específicos que se deberían cumplir.
Mecanismos: Valorar el resultado de las tareas propuestas. (Evaluación sumativa) Las tareas de Bebras corresponden a aproximaciones de soluciones de problemas de la vida real, cuya respuesta es de selección múltiple, lo que corresponde a una evaluación sumativa. La realimentación es inmediata, aunque se limita a informarle al estudiante si pudo realizar la tarea o no.	

Tabla 21. Referencia 15. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

"A principled approach to designing assessments that integrate science and computational thinking evidence-centered design" [123]	
Descripción: Este trabajo describe los principios del proceso de desarrollo de la evaluación y rúbricas en relación a los conceptos y prácticas a través de las ciencias, CT, y el modelado computacional. Proponen emplear tales evaluaciones de manera formativa e integrada en un enfoque STEM, en contextos de aprendizaje informático.	
Elementos: Algoritmos Inicialización y actualización de variables Operadores y expresiones Estructuras de control Recolección de información y visualización de gráficas	Criterios mínimos: No definidos en el documento

Capacidad de hacer inferencias y predicciones Modelado Evaluación y depuración	
<p>Mecanismos:</p> <p>Analizar automáticamente las habilidades de sus estudiantes mediante el desarrollo de las tareas.</p> <p>La evaluación automatizada que brinda a los estudiantes una guía en tiempo real y se ofrece a los docentes la información acerca de las habilidades de sus estudiantes.</p> <p>Verificar las prácticas de programación mediante una grabación de video de los estudiantes programando.</p>	

Tabla 22. Referencia 16. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

"Assessing computational thinking in students' game designs" [80]	
Descripción: Los autores discuten los resultados de CT obtenidos tras el análisis del proceso de diseño de juegos en Scratch empleando dos métodos: análisis automático con Dr. Scratch [15] y análisis cualitativo de las elecciones de diseño.	
<p>Elementos:</p> <p>Los planteados por Dr. Scratch</p> <p>Interacciones del jugador</p> <p>Tema de mitigación</p> <p>Meta juego</p>	<p>Criterios mínimos:</p> <p>Los planteados por Dr. Scratch</p> <p>Facilita la interacción con el jugador</p> <p>Aborda correctamente el tema</p> <p>La narrativa, las instrucciones y demás elementos del juego son acordes a lo esperado</p>
<p>Mecanismos:</p> <p>Analizar automáticamente el código de los estudiantes empleando Dr. Scratch</p> <p>Analizar cualitativamente las interacciones del jugador, el tema de mitigación y meta juego</p>	

Tabla 23. Referencia 17. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

3. Caracterización de criterios mínimos y mecanismos de evaluación

“Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability” [120]	
Descripción: En este estudio describen el desarrollo de una herramienta de medición de autoeficacia que puede predecir la habilidad de pensamiento computacional, los autores definen el grupo de elementos a evaluar, estos fueron seleccionados por un grupo de 11 expertos y finalmente seleccionaron 18 ítems.	
Elementos: Razonamiento Abstracción Descomposición Generalización	Criterios mínimos: Decide si los datos que se utilizarán para la solución del problema son adecuados o no Hace comentarios sobre los datos utilizados para la solución del problema. Si hay problemas secundarios en el problema, puede administrar los procesos de solución de estos Realiza conexiones entre el problema actual y los problemas encontrados anteriormente
Mecanismos: Valorar las respuestas de los estudiantes de la herramienta de medición de autoeficacia. Escala compuesta de 4 factores y 18 elementos para medir la autoeficacia del pensamiento computacional de los estudiantes de secundaria.	

Tabla 24. Referencia 18. Criterios mínimos para el desarrollo de CT y sus mecanismos de evaluación

“Assessing computational thinking process using a multiple evaluation approach” [124]	
Descripción: El autor propone varias aproximaciones para realizar evaluación de CT en el aula de clase, realizando una medición de los conceptos de CT y el comportamiento de los estudiantes cuando programan en Alice y Scratch.	
Elementos: Conceptos computacionales (secuencias, eventos, bucles, paralelismo,	Criterios mínimos: Emplea los bloques correspondientes a cada concepto

condicionales, operadores, variables y abstracción) Prácticas meta cognitivas (planeación, evaluación y monitoreo) Comportamientos de aprendizaje (resiliencia, inventiva, reflexividad y reciprocidad)	Planea con anterioridad lo que va a realizar e identifica errores y explica como los resolvió Trabaja con colaboración, piensa creativamente y críticamente, depura errores, modifica ideas y comunica estas ideas utilizando diferentes modos de intercambio conversacional
<p>Mecanismos:</p> <p>Observar a los participantes y tener conversaciones informales con ellos. Examinando el lenguaje oral y gestual que los estudiantes usaban para sus explicaciones y discusiones grupales, preguntando acerca del contexto de sus relaciones con el maestro, los compañeros y la tecnología en su aula.</p> <p>Entrevistar a los estudiantes para conocer sus prácticas y perspectivas. Mediante entrevistas semi estructuradas. Realizadas durante 10-15 minutos, usadas para desplegar el yo profundo del estudiante. Análisis de las transcripciones de las entrevistas.</p> <p>Revisar las bitácoras y hojas de resolución de problemas. Para que lleven un seguimiento de su trabajo y los planes para la solución de los problemas propuestos.</p> <p>Analizar los juegos terminados. Teniendo en cuenta la guía que propone el autor, donde define algunos elementos (conceptos) y da ejemplos de cómo se definir si un estudiante empleo o no correctamente cada concepto.</p>	

3.4. Fase de Interpretación por Núcleos Temáticos

A continuación, por parte de cada núcleo temático se cumple con la abstracción de elementos a considerar dentro del desarrollo CT, de la misma forma se consideran los criterios mínimos para estimar el desarrollo del conjunto de habilidades y sus mecanismos de evaluación.

3. Caracterización de criterios mínimos y mecanismos de evaluación

3.4.1. Mapeo de términos de los elementos de CT

Como se pudo observar en la fase anterior, no hay consenso de los elementos de CT que se deben considerar para estimar su desarrollo, lo que pone a la vista que tampoco hay homogeneidad en las definiciones. Eliminando las ambigüedades y redundancias más evidentes encontradas en la lista completa de elementos de CT (Anexo A), se obtuvieron en total 73 elementos.

Por esto, se hace necesario cumplir con el proceso de mapeo y depuración de términos, en donde primero se seleccionan los elementos mayormente considerados en los trabajos analizados de los núcleos temáticos, y posteriormente, se relacionan las definiciones que corresponden a los elementos de CT que algunos autores han tomado en consideración, ya que no todos los definen.

El mapeo de elementos con respecto a los trabajos estudiados tiene como núcleos temáticos: (1) autores conocedores del área; (2) asociaciones educativas y universidades; (3) compañías dentro de la industria informática y (4) aproximaciones como marcos de referencia y modelos; (5) herramientas diseñadas para la evaluación de CT o para evaluar prácticas de programación y (6) propuestas implementadas en el aula de clase.

Se determinó para todos los núcleos temáticos, que el criterio de selección para el mapeo de los elementos de CT fuera superar el 30% del porcentaje de uso, de tal manera que todos los elementos que superen este porcentaje serán tenidos en consideración y señalados en las tablas.

Para llegar a sus conclusiones, los autores referidos dentro de esta caracterización realizaron revisiones literarias en las que tuvieron en cuenta diferentes trabajos relacionados con el desarrollo de CT. A pesar de que uniendo los resultados de los cuatro (4) autores el recuento de todos los elementos CT tenidos en cuenta es treinta (30), únicamente los ocho (8) elementos que se encuentran consignados en la Tabla 25 cumplieron con el criterio de selección, en este caso los seleccionados tienen un patrón de uso equivalente o superior a dos (2). Se puede notar que los autores por unanimidad proponen que los algoritmos y la depuración son elementos que se deben considerar.

Tabla 25. Mapeo de terminos a nivel de autores

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	3	75
Generalización de patrones	3	75
Modularización, Modularidad	2	50
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	4	100
Formulación o representación de problemas	2	50
Descomposición, Descomposición de problemas	3	75
Declaraciones condicionales, Lógica condicional	2	50
Depuración	4	100

El mapeo de términos a nivel de asociaciones educativas, arrojó información de 10 elementos que cumplieron con el criterio de selección igual o mayor a 2, como se evidencia en la Tabla 26. El recuento total de elementos fue de 34, de los que se descartaron: reconocimiento y generalización de patrones, los elementos referentes a la gestión de la información, paralelismo, entre otros. En esta selección, no se encontraron elementos comunes que hayan sido propuestos unánimemente.

Dentro de los elementos encontrados en este grupo, se puede apreciar que varios de los elementos están relacionados con prácticas que van más allá de los saberes de los estudiantes.

Tabla 26. Mapeo de terminos a nivel de asociaciones educativas

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	4	80

3. Caracterización de criterios mínimos y mecanismos de evaluación

Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	4	80
Descomposición, Descomposición de problemas	3	60
Evaluar	2	40
Depuración	2	40
Experimentar	2	40
Crear	2	40
Colaborar	3	60
Analizar	2	40
Comunicar	2	40

Las dos propuestas de las compañías de la industria informática coinciden en que la abstracción, el reconocimiento de patrones, la descomposición y los algoritmos son elementos de CT. Los otros ocho (8) elementos contenidos en la tabla 27 fueron propuestos únicamente por Google, a pesar de esto, cumplen con el criterio de selección.

Tabla 27. Mapeo de terminos a nivel de compañías de la industria informática

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	2	100
Generalización de patrones	1	50
Reconocimiento de patrones	2	100
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	2	100
Simulación	1	50
Descomposición, Descomposición de problemas	1	50
Paralelismo	1	50
Recolección de información	1	50
Análisis de información	1	50
Representación de información	1	50
Automatización	1	50

La tabla 28 muestra una lista de 26 elementos, resultado del mapeo a nivel de marcos de referencia, donde se tomaron 3. Al igual que en el caso anterior, el criterio de selección es superado con una sola ocurrencia. La mayoría de los elementos tiene un patrón de uso igual a 1, lo que demuestra disparidad en la selección de elementos. Por estas razones esta es la lista más extensa. Además, se puede apreciar que no hay unanimidad en ninguno de los elementos propuestos.

Tabla 28. Mapeo de terminos a nivel de marcos de referencia

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	2	66,67
Generalización de patrones	1	33,33
Modularización, Modularidad	1	33,33
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	1	33,33
Modelado	1	33,33
Simulación	1	33,33
Descomposición, Descomposición de problemas	1	33,33
Solución y resolución de problemas	1	33,33
Paralelismo	1	33,33
Recolección de información	2	66,67
Análisis de información	2	66,67
Representación de información	2	66,67
Evaluar	1	33,33
Probar	1	33,33
Depuración	1	33,33
Secuencias	1	33,33
Bucles o ciclos	1	33,33
Eventos	1	33,33
Condicionales	1	33,33
Operadores	1	33,33
Incremental e iterativo, Iteración	1	33,33

3. Caracterización de criterios mínimos y mecanismos de evaluación

Reutilización y remezclado	1	33,33
Expresar	1	33,33
Conectar	1	33,33
Cuestionar	1	33,33
Pensamiento sistémico	1	33,33

De los 20 elementos resultantes del recuento teniendo en cuenta las propuestas de las 6 herramientas en este núcleo, quedan 12 elementos que superaron el criterio de selección tabla 29.

Tabla 29. Mapeo de terminos a nivel de herramientas de evaluación

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	5	83,33
Generalización de patrones	2	33,33
Reconocimiento de patrones	2	33,33
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	2	33,33
Descomposición, Descomposición de problemas	3	50
Declaraciones condicionales, Lógica condicional	2	33,33
Razonamiento lógico, Lógica, Pensamiento lógico	3	50
Paralelismo	3	50
Representación de información	4	66,67
Control de flujo	2	33,33
Interactividad del usuario, Interfaz de usuario	4	66,67
Sincronización	3	50

Como se tuvieron en cuenta 13 trabajos relacionados con prácticas de programación y evaluación en el aula de clase; el patrón de uso en este caso es de 4. En este caso

tampoco hay elementos seleccionados por unanimidad, pero se encuentran porcentajes de uso altos para abstracción y algoritmos.

Tabla 30. Mapeo de terminos a nivel de implementaciones en el aula de clase

Elemento CT	Patrón de uso	Porcentaje de uso
Abstracción	8	61,53
Generalización de patrones	4	30,76
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	10	76,92
Descomposición, Descomposición de problemas	7	53,84
Recolección de información	4	30,76
Análisis de información	4	30,76
Depuración	6	46,15

3.4.2. Mapeo de los criterios mínimos para estimar el desarrollo de CT

Para realizar este mapeo se tuvieron en cuenta los criterios mínimos obtenidos en la fase de descripción y los elementos resultantes en las tablas 28, 29 y 30 del mapeo de términos de los elementos de CT. El proceso de mapeo se realizó considerando los siguientes los núcleos temáticos: (1) aproximaciones como marcos de referencia y modelos; (2) herramientas diseñadas para la evaluación de CT o para evaluar prácticas de programación y (3) propuestas implementadas en el aula de clase.

Las tablas 31, 32 y 33 muestran la lista de los criterios mínimos en relación a los elementos que involucran. Primero se pueden ver los criterios que han sido considerados en los marcos de referencia estudiados, dos de los marcos coinciden en todos sus criterios (referencia 1 y 3). La referencia 2 incluye elementos importantes que no consideran los otros como lo son descomposición, generalización y evaluación.

Tabla 31. Mapeo de criterios mínimos encontrados en Marcos de referencia

3. Caracterización de criterios mínimos y mecanismos de evaluación

Criterios	Elementos relacionados	Referencia
Reconoce que existen diferentes soluciones para el mismo problema	Abstracción	Referencia 2
Utiliza una variable y operadores relacionales dentro de un bucle para manejarlo	Abstracción	Referencia 2
Sabe que se puede utilizar un procedimiento para ocultar los detalles con una sub-solución	Abstracción	Referencia 2
Emplea correctamente las declaraciones condicionales	Abstracción	Referencia 2
Puede construir algo grande y complejo, partiendo de pequeñas partes	Abstracción y modularización	Referencia 1 y 3
Repite pasos mediante el uso de bloques	Bucles	Referencia 1 y 3
Ejecuta acciones como consecuencia de un acontecimiento	Condicionales	Referencia 1 y 3
Comparte con otros desarrolladores y buscar entre los diferentes proyectos	Conectarse	Referencia 1 y 3
Pregunta por lo que lo rodea y en algunos casos responder a través del diseño	Cuestionarse	Referencia 1 y 3
Diseña soluciones descomponiendo problemas y soluciona cada parte	Descomposición	Referencia 2
Realiza búsquedas de información	Descomposición	Referencia 2
Analiza y evalúa la información	Evaluación	Referencia 2
Utiliza criterios para evaluar la calidad de las soluciones	Evaluación	Referencia 2
Hace juicios sobre el contenido digital cuando lo evalúa y lo reutiliza	Evaluación	Referencia 2
Reconoce a la audiencia al diseñar y crear contenido digital	Evaluación	Referencia 2
Comprende el potencial de la tecnología de la información para la colaboración	Evaluación	Referencia 2

Muestra como una tarea se efectúa como una consecuencia de un acontecimiento	Eventos	Referencia 1 y 3
Comunica a sus compañeros o docente información de lo que está desarrollando y como lo hace	Expresar	Referencia 1 y 3
Diseña, escribe y depura programas modulares utilizando procedimientos	Generalización	Referencia 2
Construye la solución por pequeños pasos	Incremental e iterativo	Referencia 1 y 3
Almacenamiento, recuperación y actualización de la información	Información	Referencia 1 y 3
Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres	Operadores	Referencia 1 y 3
Usa secuencias ejecutándose al mismo tiempo	Paralelismo	Referencia 1 y 3
Prueba y reconoce errores en las secuencias de programas y es capaz de arreglarlos	Pruebas de depuración	y Referencia 1 y 3
Emplea código existente en un nuevo proyecto	Reutilización y remezcla	Referencia 1 y 3
Representa las soluciones como una serie de pasos	Secuencias	Referencia 1 y 3

Apreciando el mapeo de criterios mínimos de la tabla 32, se puede notar que la abstracción y el paralelismo son los elementos de CT que son tenidos mayormente en cuenta por los autores que proponen criterios de evaluación mediante herramientas. Los criterios relacionados con el manejo de información e interactividad con el usuario también se consideran en varias referencias, seguidos de los relacionados con control de flujo, depuración, sincronización y otros que se encuentran en menor medida.

Tabla 32. Mapeo de criterios mínimos encontrados en herramientas

Criterios	Elementos relacionados	Referencia
-----------	------------------------	------------

3. Caracterización de criterios mínimos y mecanismos de evaluación

Descompone el problema en partes pequeñas	Abstracción	Referencia 6
Emplea más de un hilo y más de un sprite, define bloques, usa clones	Abstracción y descomposición del problema	Referencia 4 y 7
Dispone de dos o más hilos inician con el mismo bloque	Abstracción y descomposición del problema	Referencia 4 y 7
Toma la información para manipularla, analizarla y visualizarla	Análisis de información	Referencia 8
Emplea bloques como repetir, por siempre y otros bucles.	Control de flujo	Referencia 4 y 7
Determina eventos a través del uso del teclado, ratón y recursos multimedia	Eventos	Referencia 7
Permite realizar corrección de los errores	Facilidad de pruebas (depuración)	Referencia 6
Realiza una adecuada interacción con el usuario	Interactividad con el usuario	Referencia 6
Brinda herramientas al usuario para interactuar con el sistema, a través del teclado y ratón	Interactividad del usuario	Referencia 4
Emplea modelos computacionales para entender conceptos y/o solucionar problemas	Modelado y simulación	Referencia 8
Hace uso de condicionales y operadores lógicos	Paralelismo	Referencia 4 y 7
Reconoce la diferencia entre procesos paralelos y seriales	Paralelismo	Referencia 6
Maneja correctamente los mensajes, sincroniza el fondo	Paralelismo	Referencia 7
Utiliza bloques como esperar y parar, maneja mensajes correctamente	Pensamiento lógico	Referencia 4
Emplea correctamente las variables, emplea listas	Representación de la información	Referencia 6 y 7

Modifica las propiedades de los sprites, usa operaciones con variables o listas	Representación de los datos	Referencia 4
Emplea una comunicación entre los objetos correcta, usa comandos disponibles para ayudar al flujo del programa	Sincronización	Referencia 6
Selecciona herramientas computacionales efectivas, evalúa diferentes enfoques de un problema, brinda soluciones informáticas modulares a los problemas, emplea estrategias de resolución de problemas y creación de abstracciones	Solución de problemas computacionales	Referencia 8

En el mapeo de criterios mínimos encontrados en las implementaciones, se puede observar que los criterios son muy variados ya que se relacionan con diversos elementos. Sin embargo, se puede notar que hay una tendencia entre los autores a definir criterios relacionados con los elementos: abstracción, descomposición de problemas, información y paralelismo.

Tabla 33. Mapeo de criterios mínimos encontrados en implementaciones

Criterios	Elementos relacionados	Referencia
Hace comentarios sobre los datos utilizados para la solución del problema.	Abstracción	Referencia 17
Es capaz de generalizar por atributos	Abstracción	Referencia 12
Intenta simplificar un problema ignorando detalles innecesarios, busca patrones	Abstracción	Referencia 11
Emplea más de un hilo y más de un sprite, define bloques, usa clones	Abstracción y descomposición del problema	Referencia 16

3. Caracterización de criterios mínimos y mecanismos de evaluación

Dispone de dos o más hilos inician con el mismo bloque	Abstracción y descomposición del problema	Referencia 16
Descompone un problema en partes más pequeñas	Algoritmos	Referencia 13
Utiliza grupos de bloques como unidades en programas más grandes	Algoritmos	Referencia 13
Repite pasos mediante el uso de bloques	Bucles	Referencia 18
Ejecuta acciones como consecuencia de un acontecimiento	Condicionales	Referencia 18
Emplea bloques como repetir, por siempre y otros bucles.	Control de flujo	Referencia 16
Emplea bucles correctamente, responde a eventos, los comandos se ejecutan al cumplir con las condiciones esperadas	Control de flujo	Referencia 11
Reconoce errores en las secuencias de programas y es capaz de arreglarlos.	Depuración	Referencia 13
Si hay problemas secundarios en el problema, puede administrar los procesos de solución de estos	Descomposición	Referencia 17
Aísla los atributos	Descomposición de problemas	Referencia 12
Fracciona un problema en varias partes y resuelve una a una	Descomposición de problemas	Referencia 11
Puede generar secuencias de comportamientos que se repiten	Diseño de algoritmos	Referencia 12
Muestra como una tarea se efectúa como una consecuencia de un acontecimiento	Eventos	Referencia 18
Formula problemas de forma que el computador puede ayudar a resolverlos	Formulación y solución de problemas	Referencia 10

Realiza conexiones entre el problema actual y los problemas encontrados anteriormente	Generalización	Referencia 17
Transfiere conocimiento en la solución de otros problemas	Generar, revisar y evaluar soluciones	Referencia 10
Ha completado la solución	Implementación de una solución	Referencia 12
Realiza mejoras paulatinas ampliando la solución	Incremental e iterativo	Referencia 11
Procesa información de una manera lógica, representa la información de manera abstracta	Información	Referencia 10
Almacenamiento, recuperación y actualización de la información	Información	Referencia 11 y 18
Facilita la interacción con el jugador	Interacciones del jugador	Referencia 16
Brinda herramientas al usuario para interactuar con el sistema, a través del teclado y ratón.	Interactividad del usuario	Referencia 16
Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres	Operadores	Referencia 18
Hace uso de condicionales y operadores lógicos	Paralelismo	Referencia 16
Trabaja con otros para resolver diferentes partes del problema al mismo tiempo, realiza más de una acción al mismo tiempo	Paralelismo	Referencia 11
Usa secuencias ejecutándose al mismo tiempo	Paralelismo	Referencia 18
Crea una lista de pasos para resolver los problemas, usa las matemáticas	Pensamiento algorítmico	Referencia 11
Utiliza bloques como esperar y parar, maneja mensajes correctamente	Pensamiento lógico	Referencia 16

3. Caracterización de criterios mínimos y mecanismos de evaluación

Evidencia un proceso iterativo	Proceso de diseño (Incremental e iterativo)	Referencia 13
Prueba un atributo al tiempo	Pruebas sistemáticas	Referencia 12
Corre el programa para verificar que la solución sea correcta y soluciona los errores	Pruebas y depuración	Referencia 11
Decide si los datos que se utilizarán para la solución del problema son adecuados o no	Razonamiento	Referencia 17
Observa los elementos comunes y los reconoce	Reconocimiento de patrones	Referencia 12
Identifica los símbolos de los bloques y cartas de acción.	Representación de la información	Referencia 13
Modifica las propiedades de los sprites, usa operaciones con variables o listas	Representación de los datos	Referencia 16
Resuelve problemas de una manera eficiente	Representación y solución de problemas	Referencia 10
Comparte sus programas y revisa los de sus compañeros para mejorar el propio	Reutilización, conexión y cuestionamiento	Referencia 11
Representa las soluciones como una serie de pasos	Secuencias	Referencia 18
Programa secuencias sintácticamente correctas.	Software (secuencias)	Referencia 13
Aborda correctamente el tema	Tema de mitigación (solución de problemas)	Referencia 16

3.4.3. Mapeo de los mecanismos de evaluación

La fase de descripción brindó la información de los mecanismos de evaluación encontrados en la literatura. El proceso de mapeo se realizó teniendo en cuenta los núcleos temáticos: (1) aproximaciones como marcos de referencia y modelos; (2)

herramientas diseñadas para la evaluación de CT o para evaluar prácticas de programación y (3) propuestas implementadas en el aula de clase.

Aunque se hallan similitudes entre algunos de los mecanismos de evaluación, estos están definidos de diferentes maneras. Para facilitar el mapeo, se clasificaron los mecanismos en 6 grupos:

- **Análisis manual:** corresponde al examen del cumplimiento de las tareas de los estudiantes de forma manual.
- **Análisis automático:** se refiere al estudio de las tareas de los estudiantes mediante una herramienta que facilita la automatización.
- **Entrevista:** es una conversación con preguntas estructuradas o semi-estructuradas en las que el evaluador pregunta y el estudiante responde.
- **Encuesta:** es un instrumento donde se recopilan los datos de las respuestas mediante un cuestionario previamente diseñado.
- **Observación:** es un procedimiento que permite al docente apreciar o verificar el desempeño de los estudiantes, la observación directa facilita la apreciación, mientras que la observación indirecta facilita la verificación
- **Revisión de registros anecdótico:** es la verificación por parte del docente acerca de los planeadores empleados por los estudiantes en su proceso de diseño e implementación, por ejemplo: las bitácoras, hojas de resolución de problemas e informes de reflexión y diseño.

El mapeo de mecanismos a nivel de marcos de referencia se puede apreciar en la tabla 34. Al grupo de mecanismos de análisis manual corresponden la revisión manual del desarrollo de las tareas, teniendo en cuenta los elementos y criterios seleccionados en cada tarea. Esta revisión manual fue propuesta en los tres (3) marcos de referencia estudiados. La entrevista fue tomada en cuenta en [82], para indagar acerca de los artefactos y acerca del proceso de diseño y la revisión de registros anecdóticos fue considerada en [125] para examinar los informes de diseño y reflexión.

Tabla 34. Mapeo de mecanismos encontrados en Marcos de referencia

Mecanismos de evaluación	Referencia	Grupo
--------------------------	------------	-------

3. Caracterización de criterios mínimos y mecanismos de evaluación

Analizar los proyectos, registrando los bloques de los proyectos y su frecuencia de uso.	Referencia 1	Análisis manual
Entrevistar a los estudiantes a través de preguntas basadas en artefactos-objetos.	Referencia 1	Entrevista
Examinar el informe de diseño creativo.	Referencia 3	Revisión de registros anecdóticos
Inspeccionar el informe de reflexión.	Referencia 3	Revisión de registros anecdóticos
Preguntar acerca del diseño de escenarios.	Referencia 1	Entrevista
Revisar el cumplimiento de cada criterio de acuerdo a tareas realizadas por sus estudiantes.	Referencia 2	Análisis manual
Revisar el desarrollo de tareas diseñadas de acuerdo al marco conceptual definido.	Referencia 3	Análisis manual

La tabla 35 muestra el mapeo de mecanismos a nivel de herramientas, se puede apreciar que proponen análisis automático, excepto en [84] que se considera un análisis manual soportado en la tabla que entrega la herramienta, que facilita el proceso, pero lo no automatiza completamente.

Tabla 35. Mapeo de mecanismos encontrados en herramientas

Mecanismos de evaluación	Referencia	Grupo
Analizar automáticamente del código Scratch.	Referencia 4	Análisis automático
Analizar automáticamente el código escrito por los estudiantes por medio del prototipo online MindReader.	Referencia 9	Análisis automático
Analizar automáticamente las actividades realizadas a través de herramientas computacionales definidas en el marco de referencia propuesto.	Referencia 8	Análisis automático
Analizar automáticamente los proyectos entregados por los estudiantes.	Referencia 7	Análisis automático

Analizar automáticamente y en tiempo real de los ejercicios de los estudiantes.	Referencia 5	Análisis automático
Analizar del cuadro entregado por la herramienta, que relaciona la solución con los elementos definidos	Referencia 6	Análisis manual

Los mecanismos de evaluación de los seis (6) grupos han sido considerados en los estudios contemplados dentro del mapeo de términos a nivel de implementaciones (Tabla 36). Los mecanismos de los grupos análisis automático y análisis manual, realizan la misma función de manera diferente, es decir, proponen examinar el cumplimiento del desarrollo de las tareas uno soportado en una herramienta que calcula el resultado y el otro mediante el trabajo manual del docente.

Algunos mecanismos son aplicados a los estudiantes de manera directa como las entrevistas que buscan indagar acerca de las prácticas y perspectivas de programación de los estudiantes, las encuestas de autoeficacia que preguntan acerca de la confianza de cada uno en su capacidad para lograr los resultados, la revisión de registros anecdóticos donde los estudiantes deben entregar sus bitácoras y hojas de resolución de problemas para ser evaluados y la observación directa de sus actitudes.

Tabla 36. Mapeo de mecanismos encontrados en implementaciones

Mecanismos de evaluación	Referencia	Grupo
Analizar automáticamente el código de los estudiantes empleando Dr. Scratch	Referencia 16	Análisis automático
Analizar automáticamente las habilidades de los estudiantes en el desarrollo de las tareas.	Referencia 15	Análisis automático
Analizar cualitativamente los elementos CT.	Referencia 16	Análisis manual
Analizar los juegos terminados empleando una guía.	Referencia 18	Análisis manual
Calificar de un instrumento de 23 ítems de respuesta múltiple y respuestas abiertas	Referencia 10	Análisis manual

3. Caracterización de criterios mínimos y mecanismos de evaluación

Entrevistar a los estudiantes para conocer sus prácticas y perspectivas.	Referencia 18	Entrevista
Estimar el grado de cumplimiento en el desarrollo de los retos propuestos en el juego.	Referencia 12	Análisis automático
Indagar los resultados del desarrollo de actividades interactuando con KIBO.	Referencia 13	Entrevista
Observar a los participantes y tener conversaciones informales con ellos.	Referencia 18	Observación
Revisar las bitácoras y hojas de resolución de problemas.	Referencia 18	Revisión de registros anecdóticos
Valorar el resultado de las tareas propuestas.	Referencia 14	Análisis manual
Valorar las respuestas de los estudiantes de la encuesta de autoeficacia.	Referencia 17	Encuesta
Valorar una encuesta de autoeficacia.	Referencia 11	Encuesta
Verificar las prácticas mediante una grabación de video de los estudiantes programando.	Referencia 15	Observación
Verificar un video grabación de los estudiantes interactuando con la herramienta.	Referencia 13	Observación
Verificar un video grabación de los estudiantes mientras desarrolla las tareas diseñadas.	Referencia 12	Observación

3.5.Fase de construcción teórica global

En esta fase se presentan los resultados del estudio. Se ha podido evidenciar que para estimar el desarrollo de CT: (1) los autores estudiados aportan definiciones de elementos bajo el razonamiento de sus revisiones literarias desde la perspectiva

teórica, (2) las asociaciones educativas y universidades tienden a considera CT de una manera holística, considerando más allá de los saberes de los estudiantes, (3) las compañías de la industria informática proponen cursos para docentes en los que definen CT y sus características, (4) los marcos de referencia aportan elementos que se han definido ya sea en sistemas reales o modelos implementados, (5) las herramientas proponen elementos dirigidos a las prácticas de programación, ya que ese es el enfoque de la mayoría de herramientas estudiadas y (6) los autores que se refieren a implementaciones en el aula de clase describen los elementos de acuerdo a sus prácticas en el aula, definiendo los elementos, criterios y mecanismos de acuerdo a su experiencia.

3.5.1. Elementos de CT

Para la caracterización de elementos de CT se realizaron diferentes procedimientos como selección de documentos, generación de categorías a partir de los autores, extracción de la información y mapeo de los elementos para cada categoría empleando criterios de selección.

Los autores proponen elementos CT que han sido el resultado de revisiones literarias, las asociaciones educativas se enfocan en los estándares que se deben plantear para CT, la industria lo toma desde la perspectiva de las competencias a nivel laboral, los marcos de referencia buscan integrarlo todo, las herramientas en su mayoría se centran en la evaluación automatizada y las implementaciones en el aula buscan concluir desde la práctica.

Se puede observar en el Anexo A, Tabla a2 todos los elementos resultantes de cada mapeo unidos, aunque son muchos, la mayoría solo están contemplados en una categoría, por lo que se tomaron los elementos resultantes en dos (2) categorías o más, como criterio de selección. Los elementos resultantes se pueden ver en la Tabla 37.

Tabla 37. Elementos de CT

Elemento CT	No Categorías	Porcentaje
Abstracción	6	100
Generalización de patrones	5	83,33

3. Caracterización de criterios mínimos y mecanismos de evaluación

Reconocimiento de patrones	2	33,33
Modularización, Modularidad	2	33,33
Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	6	100
Simulación	2	33,33
Descomposición, Descomposición de problemas	6	100
Declaraciones condicionales, Lógica condicional	2	33,33
Paralelismo	3	50
Recolección de información	3	50
Análisis de información	3	50
Representación de información	3	50
Evaluación	2	33,33
Depuración	4	66,67

Como se mencionó anteriormente, existe ambigüedad en algunas de las definiciones de los elementos, por lo que se hizo necesario definirlos. Las definiciones se pueden apreciar en la Tabla 38.

Tabla 38. Definiciones de elementos de CT

No.	Elemento	Definición
1	Abstracción	<p>Construir algo grande uniendo pequeñas partes que lo componen [82].</p> <p>Reducir la complejidad hasta llegar a la idea principal [26].</p> <p>Identificar y extraer la información relevante para definir la idea principal [23].</p> <p>Remover detalles innecesarios [100].</p> <p>Proceso de crear algo simple a partir de algo complicado al omitir los detalles irrelevantes, al encontrar los patrones relevantes y al separar las ideas de los detalles tangibles.</p> <p>La abstracción se utiliza en la definición de patrones, generalización de instancias y parametrización [3].</p>

		<p>Habilidad de generalizar un patrón y priorizar que es pertinente y que no en relación a un problema [105].</p> <p>Y generalización de patrones: es la habilidad de filtrar información no necesaria para resolver problemas, generalizar información y extrapolar principios y conceptos desde la información presente [116], [117].</p> <p>Eliminar los detalles específicos de un conjunto de problemas para encontrar la solución genérica que se pueda utilizar para todos ellos, o representando un sistema complicado con un modelo simple reflejando la capacidad de dividir un problema en partes más pequeñas que sean más fáciles de entender, programar y depurar [109].</p> <p>Desarrollar modelos o simulaciones de un fenómeno natural o artificial usarlo para hacer predicciones acerca del mundo y analizar su eficacia y validez [102].</p> <p>Crear un modelo o simulación visual de un problema que incorpore solo los partes más importantes [104].</p> <p>Sub categorías de abstracción: recolección y análisis de información, reconocimiento de patrones y modelado [96].</p> <p>Mecanismos de abstracción: recursión, generalización; ocultamiento de la información y encapsulamiento [128].</p>
2	Generalización de patrones	<p>Crear reglas, principios o teorías de observación de patrones para probar resultados predichos [23]</p> <p>Reconocer y usar similitudes [100]</p> <p>Transferir el proceso de solución de un problema a una gran variedad de situaciones.[113] para resolver problemas efectivamente y eficientemente [96]</p> <p>Y abstracción: resolver problemas de tipos similares porque en pasadas experiencias se han resuelto. [112]</p>
3	Reconocimiento de patrones	<p>Observar patrones, tendencias y regularidades en la información [23], [104], ayuda a los estudiantes a predecir con mayor cercanía que ocurrirá después [105] o que resultados se obtendrán [112].</p> <p>Capacidad de detectar puntos en común o disparidades que lo ayudarán a hacer predicciones o conducir a atajos [117].</p>

3. Caracterización de criterios mínimos y mecanismos de evaluación

4	Modularidad	Desarrollo de procesos autónomos, los cuales están encapsulados en un conjunto de comandos usados con frecuencia que desempeñan una función específica y podrían ser usado en el mismo o en diferente problemas [113].
5	Diseño de algoritmos	<p>Un algoritmo es una abstracción de un procedimiento paso a paso para tomar una entrada y producir un resultado deseado. [129]</p> <p>Exactitud del algoritmo, que siga las especificaciones. Correcto funcionamiento del algoritmo, incluso en los casos de límites [45].</p> <p>Crear y ordenar una serie de instrucciones o pasos para resolver problemas o hacer tareas [23], [26], [116], [117] y otros similares a ese [104], [113]</p> <p>Hacer pasos y reglas [100]</p>
6	Simulación (modelado)	<p>Representación o modelo de un proceso [26]</p> <p>Desarrollar modelos para imitar los procesos del mundo real [23]</p> <p>Construir modelos para representar como opera un sistema o como lo hará en el futuro [96]</p> <p>Hacer predicciones de resultados a partir del flujo de información de un problema planteado. [98] Resolver los problemas actuales a través de un modelo o desarrollar un nuevo sistema. [98]</p> <p>Implementar un algoritmo como bancos de pruebas para identificar qué circunstancias y abstracciones considerar [114], [115].</p>
7	Descomposición de problemas	<p>Desglosar tareas en partes más pequeñas y manejables [23], [26], [100], [104], [113], [116] que se pueden resolver de forma independiente. [118]</p> <p>Las partes divididas deben elementos funcionales que en conjunto comprende todo el sistema o problema. [96]</p> <p>Pequeñas piezas que forman un gran problema. [105]</p> <p>Capacidad de descomponer un sistema en componentes o una tarea en procedimientos para que uno pueda explicar</p>

		<p>los elementos de un proceso a otra persona o una computadora. [117]</p> <p>Explicación de una acción en detalle [112]</p>
8	Lógica – Razonamiento lógico	<p>Predecir y analizar [100].</p> <p>Se refiere a las consecuencias de un valor dada una declaración (falso o verdadero) [115].</p> <p>Capacidad del proyecto para comportarse de manera diferente dependiendo de la situación [109].</p> <p>Encontrar la asociación de patrones entre diferentes eventos [98]</p> <p>Habilidad para tomar decisiones en determinada circunstancia, soportando muchos resultados [82]</p>
9	Paralelismo	<p>El paralelismo se refiere al procesamiento de tareas o secuencias de instrucciones que se realizan al mismo tiempo [74], [82], [109] con el fin de alcanzar un objetivo común [23].</p> <p>Organizar recursos para llevar a cabo tareas simultáneamente con el fin de alcanzar un objetivo común [26], [102].</p>
10	Recolección de información	Reunir información apropiada [26], [23].
11	Análisis de información	<p>Encontrar el sentido de la información, reconocer patrones y trazar conclusiones [26], [23]</p> <p>Organizar y analizar información [116]</p> <p>Y la visualización: información resumida en forma esquemática, que incluye atributos o variables para unidades de información o la capacidad de representar redes o sistemas complejos en representaciones visuales [117]</p>
12	Representación de información	Representar y organizar datos en gráficos, cuadros, palabras o imágenes apropiados [23], [26]
13	Probar y depurar	<p>Detectar, identificar y solucionar errores [104], [100] cuando una solución no funciona como se supone que debería [96], mientras se está desarrollando [118].</p> <p>Analizar problemas y errores en lógica o en</p>

3. Caracterización de criterios mínimos y mecanismos de evaluación

		actividades[114]. Determinar problemas en un algoritmo [115].
14	Evaluar	Generar juicios [100]. Asegurar que la solución está bien [104].

El análisis de las definiciones lleva a concluir que Modularidad y Simulación se encuentran inmersas en las prácticas de programación. La programación se relaciona con la capacidad de analizar problemas y situaciones, identificar sus elementos, modelar los datos y procesos, y crear o refinar un conjunto de instrucciones para indicarle a un computador como desarrollar una tarea determinada [130].

La modularidad según su definición: emplea comandos usados con frecuencia para realizar tareas [113], se encuentra estrechamente relacionado, al igual que simulación que es la construcción de modelos [26], [96], [98] es decir abstracciones del mundo real [23], [114], [115], las soluciones realizadas en programación representan estas abstracciones.

El diseño de algoritmos se refiere a la creación de pasos e instrucciones [23], [26], [116], [117] que funcionen correctamente [45]. Lo que nos lleva a entender que también se encuentra inmerso en la programación, no obstante, para los propósitos de este estudio se va a asumir dentro de los elementos a evaluar, porque en programación se requiere de la creación de secuencias de manera correcta, lo que constituye el elemento más básico a evaluar.

3.5.2. Criterios mínimos para la evaluación de CT y sus mecanismos de evaluación

Una vez identificados los elementos que componen CT, se describen los criterios con los que estiman su desarrollo y los mecanismos con los que los evalúan, estos resultados provienen del mapeo realizado, ver Anexo A, tabla a2. Los resultados se pueden encontrar en la Tabla 39.

Tabla 39. Elementos de CT, criterios mínimos y mecanismos de evaluación

Elementos CT	Criterios mínimos	Mecanismo de evaluación sugeridos
Abstracción	Simplifica un problema ignorando detalles innecesarios Es capaz de generalizar por atributos Descompone el problema en partes pequeñas.	Análisis automático, análisis manual, observación indirecta, revisión de registros anecdóticos
Generalización de patrones	Realiza conexiones entre el problema actual y los problemas encontrados anteriormente	Entrevista, encuesta
Reconocimiento de patrones	Observa los elementos comunes y los reconoce	Análisis automático, análisis manual, observación indirecta
Diseño de algoritmos	Genera secuencias de comportamientos que se repiten Plantea soluciones paso a paso para tomar entrada y producir la salida deseada	Análisis automático, análisis manual, observación indirecta, observación directa
Descomposición	Fracciona un problema en varias partes y resuelve una a una por separado Aísla los atributos	Análisis automático, análisis manual, entrevista, observación indirecta
Razonamiento lógico	Ejecuta acciones como consecuencia de un acontecimiento	Análisis automático, análisis manual, observación indirecta
Paralelismo	Usa secuencias ejecutándose al mismo tiempo Reconoce la diferencia entre procesos paralelos y seriales	Análisis automático, análisis manual, observación directa Entrevista

3. Caracterización de criterios mínimos y mecanismos de evaluación

Recolección de información	Recoge información en un contexto	Análisis automático, análisis manual, observación indirecta
Análisis de información	Toma la información para manipularla, analizarla y visualizarla	Análisis automático, análisis manual, observación indirecta
Representación de información	Emplea variables, listas y otras estructuras de datos	Análisis automático, análisis manual, observación indirecta
Depuración	Reconoce errores y es capaz de corregirlos	Entrevista, encuesta, observación indirecta
Evaluación	Analiza y examina la información Utiliza criterios para estimar la calidad de las soluciones	Entrevista, encuesta

La información contenida en la Tabla 39, resultante del mapeo de elementos, criterios mínimos y mecanismos de evaluación respecto a los distintos objetos de estudio (marcos de referencia, conceptos teóricos, implementaciones prácticas, etc.), permite evidenciar lo siguiente:

- La abstracción tiene una estrecha relación con reconocimiento de patrones y descomposición de problemas.
- Los mecanismos de evaluación se pueden dividir en dos grupos: (i) los que se encargan de evaluar los resultados obtenidos y (ii) los que se encargan de indagar al estudiante.
- Los mecanismos como entrevista y encuesta son muy subjetivos, ya que su resultado depende de las creencias que el estudiante tiene sobre sí mismo.
- Se recomienda reemplazar el análisis manual cada vez que se pueda por análisis automático con el fin de ahorrar tiempo de calificación y aumentar la validez de las pruebas, eliminando del proceso errores humanos.

- Los mecanismos como entrevista y observación indirecta se recomiendan únicamente en casos donde los evaluadores cuenten con el tiempo suficiente, ya que estos instrumentos requieren de un gran consumo de tiempo debido a la individualidad de la prueba.
- La observación directa es práctica cuando se evalúan pocos estudiantes, sin embargo, cuando se tienen varios estudiantes se recomienda como un mecanismo de apoyo a otros que deben ser empleados.
- La revisión de registros anecdóticos da al docente información acerca de la planeación y solución de problemas por parte de sus estudiantes, lo que constituye una información muy valiosa para determinar el desarrollo de CT, sin embargo, mientras este no sea un mecanismo automatizado, va a tomar mucho tiempo revisar cuando hay muchos estudiantes.
- Independientemente del mecanismo que el docente vaya a emplear para la evaluación en el aula, se recomienda siempre contar con rubricas que aseguren que todos los estudiantes van a ser valorados de la misma manera.

3.6.Fase de extensión y publicación

Esta fase del estudio se cumplió por medio de la inclusión de los resultados del proceso de caracterización de los criterios mínimos y los mecanismos de evaluación en el artículo: “Modelo de evaluación basado en evidencia para estimar el desarrollo de CT”, esta información ha servido como insumo para soportar la propuesta.

3.7.Conclusiones

- Debido a que CT es una habilidad que comprende una serie de elementos es importante que estos se identifiquen y se definan, ya que no hay un consenso en la literatura, incluso en su grado de especificidad.
- Los criterios de evaluación deben ser evaluados a través de su comparación con el valor medio de todas las evaluaciones anteriores.
- Cabe resaltar que el interés de esta caracterización era hacerlo para los criterios mínimos y mecanismos de evaluación, algunos autores definen algunos

3. Caracterización de criterios mínimos y mecanismos de evaluación

criterios que se pueden evaluar en estudiantes avanzados, aunque estos salen del marco de trabajo.

- Los criterios mínimos deben ser muy concretos, observables y evaluables, deben reflejarse como consecución de un aprendizaje básico.
- Como la evaluación comprende un proceso extenso que abarca diferentes elementos, se recomienda combinar varios mecanismos de evaluación para llegar a una estimación más precisa a la realidad.
- Los mecanismos de evaluación automatizados facilitan la evaluación de muchos estudiantes al mismo tiempo, no obstante, es importante tener en cuenta que la realimentación que a los estudiantes le dé la herramienta debe ser cercana a sus necesidades.

Capítulo 4

4. Modelo de evaluación basada en evidencia

En el capítulo anterior se presentó el estudio realizado para identificar y caracterizar los criterios mínimos para estimar el desarrollo de CT y sus mecanismos de evaluación, insumo necesario para el diseño del modelo propuesto en este capítulo.

El proceso de evaluación para estimar el desarrollo de CT en el aula de clase ha alcanzado gran relevancia y se ha promovido debido a los siguientes factores: (i) se considera fundamental promover el desarrollo de habilidades de alto nivel en los estudiantes en edad escolar, (ii) se reconoce que los docentes no tienen facilidades para implementar esta evaluación debido a la falta de consenso y herramientas que faciliten este proceso y (iii) el avance tecnológico posibilita la automatización del proceso de evaluación, manejando una gran cantidad de información acerca de los estudiantes, que el docente no podría manejar regularmente debido al volumen de los datos y la precisión en las medidas.

A continuación se describe el Modelo de Evaluación objeto de la investigación. Este está integrado por los hallazgos y propuestas realizados en virtud del proceso investigativo cumplido; su estructura está basada en términos de los reportes que dan cuenta de los productos desarrollados y que integrados, corresponden con el propósito establecido. Teniendo en cuenta las necesidades identificadas y las soluciones planteadas, el modelo de evaluación facilita la estimación del desarrollo del CT basado en la evidencia entregada por los estudiantes en edad escolar, a través de las prácticas de programación.

4.1. Alcance

El diseño del modelo de evaluación propuesto se deriva de un amplio estudio relacionado con las características que constituyen el Pensamiento Computacional, los criterios mínimos que determinan la estimación de su desarrollo y los mecanismos con los que se puede verificar el cumplimiento de dichos criterios, el detalle de este procedimiento se encuentra en el Capítulo 3. Se especifica el mapeo de términos y los resultados obtenidos, no obstante, a pesar de la rigurosidad del estudio, se resalta la posibilidad que ciertas características deseables por ciertos diseñadores no fuesen incluidas de manera explícita, al igual que se brindó mayor importancia a los estudios que más se relacionan con este trabajo.

El propósito del proceso de evaluación es determinar en qué medida se están cumpliendo los objetivos relacionados con los aprendizajes que se espera logren los estudiantes. Para el diseño del modelo fue necesario acotar la evaluación, teniendo en cuenta a quienes va dirigida, el propósito específico, los elementos que se van a evaluar, de qué manera se hace, cuando y donde tiene lugar dicho proceso. La Figura 1 responde las preguntas relacionadas.



Figura 1. Resumen del alcance del proyecto

4. Modelo de evaluación basada en evidencia

Este modelo es una guía que sirve para evaluar continuamente los criterios mínimos que permiten estimar el desarrollo del pensamiento computacional en estudiantes en edad escolar, mediante mecanismos de evaluación que consideran las tareas realizadas por los estudiantes en sus prácticas computacionales en el aula de clase.

4.2. Proceso de evaluación

La evaluación es un proceso sistemático, que se incorpora a entornos educativos. Básicamente se compone de tres pasos: (i) obtener información, consiste en conseguir datos a través de instrumentos y procedimientos válidos y fiables, (ii) formular juicios, se refiere a coger los datos obtenidos, realizar un análisis y emitir una valoración, y (iii) tomar decisiones, de acuerdo con los juicios de valor formulados se determinan las medidas a seguir [34]. La figura 2 muestra el diagrama del proceso de evaluación.

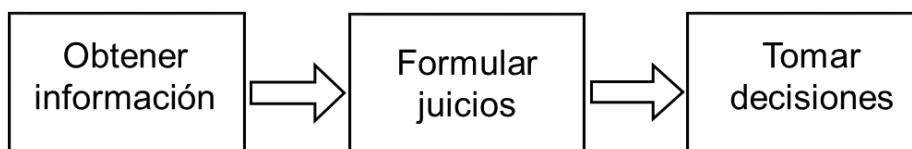


Figura 2. Proceso de evaluación. Basado en [34].

En este sentido, es necesario definir los detalles de cada uno de los pasos en el proceso de evaluación, dar a conocer cómo se pueden realizar enfocándose en estimar el desarrollo de CT. De esta manera, el modelo propuesto se dirige principalmente a fortalecer los componentes del proceso de evaluación y relacionarlos con la estimación del desarrollo de CT, teniendo en cuenta los resultados obtenidos en el Capítulo 3 y el alcance del estudio.

4.3. Construcción del modelo

El modelo diseñado busca brindar información y herramientas a los docentes, que soporten el proceso de evaluación para estimar el desarrollo de CT. Está planteado considerando el diseño de evaluaciones basadas en evidencia (ECD) y el análisis

exploratorio de datos, a través de un proceso de minería de datos educativa (EDM: Educational Data Mining). A continuación se presenta el diseño del modelo, las partes que lo constituyen y el esquema que describe el proceso.

4.3.1. Diseño de evaluaciones basadas en evidencia

Como se mencionó anteriormente, el modelo de evaluación toma como marco de referencia el ECD. Cada una de sus capas se describe a continuación de acuerdo a su implementación.

Capa del análisis de dominio: se evidencia en el Capítulo 2 (marco teórico y estado del arte), en ésta se tuvo en cuenta la información más relevante acerca de CT.

Capa de modelado del dominio: se puede comprobar en el Capítulo 3, en la caracterización se organizó la información de los elementos CT, criterios mínimos y mecanismos de evaluación, de acuerdo con los argumentos de valoración establecidos. Esta capa se enfocó en la categorización de la información más relevante acerca de la evaluación de criterios para estimar el desarrollo de CT.

Marco de evaluación conceptual (CAF): condujo el proceso de diseño de los modelos de competencia, evidencia y tareas.

- El modelo de competencias detalla los elementos de CT a tener en cuenta, cada uno representado por un nodo. Como no existe un acuerdo formal acerca de los elementos de CT, se caracterizaron en el Capítulo 3; los resultados se encuentran consignados en la Tabla 37.
- El modelo de evidencia está compuesto por las reglas de evidencia y el modelo estadístico. Las reglas de evidencia corresponden a los criterios con los que se evalúa a los estudiantes, en este caso los criterios mínimos consignados en la Tabla 39, y el modelo estadístico está compuesto por las reglas de clasificación dispuestas para la evaluación, estas se pueden encontrar en la sección 4.3.4.

4. Modelo de evaluación basada en evidencia

- El modelo de tareas describe las situaciones con las que el estudiante se va a encontrar para proporcionar evidencia sobre aspectos de la competencia. Lo conforman la herramienta HERA-CT, que presenta los desafíos para ser resueltos por los estudiantes y brinda realimentación, detallada en el capítulo 5, y por el entorno gráfico de desarrollo Scratch, un lenguaje de programación visual basado en bloques que los estudiantes emplean para solucionar los desafíos

El diagrama del CAF para el modelo descrito, se puede apreciar en la Figura 3.

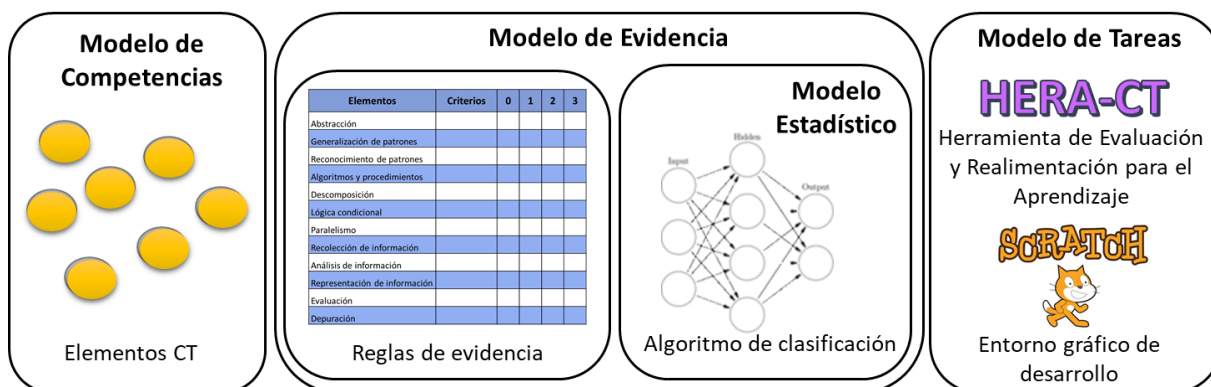


Figura 3. Implementación ECD

4.3.2. Modelo de evaluación para estimar el desarrollo de CT

El modelo de evaluación propuesto describe un proceso que busca brindar una guía a los docentes en la creación de sus evaluaciones, de tal manera que estas consideren la estimación del desarrollo de CT en los estudiantes, a través de prácticas de programación de problemas contextualizados en diferentes áreas del conocimiento.

La propuesta se compone de cuatro (4) fases que permiten el desarrollo del proceso de evaluación. Los docentes y estudiantes tienen actividades específicas dentro del modelo, se pretende que estas actividades pertenecientes a las cuatro (4) fases sean implementadas por los docentes: (i) fase de iniciación, abarca la definición del área del conocimiento en la que se van a desarrollar los problemas, el objetivo general y los

mecanismos de evaluación, (ii) fase de planeación, en esta se determinan las tareas que los estudiantes van a realizar y se brindan los criterios mínimos de evaluación de cada tarea, (iii) fase de ejecución, se recolecta la información de la evaluación, en esta fase los estudiantes generan la evidencia necesaria para el análisis de los resultados, se compone de dos módulos: del estudiante y estadístico, y (iv) fase de análisis, consiste en la revisión de los resultados entregados después de la clasificación automática, los docentes realizan esta fase con el fin de detectar problemas en el aprendizaje de los estudiantes, dificultades en el desarrollo de CT, fallas en las tareas planteadas y posibles mejoras a este proceso. El diagrama se puede apreciar en la Figura 4, donde también se muestra la conexión entre las fases.

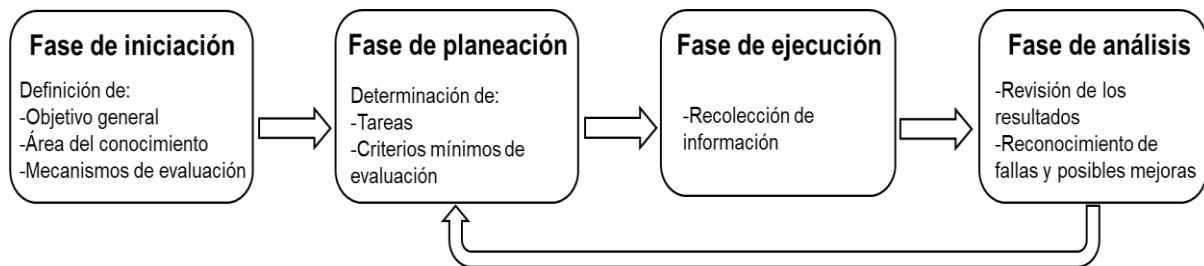


Figura 4. Fases del modelo de evaluación

El modelo propuesto puede ser visto en detalle en la Figura 5, donde se muestran las fases, su interacción y los componentes más importantes que conforman cada una. En la sección 4.3.3 se describen cada una de las fases y sus componentes.

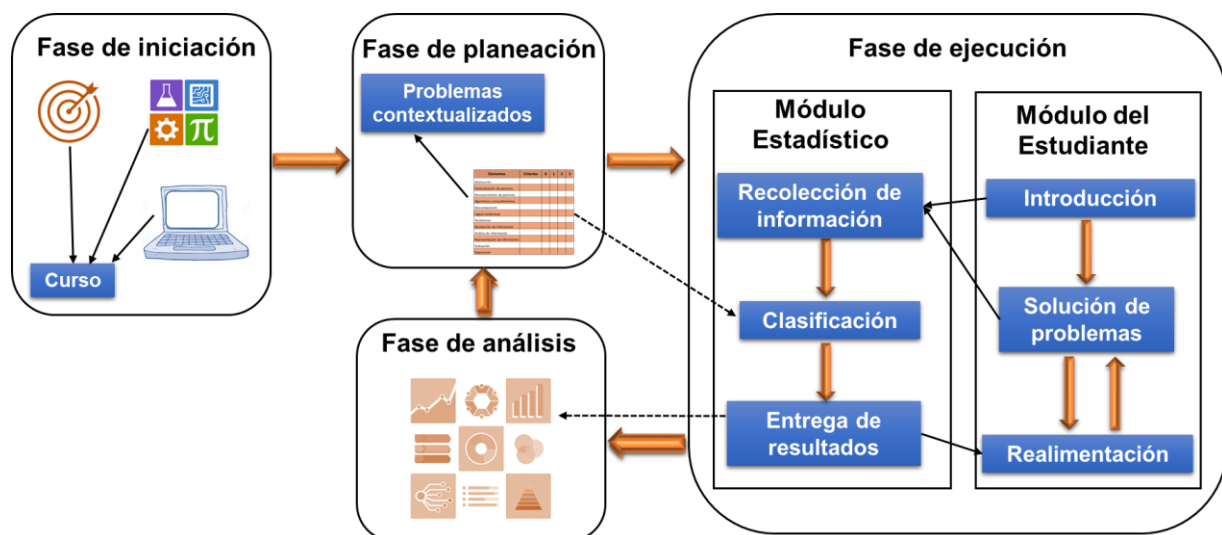


Figura 5. Diagrama detallado: Modelo de evaluación

4. Modelo de evaluación basada en evidencia

Las flechas gruesas indican el flujo de las fases, cuando termina una fase se da inicio a la siguiente. Dentro de los módulos estas flechas indican la secuencia de los componentes.

Las flechas simples indican la comunicación entre componentes de la misma fase y las flechas punteadas indican la interacción entre componentes de diferentes fases.

4.3.3. Fases del modelo

El modelo de evaluación planteado está compuesto por cuatro (4) fases, presentadas en la figura 8, cuya realización es consecutiva y cíclica, ya que el proceso se repite durante todo el periodo académico.

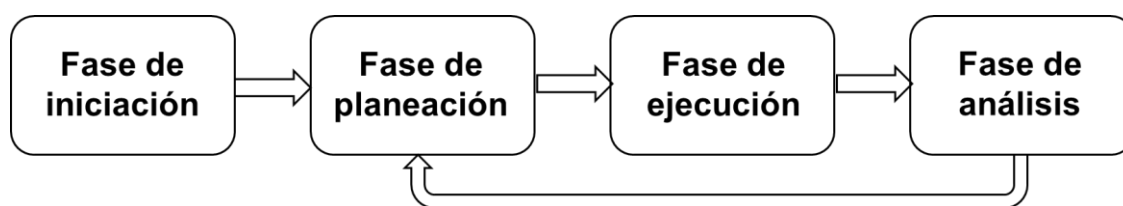


Figura 6. Diagrama general: Fases del modelo de evaluación

Las fases indican los pasos que los docentes deben seguir para diseñar sus evaluaciones de manera sistemática y organizada, teniendo en cuenta los componentes de cada fase y la dependencia que se genera entre estos. A continuación se describe cada una de las fases propuestas en este modelo.

Fase de iniciación

La fase de iniciación consiste en la definición del curso, que contiene: área de conocimiento, grado de los estudiantes y objetivo general de la evaluación, esta información se refiere a la motivación del docente para realizar las prácticas de programación y estimular el desarrollo de CT en sus estudiantes. También, en esta fase se definen los mecanismos que se van a emplear en la evaluación, las herramientas y materiales que se requieren para soportar este proceso y que facilitan la obtención de evidencia confiable y válida. La figura 7 presenta esta fase.



Figura 7. Fase de iniciación

Las actividades de la fase de iniciación deben ser realizadas por el docente, estas incluyen:

- Determinar las herramientas a emplear tanto para codificar como para evaluar
- Definir uno o varios objetivos de acuerdo al área del conocimiento y grado escolar de los estudiantes

Aunque se pueden emplear diferentes herramientas con este modelo, se seleccionó la herramienta de programación Scratch para la ejecución de las tareas por parte de los estudiantes. Scratch es ampliamente usada en contextos escolares y de manera particular por muchos niños [48], además, uno de sus objetivos principales es que se emplee la programación como una herramienta para desarrollar habilidades y mejorar el aprendizaje de diferentes disciplinas [131], lo que permite a los docentes asignar tareas de diferentes áreas del conocimiento para ser desarrolladas con esta misma herramienta. Según los resultados obtenidos en el capítulo 3, para las prácticas de programación se recomienda un análisis automático, razón por la cual se propone en el capítulo 5, una herramienta que soporta el proceso de evaluación y facilita la ejecución de las fases del modelo propuesto.

Fase de planeación

La fase de planeación está relacionada con la preparación de cada una de las sesiones, a realizar en el aula de clase con los estudiantes. La figura 8 muestra los componentes de esta fase: las tareas consisten en la resolución de problemas contextualizados y los criterios mínimos relacionados con ellos.

4. Modelo de evaluación basada en evidencia



Figura 8. Fase de planeación

Las actividades de la fase de planeación deben ser realizadas por el docente, estas son:

- Definir los problemas a resolver por los estudiantes.
- Determinar los criterios mínimos en cada problema, para la estimación del desarrollo de CT en los estudiantes.

Los problemas contextualizados corresponden al área del conocimiento seleccionada por el docente. Se tiene referencia de estrategias para incorporar el desarrollo de CT junto con matemáticas [132], [133], [134], y ciencias naturales [108]. Se ha determinado que para la definición de los problemas el docente deberá incluir un nombre que identifique al problema, un objetivo específico que puede guiar a otros docentes y la descripción que permite a los estudiantes conocer la tarea específica.

Los criterios mínimos son seleccionados por los docentes y tienen una correspondencia directa con cada uno de los problemas. El docente deberá escoger que va a evaluar de acuerdo a la evidencia que los estudiantes pueden aportar a través de las herramientas elegidas. La tabla 39 muestra los criterios mínimos obtenidos en el proceso de caracterización.

Con fines de automatizar el proceso de análisis de información, se emplean dos instrumentos para recolectar la información de la evaluación. Un instrumento se encarga de recolectar la evidencia entregada por el estudiante en sus prácticas de

programación y el otro consiste en una encuesta acerca de su autoeficacia referente a la depuración, generalización de patrones y evaluación.

Para evaluar la evidencia, se realizó un mapeo entre el conjunto de los criterios mínimos resultado de la tabla 39 y la evidencia que se puede recolectar de las prácticas de programación en Scratch, Anexo B, tabla b1. El resultado del mapeo se puede apreciar en la tabla 40. Donde se relacionan los elementos de CT, las acciones que pueden evidenciar con el producto del desarrollo en Scratch y una valoración a cada una dependiendo de su complejidad.

Tabla 40. Mapeo de criterios mínimos y evidencia entregada por Scratch

Elementos de CT		1	2	3
Diseño de algoritmos		Elabora secuencias	No hay código muerto	No hay código duplicado
Abstracción		No hay código muerto		
	Reconocimiento de patrones	Define uno o más bloques	Emplea clones	
	Descomposición de problemas	# hilos por sprite (dos o más)	# sprite tiene eventos (dos o más)	
Paralelismo		# hilos iniciando en bandera verde (dos o más)	# programas en iniciando en clic mouse a sprite o clic any key (2 o más), Envío y recepción de mensajes	# programas en los otros eventos (2 o más) Envío y recepción de mensajes a varios sprites
Información	Recolección	Sensores (entrada	Uso de variables Scratch (no creadas, Ej.: la	Sensores de sprite

4. Modelo de evaluación basada en evidencia

		ratón y teclado)	respuesta a una pregunta, posición x, etc.)	
	Representación	Creación de variables	Variables compartidas con todos los sprites	Creación de listas
	Análisis	Operadores matemáticos básicos, unir palabras	Operadores matemáticos poco comunes, número al azar, mayor, menor, igual	Operadores lógicos, operadores anidados
Lógica condicional		Esperar, repetir, repetir por siempre, si, detener este	Esperar hasta, repetir hasta, si, si no, detener todos los programas del sprite	Condicionales anidados

El docente debe seleccionar de la rúbrica presentada en la tabla 40, los criterios mínimos que van a evaluar en cada uno de los problemas. La valoración equivalente a cero (0) se asigna cuando el estudiante no cumple con ninguno de los criterios de la rúbrica para el elemento de CT.

Se propone una encuesta que debe ser opcional y debe ser elegida por el docente, en caso de que lo requiera y cuando lo requiera. Se propone plantear preguntas simples y cortas, para que los estudiantes puedan comprenderlas, y respuestas múltiples con el fin de facilitar la automatización de los datos recolectados y la visualización de los mismos para su análisis. Para los propósitos de este estudio se van a incluir preguntas referentes a los elementos: evaluación, depuración y generalización. La tabla 41 presenta las preguntas propuestas para cada elemento y sus opciones múltiples.

Tabla 41. Encuesta para los estudiantes

Elemento CT	Preguntas y opciones
-------------	----------------------

Evaluación	¿El producto terminado cumple con los requerimientos planteados?				
	Completamente	Casi todo	Parcialmente	Casi nada	No
Depuración	¿Probaste el proyecto antes de enviarlo?			Si	No
	Si lo probaste, ¿encontraste errores?				
	Encontró errores y no hizo nada	Encontró errores y pregunto	Encontró errores, revisó y preguntó		
Generalización	¿Empleaste programación de otro proyecto?				
	No reutilizo código	Reutilizó código encontrado en Google o de un compañero	Reutilizó código encontrado en Scratch	Reutilizó código de un proyecto propio	
Evaluación	¿Cómo fue tu compromiso en el desarrollo del proyecto?				
	Excelente	Bueno	Regular	Malo	Nulo
Evaluación	¿Cómo fue el compromiso de tu compañero?				
	Excelente	Bueno	Regular	Malo	Nulo

Las preguntas de depuración se dirigen a conocer si el estudiante prueba su código, identifica los errores y es capaz de solucionarlos. La pregunta de generalización es acerca de la reutilización de código, con el fin de recoger datos de la capacidad del estudiante en reconocer que una solución se parece a otra.

La evaluación se afronta desde tres (3) perspectivas; (i) lo que el estudiante piensa de la solución entregada, (ii) lo que piensa de su desempeño, (iii) y en caso de contar con uno, lo que piensa del desempeño de su compañero. Las dos primeras van dirigidas a la autoevaluación y la última a la evaluación de pares.

Fase de ejecución

La fase de ejecución consiste en la implementación de la evaluación en el aula de clase. En esta fase, se les presenta a los estudiantes las tareas que el docente define

4. Modelo de evaluación basada en evidencia

en la fase de planeación; incluye el proceso desde que el estudiante inicia la sesión hasta que él obtiene realimentación de acuerdo a su desempeño, esta información se le entrega al docente como resultados para ser analizados, a través de tablas y gráficas estadísticas. Esta fase se compone por dos módulos: estadístico y del estudiante, la figura 9 muestra el diagrama de esta fase y las partes que componen sus módulos.

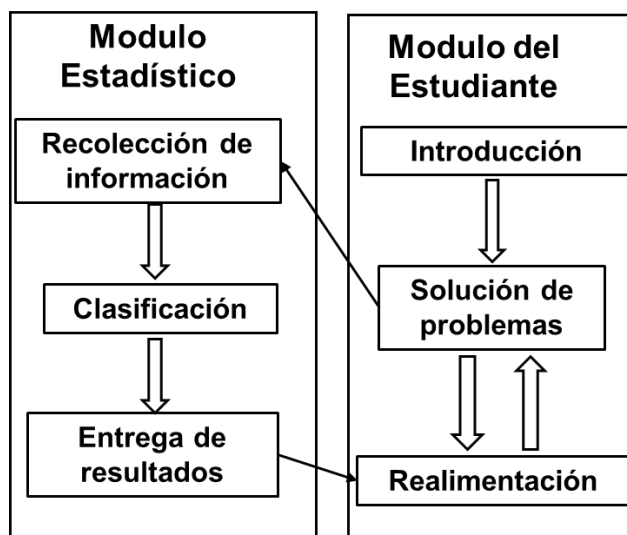


Figura 9. Fase de ejecución

Las actividades de la fase de ejecución deben ser realizadas por los estudiantes principalmente. El docente debe estar soportando el proceso de todos los estudiantes en el aula, es el espacio donde interactúa con los estudiantes mientras resuelve dudas y contesta preguntas e inquietudes. Las actividades que los estudiantes realizan en esta fase son:

- Entregar la información de inicio (datos personales, etc.)
- Conocer el manejo de las herramientas a emplear en codificación y en la evaluación
- Resolver los problemas propuestos por el docentes
- Recibir la realimentación correspondiente a la relación entre el desempeño y los criterios establecidos por el docente.
- Analizar la realimentación obtenida, detectar los errores cometidos y establecer avances.

Las dos últimas actividades se realizan iterativamente durante un periodo de tiempo prolongado, que depende de los objetivos trazados por el docente.

Módulo estadístico

El módulo estadístico interactúa con: (i) la fase de planeación, de donde toma la información de los criterios mínimos, (ii) el módulo de estudiante de donde toma la evidencia a valorar y a donde entrega el resultado para la realimentación, (iii) y la fase análisis, ya que este módulo le entrega la información de manera ordenada y sistemática para que el docente pueda realizar sus análisis y determinar las acciones a seguir. Las actividades de este módulo son realizadas por el sistema de manera automática, cada componente se relaciona con uno o más actividades. Como se puede observar en la figura 9, este módulo está formado por tres (3) componentes principales: recolección de información, clasificación y entrega de resultados.

- **Recolección de información:** consiste en reunir la información que los estudiantes brindan durante la introducción y la evidencia que los estudiantes entregan. La información de evidencia se recolecta de acuerdo a lo indicado en las tablas 39 y 40, y se almacena en una base de datos para su análisis. La información entregada en la introducción corresponde a la información consignada en la Tabla b2, Anexo B.
- **Clasificación:** constituye el proceso de selección de acuerdo a la correlación entre el mapeo de la información de los criterios mínimos seleccionados por el docente con respecto a la información de evidencia recolectada. El proceso que describe este componente se detalla en la sección 4.4.
- **Entrega de resultados:** se refiere a la presentación ordenada de los resultados obtenidos y la clasificación correspondiente. Con esta información se brinda a los estudiantes una realimentación cercana a sus necesidades y al docente los resultados obtenidos por sus estudiantes de manera individual y/o grupal. Las tablas de visualización de los resultados se proponen en el Anexo B, Tabla b3.

Módulo del estudiante

4. Modelo de evaluación basada en evidencia

Este módulo toma en cuenta la información de la fase de planeación donde el docente define las tareas que debe realizar el estudiante, para enseñárselas y que el estudiante las pueda ejecutar. La evidencia del desarrollo de las tareas es entregada por los estudiantes en esta fase.

La Figura 10 muestra el módulo, presenta el proceso que siguen los estudiantes: primero deben conocer las herramientas que se van a emplear, cómo es su manejo, tanto para la programación como para realizar la evaluación, además, en este punto del proceso los estudiantes brindan información personal (nombre, edad, grado, género) y de autoeficacia referente al manejo de computadores. Más adelante, los estudiantes inician la solución de problemas en Scratch, los entienden y resuelven para después enviarlos a ser calificados, y finalmente recibir una realimentación, ésta se genera de acuerdo a los criterios y/o estrategias de evaluación que el docente considera en cada problema. La realimentación constituye una de las partes más importantes de la evaluación [39], por esta razón se debe garantizar que se brinde a cada estudiante una realimentación adecuada y significativa, con el propósito de contribuir con el mejoramiento de su desempeño.

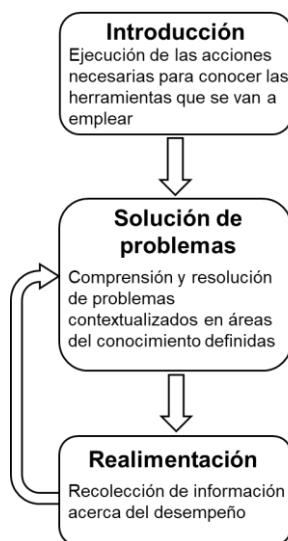


Figura 10. Módulo del estudiante

Como se muestra en la figura 10, el proceso de solución de problemas y realimentación es iterativo, se realiza para cada tarea, los estudiantes desarrollan muchas tareas consecutivamente y de cada una el estudiante obtiene la realimentación

correspondiente. Dentro de este módulo, el ciclo de solución de problemas y realimentación también se repite durante todo un periodo académico, la introducción no se repite hasta que las herramientas o información brindada por los estudiantes varíen. La figura 11 muestra el módulo del estudiante en el tiempo, donde cada momento representa una sesión, clase o fracción de tiempo (hora) en el aula de clase.

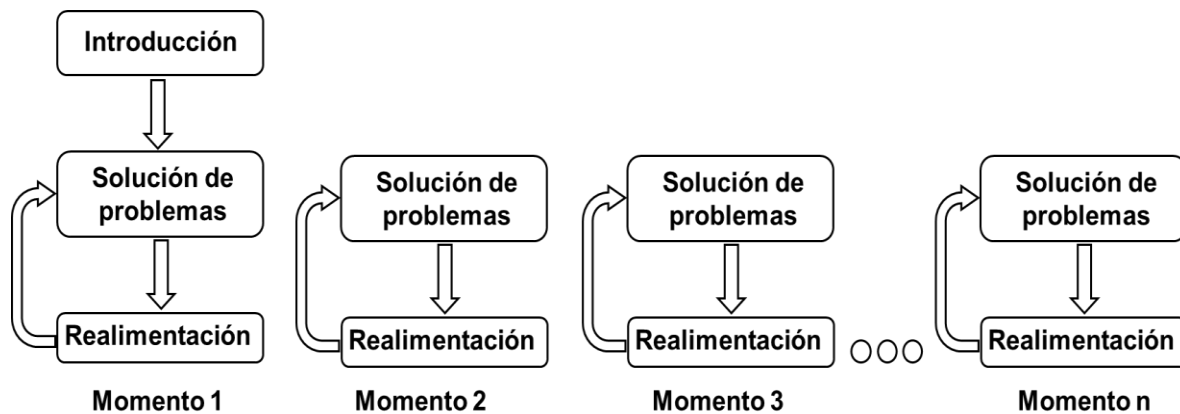


Figura 11. Módulo del estudiante en el tiempo

Fase de análisis

La fase de análisis consiste en la revisión de los resultados por parte del docente con el objetivo de reconocer dificultades y determinar mejoras tanto en la enseñanza como la evaluación.

Los resultados son entregados después de que se realiza la automatización de la evaluación y el proceso de clasificación. Dentro de los resultados se evidencia el desempeño general del grupo o el de un estudiante, en relación a los criterios mínimos de evaluación para estimar el desarrollo de CT y a la clasificación del estudiante en cuanto al cumplimiento de los criterios mínimos seleccionados por el docente.

Con el fin de detectar las dificultades en los estudiantes, fallas en el proceso de evaluación y proponer las mejoras, los docentes deben observar y examinar muy bien los resultados; ya que estos evidencian el cumplimiento de los criterios mínimos, lo que da la información necesaria para conocer quienes requieren atención especial, porque se encuentran estancados en algún concepto de CT ya que no lo conocen o

4. Modelo de evaluación basada en evidencia

entienden. También, los resultados indican el desempeño general de los estudiantes en relación al CT, con esta información los docentes pueden conocer quienes demuestran haber adquirido habilidades y quiénes no. Las soluciones y nuevas propuestas que el docente determine las puede definir en la siguiente iteración de la fase de planeación. Las actividades de esta fase son realizadas por el docente, estas incluyen:

- Observar los resultados generales
- Revisar los resultados individuales de los casos identificados como los más interesantes a juicio del docente, ya sea porque observó un alto o bajo desempeño.
- Reconocer las fallas presentadas
- Determinar las posibles mejoras

4.4. Clasificación

Dentro de la fase de ejecución en el módulo estadístico se realiza la clasificación. Ésta permite catalogar los estudiantes en clases definidas, determinando la asociación entre la evidencia entregada por cada estudiante y los criterios mínimos seleccionados por el docente para la evaluación. A continuación, se presenta la metodología empleada en este proceso y la descripción de la implementación de cada una de sus fases.

4.4.1. Metodología

Para la clasificación se siguió la metodología de extracción de datos CRISP-DM [135]. Esta metodología se describe en términos de las siguientes fases:

- **Comprensión del negocio:** consiste en el entendimiento de los objetivos del proyecto y sus requisitos. En esta fase inicial se hace una revisión sobre el contexto, los objetivos y el plan para alcanzarlos.
- **Comprensión de la información:** esta fase incluye un proceso de familiarización con los datos, con el fin de detectar los problemas de

calidad de los datos, descubrir las primeras perspectivas de los datos o detectar subconjuntos.

- **Preparación de la información:** abarca las actividades necesarias para formar el conjunto de datos: *data set*: selección de datos, limpieza y formateo.
- **Modelado:** en esta fase se seleccionan y aplican diversas técnicas de modelado. y sus parámetros se calibran a valores óptimos.
- **Evaluación:** esta fase incluye la construcción de uno o varios modelos que cumplan con una alta calidad de acuerdo a la evaluación de los mismos.
- **Despliegue:** consiste en la aplicación del modelo propuesto en la fase de evaluación.

Cada fase se compone por una serie de actividades que entregan unos resultados necesarios para las siguientes actividades y fases. Como CRISP-DM es una metodología ampliamente implementada en contextos empresariales, ha definido algunas actividades dentro de las fases que tienen en lugar específicamente en esos contextos. En el estudio actual, al tratarse de un contexto educativo algunas de estas actividades no tienen pertinencia ni relación con lo que se está realizando, razón por la cual se excluyen del estudio.

4.4.2. Contexto de minería de datos

Antes de iniciar con la descripción de la implementación de cada una de las fases, es necesario determinar el contexto de minería de datos. Para lo que se define: (i) el dominio de aplicación, que se refiere a la definición del área específica en la que se lleva a cabo el proyecto de minería de datos, (ii) tipo de problema de minería de datos, hace referencia a la clase específica de objetivo del proyecto, (iii) aspectos técnicos, abarca los problemas particulares que describen desafíos técnicos, y (iv) herramienta y técnica, hace referencia a la técnica de minería de datos que se va a emplear.

- (i) **Dominio de aplicación:** Proceso de evaluación escolar. La minería de datos tiene lugar en este proyecto ya que facilita al docente a acceder a los

4. Modelo de evaluación basada en evidencia

niveles de rendimiento y encontrar los estudiantes que necesiten atención adicional.

- (ii) **Tipo de problema de minería de datos:** Clasificación. La clasificación tiene en cuenta el conjunto de características de cada objeto con el fin de determinar a qué clase pertenece [135]. Para nuestro caso particular, se requiere clasificar al estudiante, teniendo en cuenta el mapeo entre la evidencia entregada por él y los criterios mínimos de evaluación establecidos por el docente.
- (iii) **Aspectos técnicos:** No han sido encontrados hasta el momento.
- (iv) **Herramienta y técnica:** Redes Bayesianas. Se seleccionó esta técnica debido a que ha sido ampliamente recomendada en evaluaciones basadas en evidencia [136], [137], [138], [139].

4.4.3. Comprensión del negocio

Esta fase se orienta al entendimiento del contexto y los objetivos del proyecto. Se compone por: (i) la determinación de objetivos del negocio, se refiere al entendimiento del contexto, los objetivos del negocio y criterios de éxito del negocio, (ii) valoración de la situación, se refiere a la revisión y estimación de los recursos, requisitos, restricciones, riesgos, costos y otra información relevante, (iii) determinación de los objetivos de minería de datos y determinación de los criterios de éxito, y (iv) realización del plan del proyecto, herramientas, equipo y técnicas.

- (i) La información del proceso de determinación de objetivos del negocio se puede verificar en el Capítulo 1. El contexto en la sección 1.1, objetivos y criterios de éxito secciones 1.4 y 1.5.
- (ii) La valoración de la situación se puede verificar en los Capítulos 2 y 3. Donde se tienen en cuenta el marco teórico y estado del arte que delimitan el estudio, además de la caracterización que facilitó el entendimiento de los elementos a incluir en la evaluación.

- (iii) El objetivo de minería de datos es: clasificar a cada estudiante en una categoría teniendo en cuenta la evidencia entregada por él en correlación con los criterios mínimos seleccionados por el docente.
- Los criterios de éxito están relacionados con la tasa nula (null rate), la exactitud de clasificación, el error absoluto medio (MAE: Mean Absolute Error) y el coeficiente kappa estadístico. La tasa nula brinda información de referencia básica para determinar los valores base, se recomienda verificar esta información antes de elegir un clasificador sofisticado, en este estudio tiene lugar la tasa Naive relacionada con la técnica Naive Bayes. La exactitud de clasificación está determinada por el número correcto de instancias clasificadas, se ha determinado que los modelos deberán cumplir con un porcentaje igual al 90% o mayor. El MAE corresponde al promedio de todos los errores absolutos, razón por la cual se requiere que este valor sea bajo, se ha definido un valor máximo de 0,05. El coeficiente kappa estadístico o coeficiente de Cohen mide la relación entre dos evaluadores, cada uno de los cuales clasifica n artículos en C categorías mutuamente excluyentes, se concreto un valor superior a 0,8 como criterio de éxito.
- (iv) El plan del proyecto de minería de datos especifica los pasos a seguir dentro del proyecto, estos se encuentran descritos a continuación, sin embargo la tabla del plan del proyecto se puede apreciar en el anexo B, tabla b6 donde cada fase se relaciona con sus actividades y estas a su vez con los requisitos necesarios para realizarlas y los resultados que entregan.

4.4.4. Comprensión de la información

La comprensión de la información hace referencia a las actividades que facilitan la recolección de datos y la familiarización con estos teniendo en cuenta los objetivos del negocio. Se relaciona con las siguientes actividades: (i) recopilación inicial de datos, se refiere a adquirir dentro del proyecto toda la información necesaria para alcanzar los objetivos, (ii) descripción de los datos, se trata de examinar las propiedades de los datos y generar un reporte acerca de ellas y las relaciones encontradas, (iii) exploración de los datos, abarca lo relacionado con las cuestiones de minería de datos que pueden realizarse empleando consultas o visualización, con el objetivo de

4. Modelo de evaluación basada en evidencia

contribuir a refinar la descripción de los datos y los informes de calidad, ayudando a transformar o a preparar los datos para un mayor análisis, y (iv) verificación de calidad de datos, comprende la revisión de los datos para comprobar el cubrimiento de los casos requeridos, los errores y los valores faltantes.

- (i) La recopilación inicial de datos incluye: (a) información del estudiante (datos personales e información de autoeficacia acerca del uso de los computadores) (Anexo B, tabla b2) , (b) resultados de acuerdo a la evidencia entregada en la resolución de problemas (tabla 40), (c) información acerca de las prácticas de programación de acuerdo a la información adquirida a través de la encuesta (tabla 41), (d) criterios mínimos seleccionados por el docente para cada problema y (e) la valoración del docente, que corresponde a un valor numérico que es determinado de acuerdo a la observación directa sobre cada estudiante en el aula de clase.

- (ii) La descripción de los datos brinda información acerca de todos los datos recolectados en la recopilación inicial de datos. La información que se brinda acerca de cada variable corresponde a: la denominación de la variable, el tipo de dato (numérico, binario, cadena de caracteres, etc.), el rango en el que se encuentra (de acuerdo al tipo de dato), las opciones correspondientes a cada valor del rango, el tipo de información (personal, evidencia, etc.) y descripción (da más información acerca de la variable para aclarar a que se refiere específicamente). La descripción de los datos entregados por los estudiantes se puede visualizar en el Anexo B, tabla b7 y la descripción de los datos brindados por el docente está en el mismo anexo tabla b8.

- (iii) En la exploración se determinó con respecto a los datos recolectados realizar las siguientes acciones:
 - 1. Eliminar los campos: nombre y apellido, e-mail, contraseña y usuario Scratch, en lugar de esta información se dará un identificador para cada estudiante.
 - 2. Reemplazar las variables de evidencia entrega por el estudiante para cada ejercicio por el valor de correlación entre ésta evidencia y los criterios mínimos. Como la evidencia entregada por el estudiante tiene

un gran volumen, se determina tomar una medida de correlación en cada caso. A través de la correlación entre la evidencia entregada por el estudiante y los criterios mínimos seleccionados por el docente para cada problema.

Con el fin de facilitar la clasificación, para cada problema resuelto, se halla el coeficiente de correspondencia lineal de acuerdo a la ecuación 1. Esta es una medida lineal entre dos variables aleatorias, en este caso “x” corresponde a los criterios mínimos seleccionados por el docente y “y” a la evidencia entregada por el estudiante (resultados de los problemas resueltos).

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

Ecuación 1. Coeficiente de correlación lineal

Donde σ_{xy} es la covarianza de “x” y “y”, σ_x es la desviación estándar de la variable “x” y σ_y es la desviación estándar de la variable “y”.

- Determinar la variable: Valoración docente como la CLASE para el proceso de clasificación que se lleva a cabo.

Como resultado se obtiene la tabla 42.

Tabla 42. Variables seleccionadas tras el proceso de exploración de datos

Variable	Tipo de dato	Rango	Tipo de información
Identificador del estudiante	Cadena de caracteres	N.A.	Información de identificación
Edad	Numérico	8-18	Información personal
Género	Binario	0-1	Información personal
Grado	Numérico	3-11	Información personal
Materia favorita	Numérico	0-8	Información personal
Actividad favorita del PC	Numérico	0-8	Información personal
Conocimiento del PC	Numérico	0-10	Pregunta de autoeficacia
Manejo PC	Numérico	0-10	Pregunta de autoeficacia

4. Modelo de evaluación basada en evidencia

Frecuencia Uso PC	Numérico	0-3	Pregunta de autoeficacia
Correlación_ejercicio_1	Numérico	(-1) -(1)	Correlación entre la evidencia y los criterios mínimos
Correlación_ejercicio_2	Numérico	(-1) -(1)	Correlación entre la evidencia y los criterios mínimos
...			
Correlación_ejercicio_n	Numérico	(-1) -(1)	Correlación entre la evidencia y los criterios mínimos
Evaluación1	Numérico	0-4	Evaluación de la solución
Depuración	Numérico	0-3	Pregunta acerca de prácticas de depuración
Generalización	Numérico	0-3	Pregunta acerca de prácticas de generalización
Evaluación2	Numérico	0-4	Autoevaluación
Evaluación3	Numérico	0-4	Evaluación de pares
Valoración docente	Numérico	0-5	CLASE

- (iv) La verificación de calidad de datos se encarga de asegurar la integridad y exactitud de los datos, los resultados se encuentran consignados en la tabla 43, ésta se obtuvo teniendo en cuenta que:
1. La variable materia favorita carece de opciones como ciencias naturales, física y química que no se tuvieron en cuenta.
 2. No todos los estudiantes han alcanzado el mismo número de problemas resueltos, lo que quiere decir que en algunos casos no todos asisten a las sesiones, algunos aunque asisten se encuentran atrasados con respecto a la media y otros adelantados, por lo tanto en el momento de realizar la correlación se debe:
 - Considerar el número de sesiones a las que asiste cada estudiante. Se notó que es necesario agregar un campo en el que

se indique el número de sesiones que los estudiantes han asistido al aula a desarrollar soluciones, este dato es importante al momento de comparar el desempeño de estudiantes que no han asistido a todas las sesiones y tienen un buen desempeño, con el desempeño de los demás.

- Para los propósitos de este estudio, se considera que si alguna evidencia entregada supera los criterios mínimos, los está cumpliendo y por lo tanto la correlación es positiva y lineal.
3. La variable evaluación 1 corresponde a la autoevaluación de la solución entregada, fue calificada con la mayor valoración por todas las instancias.
 4. La variable evaluación 3 se refiere a la evaluación de pares, para algunas instancias carece de valores en sus campos debido a que no todos los estudiantes respondieron esta pregunta, ya sea porque no contaron con un compañero de trabajo o porque tomaron la elección de no hacerlo, razón por la cual esta variable ha sido eliminada.
 5. No se encontraron instancias de la variable: valoración del docente, con valores menores a 2.

Tabla 43. Variables seleccionadas tras el proceso de verificación de la calidad de los datos

Variable	Tipo de información
Usuario Id	Identificador del estudiante
Edad	Información personal
Genero	Información personal
Grado	Información personal
No. de sesiones	Información de asistencia a las sesiones
Actividad favorita del PC	Información personal
Conocimiento del PC	Pregunta de autoeficacia
Manejo PC	Pregunta de autoeficacia
Frecuencia Uso PC	Pregunta de autoeficacia

4. Modelo de evaluación basada en evidencia

Correlación_ejercicio_1	Correlación entre la evidencia y los criterios mínimos
Correlación_ejercicio_2	Correlación entre la evidencia y los criterios mínimos
...	
Correlación_ejercicio_n	Correlación entre la evidencia y los criterios mínimos
Depuración	Pregunta acerca de prácticas de depuración. Pregunta de autoeficacia
Generalización	Pregunta acerca de prácticas de generalización. Pregunta de autoeficacia
Evaluación2	Autoevaluación
Valoración docente	CLASE

4.4.5. Preparación de la información

La fase de preparación de datos abarca las actividades para construir el conjunto final de datos (data set) a partir de los datos iniciales. Esta fase incluye: (i) selección de los datos, (ii) limpieza de los datos, (iii) construcción de la información, (iv) integración de los datos y (v) formateo de los datos.

- (i) La selección de los datos consistió en la aplicación de criterios de inclusión y exclusión de la información disponible. Los criterios se seleccionaron de acuerdo a la relevancia de los objetivos de extracción de datos, la calidad y las limitaciones técnicas, como los límites en el volumen de datos o los tipos de datos.
- Debido a inconvenientes técnicos que se presentaron durante el desarrollo del prototipo presentado en el Capítulo 5, algunos elementos de la tabla b7 del Anexo B se excluyeron, estos fueron: operadores anidados y envío y recepción de mensajes entre varios sprites.
 - Se incluyen los valores de correlación de todos los ejercicios, en este caso particular son 13 ejercicios.

- Se encontró que los valores de correlación de algunas instancias da como resultado un error, por lo tanto no se pueden tomar estos valores para el análisis de los datos.
 - Se elimina la columna con los valores de identificación de los estudiantes, ya que sirve de guía para conocer particularidades pero no es requerida para formar la data set.
- (ii) Una vez se selecciona la información a ser analizada se procede con la limpieza de los datos, se eliminan instancias repetidas, como estudiantes que se registraron dos veces; se reemplaza la información personal poco relevante para este estudio como el nombre, apellido, email, por un identificador para cada estudiante; al considerar que algunas variables no brindan información relevante se excluyen, como: evaluación 1 y actividad favorita del computador; se excluye también evaluación 3 debido a que algunos estudiantes no contestaron esta pregunta. Cómo no se cuentan con los valores de correlación para algunas instancias (debido a errores que se generan en los cálculos), se propone emplear el valor de la unidad menos el error cuadrático medio (1-MSE). La ecuación del MSE se muestra en la ecuación 2.

$$MSE = \frac{1}{n} \sum_{i=0}^n (X_i - X'_i)^2$$

Ecuación 2. MSE- Error cuadrático medio

- (iii) En la construcción de la información se toma toda la información previamente seleccionada y limpia. Se proponen en esta actividad dos acciones importantes:
1. Incluir la variable: No. de sesiones, se debe a que se requiere diferenciar los resultados de acuerdo al desarrollo de sesiones.
 2. Normalizar los datos del error cuadrático medio con el objetivo de facilitar la posible aplicación de modelos que lo requieran. El objetivo de la normalización es realizar un cambio en el conjunto de datos a una escala común, en este caso particular, a los valores de las columnas que

4. Modelo de evaluación basada en evidencia

corresponden a las correlaciones, sin distorsionar las diferencias en los rangos de valores.

La normalización basada en la unidad entrega los datos en valores entre 0 y 1, donde 0 corresponde al valor mínimo encontrado dentro del conjunto de datos. En este estudio, se empleó la Característica Scaling para la normalización, ésta es presentada en la ecuación 2.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Ecuación 3. Característica scaling

- (iv) El proceso de integración de datos incluyó las acciones necesarias para reunir la información proveniente de las diferentes tablas. Se tomaron los datos personales seleccionados, las respuestas de la encuesta, y la información de la correlación obtenida con el coeficiente de correlación lineal para formar la data set.
- (v) Formatear la información no fue una tarea necesaria, ya que las variables están definidas en valores numéricos. Lo que se puede apreciar en la tabla 45, que muestra las variables incluidas en la plantilla de la data set empleada, el tipo de dato, los rangos que puede tomar y el tipo de información a la que corresponde la variable.

Tabla 44. Variables de la plantilla de la data set

Variable	Tipo de dato	Rango	Tipo de información
Genero	Binario	(0) o (1)	Personal
Edad	Numérico	de 8 a 18	Personal
Grado	Numérico	de 3 a 11	Personal
No. de sesiones	Numérico	0-100	Personal
Frecuencia PC	Numérico	0-3	Autoeficacia
Conocimiento Computador	Numérico	0-10	Autoeficacia
Manejo Computador	Numérico	0-10	Autoeficacia

Actividad Favorita PC	Numérico	0-8	Personal
Ejercicio1	Numérico	de 0 a 1	Correlación normalizada
Ejercicio2	Numérico	de 0 a 1	Correlación normalizada
Ejercicio3	Numérico	de 0 a 1	Correlación normalizada
Ejercicio4	Numérico	de 0 a 1	Correlación normalizada
Ejercicio5	Numérico	de 0 a 1	Correlación normalizada
Ejercicio6	Numérico	de 0 a 1	Correlación normalizada
Ejercicio7	Numérico	de 0 a 1	Correlación normalizada
Ejercicio8	Numérico	de 0 a 1	Correlación normalizada
Ejercicio9	Numérico	de 0 a 1	Correlación normalizada
Ejercicio10	Numérico	de 0 a 1	Correlación normalizada
Ejercicio11	Numérico	de 0 a 1	Correlación normalizada
Ejercicio12	Numérico	de 0 a 1	Correlación normalizada
Ejercicio13	Numérico	de 0 a 1	Correlación normalizada
Evaluación2	Numérico	0-4	Autoevaluación
Generalización	Numérico	0-3	Autoeficacia
Depuración	Numérico	0-3	Autoeficacia
CLASE	Carácter	E, B, A, D, N	CLASE

Para la clasificación del desempeño se asignaron letras:

- E: Excelente
- B: Bueno

4. Modelo de evaluación basada en evidencia

- A: Aceptable
- D: Deficiente
- N: Nulo

4.4.6. Modelado

Esta fase hace referencia a la selección y aplicación de las técnicas de modelado que sean pertinentes al problema, aplicadas a la data set obtenido. Lo que puede implicar modificar la data set con el fin de poder aplicar algunas técnicas que tienen requerimientos específicos sobre el formato de los datos.

La fase de modelado se compone de las siguientes partes: (i) selección de la técnica de modelado, se pueden escoger varias para evaluarlas. (ii) generar el diseño de pruebas, abarca la definición los criterios que calculan la calidad de las medidas, la decisión de como dividir los datos entre datos de entrenamiento, de prueba y validación, (iii) construir el modelo, corresponde a correr el data set preparado en la herramienta de modelado para crear uno o más modelos, descripción el modelo, (iv) evaluación del modelo, consiste en listar las cualidades de los modelos generados, por ejemplo en términos de la precisión y clasificar su calidad en relación con otros.

(i) Selección de la técnica de modelado:

La técnica de modelado seleccionada es: Redes Bayesianas debido a que han sido ampliamente recomendada en evaluaciones basadas en evidencia [136], [137], [138], [139]. Sin embargo, la metodología recomienda para la clasificación: Análisis discriminante, Métodos de inducción de reglas, Árbol de decisiones de aprendizaje, Redes neuronales, K Vecino más cercano (KNN), Razonamiento basado en casos y Algoritmos genéticos. Se seleccionaran las dos más apropiadas para ser documentadas debido a la extensión de la información.

(ii) Diseño de la evaluación:

Para el diseño de la evaluación se describen los dos tipos de modelos más apropiados y se explican las pruebas a realizar sobre ellos.

1. Redes bayesianas corresponden a modelos gráficos estadísticos, constituyen un conjunto de condicionales variables y sus dependencias condicionales, que se pueden representar en un grafo acíclico dirigido (DAG: Directed Acyclic Graph).

2. Multilayer perceptron es una clase de red neuronal artificial que emplea una técnica de aprendizaje supervisado llamada backpropagation para entrenamiento. Se compone básicamente por tres (3) capas de nodos: capa de entrada, capa oculta y capa de salida. Cada nodo es una neurona que utiliza una función de activación no lineal (excepto los nodos de entrada).

Evaluación:

- La tasa nula: en este estudio tiene lugar la tasa Naive relacionada con la técnica Naive Bayes.
- La exactitud de clasificación: porcentaje igual al 90% o mayor.
- MAE : máximo 0,05.
- El coeficiente kappa: superior a 0,8 como criterio de éxito.
- Matriz de confusión
- Tiempo empleado en la construcción de un modelo

(iii) Construcción del modelo:

Para la construcción del modelado se recomienda la herramienta de minería de datos Weka de la Universidad de Waikato en Hamilton, Nueva Zelanda [140]. Weka proporciona implementaciones de algoritmos de aprendizaje que se puede aplicar fácilmente a un data set e incluye herramientas para transformar data sets.

La data set empleado se presenta en la tabla b9 del anexo B.

Una vez se abre el archivo del data set en la herramienta weka, se selecciona el filtro StringToNominal –R last para todas las pruebas, debido a que la clase es un string. Se seleccionó un valor de cross-validation de 20.

1: Redes bayesianas

4. Modelo de evaluación basada en evidencia

Esquema: weka.classifiers.bayes.BayesNet -D -Q
 weka.classifiers.bayes.net.search.local.K2 -- -P 5 -mbc -S BAYES -E
 weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
 Relación: Dataset-
 weka.filters.unsupervised.attribute.StringToNominal-Rlast
 Parámetros:

BatchSize: 100
 Debug: False
 soNotCheckCapabilities: False
 Estimator: SimpleEstimator -A 0.5
 numDecimalPlaces: 2
 searchAlgorithm: K2 -P 5 -mbc-S BAYES
 initAsNaiveBayes: True
 markovBlanketClassifier: True
 maxNrOfParents: 5
 randomOrder: False
 scoreType: bayes
 use ADTree: False

Resultados:

En el grafo encontrado en el Anexo B figura b10, se puede ver la relación entre los resultados de correlación de los diferentes ejercicios, los primeros seis tienen una relación con los anteriores, lo que se debe a que muchas instancias de los ejercicios siguientes, no tienen valores.

Grafo resultante: Anexo B figura b1.

Errores de clasificación: Anexo B figura b2.

2: Redes Neuronales

weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500
 -V 0 -S 0 -E 20 -H a
 Relation: Dataset-
 weka.filters.unsupervised.attribute.StringToNominal-Rlast
 Parámetros:

GUI: False

autoBuild:True
batchSize: 100
debug: False
decay: False
doNotCheckCapabilities: False
hiddenLayers:a
learningRate: 0.3
momentum:0.25rt/6y7nominalToBinaryFilter: True
normalizeAttributes: True
normalizeNumericCass: True
numDecimalPlaces:2
reset: True
seed: 0
trainingTime:500
validationSetSize:0
validationThreshold: 20

Resultados: Anexo B tabla b11

Errores de clasificación: Anexo B figura b2.

(iv) Evaluación del modelo:

La evaluación del modelo se relaciona con los criterios de éxito:

- Tasa nula (null rate):

Esquema: weka.classifiers.bayes.NaiveBayes

Relación: Dataset-weka.filters.unsupervised.attribute.StringToNominal-RIlast

Parámetros: se emplearon los parámetros por defecto, estos son:

- batchSize: 100
- debug: False
- numDecimalPlaces: 2
- useKernelEstimator:False
- useSupervisedDiscretization: False

Descripción: La descripción completa se puede apreciar en la tabla b12 del anexo B.

4. Modelo de evaluación basada en evidencia

Correctly Classified Instances	212	80.916 %
Incorrectly Classified Instances	50	19.084 %
Kappa statistic	0.7556	
Mean absolute error	0.081	
Time taken to build model: 0 seconds		

El porcentaje de instancias clasificadas correctamente, MAE y coeficiente Kappa no cumplen con el criterio establecido. Por lo tanto, no se requiere verificar este criterio de éxito.

- Exactitud de clasificación:

Redes bayesianas:

Correctly Classified Instances	239	91.221 %
Incorrectly Classified Instances	23	8.778 %

Redes neuronales:

Correctly Classified Instances	240	91.603 %
Incorrectly Classified Instances	22	8.396 %

- MAE:

Redes bayesianas:

Mean absolute error	0.043
---------------------	-------

Redes neuronales:

Mean absolute error	0.044
---------------------	-------

- Coeficiente kappa estadístico:

Redes bayesianas:

Kappa statistic	0.8919
-----------------	--------

Redes neuronales:

Kappa statistic 0.8875

- Matriz de confusión

Redes bayesianas:

```
a b c d e <-- classified as
41 3 0 0 0 | a = E
5 61 3 1 0 | b = B
3 5 61 0 1 | c = A
0 0 2 52 0 | d = D
0 0 0 0 24 | e = N
```

Redes neuronales:

```
a b c d e <-- classified as
35 9 0 0 0 | a = E
2 63 4 1 0 | b = B
0 2 65 3 0 | c = A
0 0 0 53 1 | d = D
0 0 0 0 24 | e = N
```

- Tiempo:

Redes bayesianas:

Time taken to build model: 0.28 seconds

Redes neuronales:

Time taken to build model: 4.39 seconds

4.4.7. Evaluación

La evaluación se realiza con el fin de alcanzar calidad suficiente desde una perspectiva de análisis de datos. Consiste en la revisión a fondo del modelo antes del despliegue; primero se compara el modelo con los objetivos del negocio, después se revisa el proceso y finalmente se obtiene una decisión acerca de la aplicación de los datos

4. Modelo de evaluación basada en evidencia

resultantes del proceso de análisis. Esta fase se compone de las actividades: (i) evaluación de los resultados, (ii) revisión del proceso y (iii) establecimiento de las acciones siguientes.

- (i) Evaluación de los resultados: de acuerdo a los resultados presentados en la evaluación del modelo, se puede decir que las Redes Neuronales satisfacen más unos criterios y las Redes Bayesianas satisfacen más los otros criterios establecidos para la selección de un modelo.
La exactitud de la clasificación, el MAE y el coeficiente kappa tienen valores más adecuados empleando el modelo de Redes Neuronales, sin embargo viendo la matriz de confusión se puede notar que clasifica mucho mejor a las instancias que pertenecen a las clases más altas, lo que brinda mayor garantía en el momento de clasificar estudiantes que de acuerdo a la estimación de su desarrollo de CT tienen más ventajas que los otros. Además en términos de eficiencia se puede notar que esta opción toma menos tiempo para probar cada modelo, lo que consume menos recursos. Los resultados muestran que en los dos casos existe una relación entre las diferentes variables, sin embargo las redes bayesianas hacen una diferencia de relaciones desde el ejercicio 6, que marca la diferencia entre las instancias que tienen valores para todos los ejercicios (que representan a quienes fueron a todas las sesiones) y los que no asistieron a todas.
- (ii) Revisión del proceso: en múltiples ocasiones el proceso fue revisado, se hicieron cambios en el data set, se retomaron fases de entendimiento y preparación de información en varias oportunidades. La información consignada representa las primeras y últimas iteraciones con la información y el manejo de la misma.
- (iii) Establecimiento de las acciones siguientes: como esta fase se realizó de manera iterativa, mientras se probaron distintos modelos, finalmente se tomó seleccionada la Red Bayesiana y para concluir se estableció el despliegue como paso siguiente.

Las Redes Bayesianas [141] representan un modelo gráfico para relacionar probabilidades en un conjunto de variables, manejan sin dificultar conjuntos de datos con información faltante, además con las técnicas estadísticas bayesianas permiten relacionar conocimiento de dominio y de información.

El teorema de Bayes brinda una forma de predecir la probabilidad de una hipótesis A_i , teniendo en cuenta la evidencia presentada (B). La ecuación 4 muestra la fórmula que rige el teorema de Bayes.

$$P(A_i|A) = \frac{P(A|A_i) P(A_i)}{P(A)}$$

Ecuación 4. Teorema de Bayes

$P(A)$ se denomina la constante de normalización porque es la probabilidad de la evidencia bajo cualquier hipótesis.

$P(A_i)$ es la probabilidad de la hipótesis antes de tener la evidencia, se denomina probabilidad anterior.

$P(A | A_i)$ se llama probabilidad y representa la probabilidad de la evidencia dado que la hipótesis se dio.

$P(A_i | A)$ es lo se va a calcular, la probabilidad de la hipótesis después de ver la evidencia, denominada "posterior".

4.4.8. Despliegue

La fase de despliegue corresponde a la integración del modelo para que pueda ser usado. Dependiendo de los requisitos, esta fase puede ser tan simple como la generación de un informe o tan compleja como la realización periódica de un proceso de análisis de datos. La fase se compone de las actividades: (i) planificación de despliegue, (ii) planificación de la monitorización y del mantenimiento, (iii) generación de informe final y (iv) revisión del proyecto.

- (i) Planificación de despliegue: el estudio presente incluye la implementación del modelo resultante, en el Capítulo 5 se encuentra la descripción del prototipo.
- (ii) Planificación de la monitorización y del mantenimiento, se realiza el monitoreo después de la implementación y evaluación de la implementación, esta actividad no se completa a cabalidad debido a que se encuentra por fuera de los límites del estudio por la duración del mismo.

4. Modelo de evaluación basada en evidencia

- (iii) Generación de informe final, dentro del desarrollo de las actividades de las fases previas, se encuentra la información del proyecto, lo contenido en la sección 4.3 corresponde al informe final.
- (iv) Revisión del proyecto, actividad realizada de la que surgió el presente estudio como resultado final.

4.5. Conclusiones

Los instrumentos y técnicas de evaluación del aprendizaje deberían estar de acuerdo con el enfoque educativo definido por entes internacionales como como el Ministerio de Educación Nacional [20]. Se requiere contar con soluciones que consideren el proceso evaluativo no limitado a la medición, que se propicien espacios de autoevaluación, realimentación y seguimiento en tiempo real, que permitan la detección temprana de dificultades en los estudiantes y que estime los elementos formativos, de acuerdo a los conocimientos, habilidades, actitudes de cada estudiante.

Con el fin de que los docentes puedan incluir las prácticas de programación en el aula de clase, es importante que ellos herramientas que los soporten en los procesos de enseñanza-aprendizaje y evaluación, que se facilite el tedioso trabajo de calificar manualmente y se brinden resultados confiables y válidos; el modelo propuesto guía a los docentes a planear sus evaluaciones de acuerdo a las características mencionadas.

La evaluación de CT consiste en una estimación, ya que a través de los instrumentos y mecanismos solo se pueden conocer aspectos de algunos elementos. Además, no es suficiente un único mecanismo de evaluación, al emplear varios se tiene un mayor entendimiento del aprendizaje del estudiante y su proceso de desarrollo.

Estimar el desarrollo de CT no es una tarea simple, sin embargo después del análisis de su definición, elementos y estrategias de evaluación, se logra comprender como y a través de que evidencia los estudiantes entregan información suficiente para llegar a una estimación lo más próxima a la realidad.

El proceso de clasificación requirió de un continuo manejo de la información, las fases de comprensión y preparación de la información se realizaron varias veces, con el fin de refinar los datos para ser analizados de una manera más sencilla.

Para la clasificación se probaron diversos modelos, los modelos presentados correspondieron a los más apropiados para la solución de acuerdo a los objetivos planteados y las necesidades del estudio.

Capítulo 5

5. HERA-CT: Herramienta de Evaluación y Realimentación para el Aprendizaje – CT

Teniendo en cuenta el modelo de evaluación basado en evidencia descrito, que define un proceso en el que se consideran cuatro (4) fases: iniciación, planeación, ejecución y análisis. Se describe a continuación, el diseño e implementación de un prototipo software basado en las especificaciones determinadas por el modelo propuesto. El prototipo se denomina HERA – CT (Herramienta de Evaluación y Realimentación para el Aprendizaje – Pensamiento Computacional), su propósito es estimar el desarrollo de CT en estudiantes y facilitar el proceso de evaluación a los docentes. En el capítulo se describe el proceso de desarrollo y se brinda información acerca de las etapas que se siguieron: (i) análisis, (ii) diseño, (iii) implementación y (iv) despliegue del prototipo.

5.1. Análisis del prototipo

El análisis se realiza con el objetivo de identificar las necesidades que definen el desarrollo del prototipo. Se recoge la información necesaria para realizar el diseño, como definición del prototipo, identificación de requisitos y casos de uso.

5.1.1. Definición del prototipo

Se requiere una herramienta de evaluación automatizada que facilite el proceso de evaluación para estimar el desarrollo de CT y brinde a los estudiantes realimentación.

La evaluación tiene lugar en contextos educativos escolares, dónde a través de prácticas de programación en Scratch los estudiantes entregan la evidencia necesaria para estimar el desarrollo de la competencia. El diseño debe facilitar a los docentes los procesos de evaluación, considerando el modelo basado en evidencia descrito, que especifica fases de guía.

Para la fase de iniciación, se propone aportar la gestión de cursos, dónde los docentes puedan crear cursos, incluir un nombre, área del conocimiento y objetivo general, información que sirve de guía para los estudiantes y a otros docentes que puedan requerir ideas para sus futuros cursos.

Para la fase de planeación, la herramienta debe proveer opciones para incluir los problemas a ser resueltos por los estudiantes y determinar los criterios mínimos para estimar el desarrollo de CT.

Para la fase de ejecución se requiere que el prototipo soporte el procesamiento y análisis de código en Scratch, la recolección de información del formulario de ingreso y la encuesta con preguntas de autoeficacia y autoevaluación, para facilitar la clasificación de los estudiantes y la entrega de resultados a los docentes acerca del desempeño de los estudiantes con respecto a los elementos de CT. Además, se requiere que proporcione realimentación efectiva a los estudiantes una vez terminen con la solución de cada problema.

Finalmente, para la fase de análisis el prototipo debe mostrar a los docentes los resultados obtenidos, facilitar su visualización y soportar diferentes opciones para poder observar, comparar y tomar decisiones con facilidad.

5.1.2. Identificación de requisitos

Los requisitos identificados fueron definidos teniendo en consideración los detalles definidos en el modelo basado en evidencia descrito en el Capítulo 4. Para la identificación de los requisitos se define una plantilla presentada en la tabla 45, los requisitos funcionales se encuentran identificados en las tablas 46 a 54 y los requisitos no funcionales en las tablas 55 a 57.

Tabla 45. Plantilla de identificación de requisitos funcionales y no funcionales

Nombre e Identificador	
Descripción	
Necesidad	
Prioridad	
Verificabilidad	
Estabilidad	

- **Nombre e Identificador:** información para reconocer cada requisito, el identificador tiene el formato: RQF## y RQNF##, dónde:
 - RQF: Requisito funcional.
 - RQNF: Requisito no funcional.
 - ##: Número de identificación del requisito.
 - **Descripción:** breve descripción del requisito.
 - **Necesidad:** indica el nivel de prioridad de inclusión del requisito en el sistema.
 - **Prioridad:** indica el grado de importancia del orden de implementación, determinada entre los siguientes niveles: Alto, Medio y Bajo.
 - **Verificabilidad:** indica el grado de comprobación de su implementación de forma correcta, determinada entre los siguientes niveles: Alto, Medio y Bajo.
 - **Estabilidad:** determina si el requisito está sujeto o no a modificaciones.
- **Requisitos Funcionales**

Tabla 46. Requisito funcional RQF01

Nombre e Identificador	Registro de usuarios – RQF01
Descripción	El prototipo debe permitir a los usuarios registrarse de acuerdo a su rol: docente o estudiante.

Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 47. Requisito funcional RQF02

Nombre e Identificador	Ingreso a usuarios – RQF02
Descripción	El prototipo debe permitir el ingreso al sistema a los usuarios.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 48. Requisito funcional RQF03

Nombre e Identificador	Creación de cursos – RQF03
Descripción	El prototipo debe permitir a los docentes crear cursos e incorporar información acerca del área de conocimiento y objetivo.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 49. Requisito funcional RQF04

Nombre e Identificador	Creación de problemas – RQF04
-------------------------------	-------------------------------

5. HERA-CT

Descripción	El prototipo debe permitir a los docentes crear problemas del área de conocimiento de su elección.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 50. Requisito funcional RQF05

Nombre e Identificador	Determinación de criterios mínimos – RQF05
Descripción	El prototipo debe permitir a los docentes determinar los criterios mínimos a evaluar en cada uno de los problemas.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Media
Estabilidad	Durante toda la vida del software

Tabla 51. Requisito funcional RQF06

Nombre e Identificador	Presentación de resultados – RQF06
Descripción	El prototipo debe mostrar a los docentes los resultados de desempeño de sus estudiantes y la clasificación en la que se encuentran.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 52. Requisito funcional RQF07

Nombre e Identificador	Presentación de problemas – RQF07
Descripción	El prototipo debe permitir a los estudiantes visualizar los problemas que van a resolver.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 53. Requisito funcional RQF08

Nombre e Identificador	Calificación automática – RQF08
Descripción	El prototipo debe realizar el análisis y procesamiento de código Scratch y la clasificación con respecto a la correspondencia de la solución con respecto a los criterios mínimos establecidos.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Media
Estabilidad	Durante toda la vida del software

Tabla 54. Requisito funcional RQF09

Nombre e Identificador	Visualización de realimentación – RQF09
Descripción	El prototipo debe brindar a los estudiantes la realimentación de acuerdo a la evidencia entregada.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta

Estabilidad	Durante toda la vida del software
--------------------	-----------------------------------

- **Requisitos No Funcionales**

Tabla 55. Requisito no funcional RQNF01

Nombre e Identificador	Aplicación web – RQNF01
Descripción	El prototipo debe ser una aplicación web. A la que los usuarios puedan ingresar a través de un navegador web.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Alta
Estabilidad	Durante toda la vida del software

Tabla 56. Requisito no funcional RQNF02

Nombre e Identificador	Interfaz simple e intuitiva – RQNF02
Descripción	El prototipo debe presentar una interfaz fácil de usar tanto para los docentes como los estudiantes, con el propósito de que el usuario pueda interactuar de una manera y cómoda y agradable.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Media
Estabilidad	Durante toda la vida del software

Tabla 57. Requisito no funcional RQNF03

Nombre e Identificador	Fiabilidad en la evaluación – RQNF03
-------------------------------	--------------------------------------

Descripción	El prototipo debe garantizar que para condiciones iguales, donde se entrega una misma evidencia y se evalúen los mismos parámetros, se debe conservar el resultado.
Necesidad	Esencial
Prioridad	Alta
Verificabilidad	Media
Estabilidad	Durante toda la vida del software

5.1.3. Casos de uso

Los casos de uso describen todas las posibles interacciones entre los usuarios del prototipo con el fin de alcanzar un objetivo. Se proporcionan diversos escenarios que indican la sucesión de respuestas surgidas ante las acciones iniciadas por los usuarios del prototipo denominados actores. Los actores son quienes desarrollan diferentes acciones sobre el sistema, para el prototipo corresponden a los docentes y estudiantes.

La tabla 58 muestra la plantilla utilizada para la descripción textual de los casos de uso y la nomenclatura utilizada:

Tabla 58. Plantilla de descripción textual de los casos de uso

Nombre e Identificador	
Objetivo	
Actor	
Precondiciones	
Postcondiciones	
Escenario inicial	

- **Nombre e Identificador:** información para reconocer cada caso de uso, el identificador tiene el formato: CUE#.#, dónde:

5. HERA-CT

- CU: Caso de Uso.
 - E#: Escenario No. Los escenarios se proponen de acuerdo a las fases del modelo para el caso del actor docente (E1: Fase de iniciación, E2: Fase de planeación, E3: Fase de análisis) y al módulo del estudiante en el caso del actor estudiante (E4: Introducción, E5: Solución de problemas, E6: Realimentación). Se adiciona el escenario 0: (E0: Fase 0), para representar el registro y autenticación de los docentes.
 - #: Número de identificación del caso de uso.
- **Objetivo:** breve descripción del objetivo.
 - **Actor:** quien interactúa con el caso de uso en cuestión.
 - **Precondiciones:** condiciones que deben cumplirse de forma previa a realizar la acción.
 - **Postcondiciones:** estado del sistema después de la realización de la acción.
 - **Escenario básico:** límites del sistema, descripción por fases para el cumplimiento del caso de uso.

Las figuras 12 a 18 representan los diagramas de casos de uso identificados, las tablas que acompañan las figuras contienen la descripción textual de cada caso de uso de los presentados en los diagramas (Tablas de 59 a 75).

- **Casos de uso – Escenario 0**

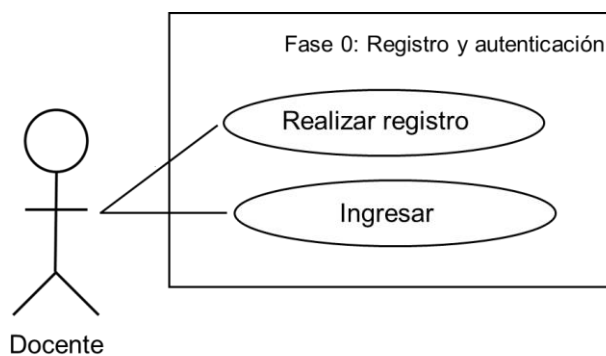


Figura 12. Diagrama de casos de uso para el escenario 0

Tabla 59. Caso de uso CUE0.1

Nombre Identificador	e	Realizar registro – CUE0.1
Objetivo		Permitir el registro al docente para que ingrese al sistema
Actor		Docente
Precondiciones		-
Postcondiciones		El usuario puede ingresar al sistema
Escenario inicial		<ol style="list-style-type: none"> 1. Se accede a la aplicación web 2. Se selecciona la opción para el registro

Tabla 60. Caso de uso CUE0.2

Nombre Identificador	e	Ingresar – CUE0.1
Objetivo		Permitir el ingreso al sistema al docente que se encuentre registrado
Actor		Docente
Precondiciones		Estar registrado
Postcondiciones		El usuario puede emplear el sistema
Escenario inicial		<ol style="list-style-type: none"> 1. Se accede a la aplicación web 2. Se selecciona la opción ingresar 3. Se proporciona el nombre de usuario y contraseña

- **Casos de uso – Escenario 1**

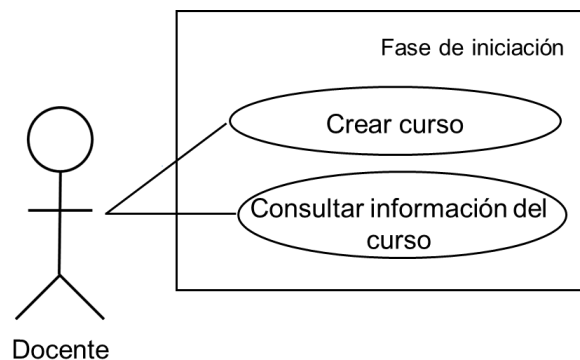


Figura 13. Diagrama de casos de uso para el escenario 1**Tabla 61.** Caso de uso CUE1.1

Nombre Identificador	e	Crear curso – CUE1.1
Objetivo		Permitir al docente crear cursos en los que puedan incluir el objetivo y especificar el área de conocimiento
Actor		Docente
Precondiciones		Haber ingresado
Postcondiciones		Se agrega un curso a la base de cursos del prototipo
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa a la aplicación web 2. Se selecciona curso - agregar 3. Se ingresa la información del curso

Tabla 62. Caso de uso CUE1.2

Nombre Identificador	e	Consultar información del curso – CUE1.2
Objetivo		Permitir al docente consultar la información de los cursos que se encuentren guardados en el sistema
Actor		Docente
Precondiciones		Haber ingresado
Postcondiciones		-
Escenario inicial		<ol style="list-style-type: none"> 1. Se accede a la aplicación web 2. Se ingresa a cursos 3. Se selecciona el curso a consultar

- **Casos de uso – Escenario 2**

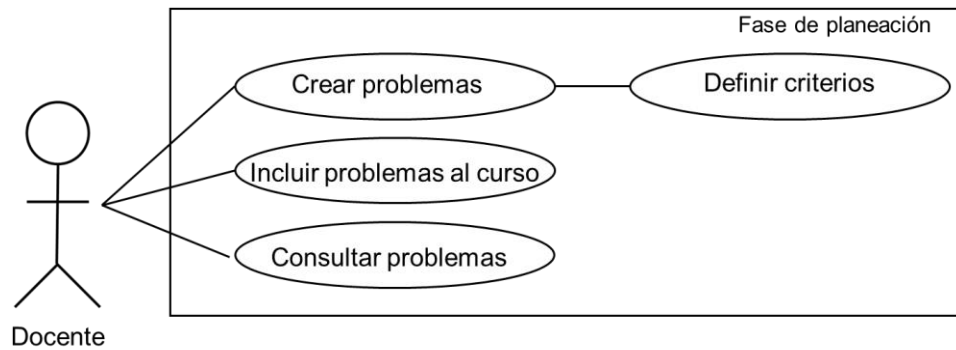


Figura 14. Diagrama de casos de uso para el escenario 2

Tabla 63. Caso de uso CUE2.1

Nombre Identificador	e	Crear problemas – CUE2.1
Objetivo		Permitir al docente crear problemas para que los estudiantes los solucionen
Actor		Docente
Precondiciones		Haber ingresado al sistema
Postcondiciones		Se agrega un problema a la base de problemas del prototipo
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona problemas 3. Se incluye la información del problema a crear

Tabla 64. Caso de uso CUE2.2

Nombre Identificador	e	Definir criterios – CUE2.2
Objetivo		Permitir la definición de los criterios de evaluación para cada problema creado
Actor		Docente
Precondiciones		Crear un problema
Postcondiciones		El problema incluirá los criterios de evaluación
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el problema

5. HERA-CT

	3. Se proporcionan los criterios mínimos
--	--

Tabla 65. Caso de uso CUE2.3

Nombre Identificador	e Incluir problemas al curso – CUE2.3
Objetivo	Permitir al docente incluir en su curso problemas creados para que sus estudiantes los solucionen
Actor	Docente
Precondiciones	Contar con problemas en la base de problemas del sistema
Postcondiciones	El curso contiene problemas
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el curso 3. Se selecciona incluir problemas 4. Se seleccionan los problemas a incluir en el curso

Tabla 66. Caso de uso CUE2.4

Nombre Identificador	e Consultar problemas – CUE2.4
Objetivo	Permitir al docente consultar la información de los problemas de un curso
Actor	Docente
Precondiciones	Haber ingresado al sistema
Postcondiciones	-
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona problemas 3. Se selecciona el problema a consultar

- Casos de uso – Escenario 3

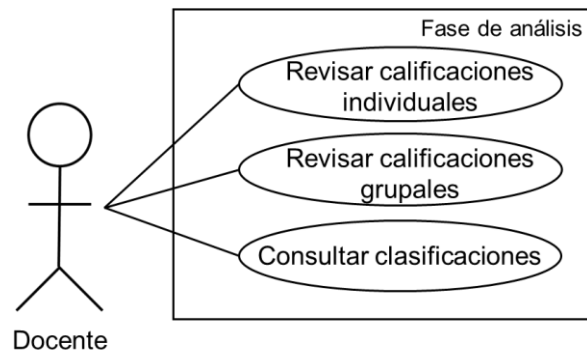


Figura 15. Diagrama de casos de uso para el escenario 3

Tabla 67. Caso de uso CUE3.1

Nombre Identificador	e	Revisar calificaciones individuales – CUE3.1
Objetivo		Permitir al docente observar y analizar las calificaciones individuales de sus estudiantes
Actor		Docente
Precondiciones		Tener un curso con problemas Los estudiantes del curso han enviado sus soluciones a calificar
Postcondiciones		El docente puede definir las acciones siguientes basado en las conclusiones a las que llegue
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el curso 3. Se selecciona el estudiante a quien se van a revisar las calificaciones

Tabla 68. Caso de uso CUE3.2

Nombre Identificador	e	Revisar calificaciones grupales – CUE3.2
Objetivo		Permitir al docente observar y analizar las calificaciones grupales de sus estudiantes
Actor		Docente
Precondiciones		Tener un curso con problemas

5. HERA-CT

	Los estudiantes del curso han enviado sus soluciones a calificar
Postcondiciones	El docente puede definir las acciones siguientes basado en las conclusiones a las que llegue
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el curso 3. Se seleccionan los estudiantes a quienes se van a revisar las calificaciones

Tabla 69. Caso de uso CUE3.3

Nombre Identificador	e Consultar clasificaciones – CUE3.3
Objetivo	Permitir al docente consultar las clasificaciones de sus estudiantes
Actor	Docente
Precondiciones	Tener un curso con problemas Los estudiantes del curso han enviado sus soluciones a calificar
Postcondiciones	El docente puede definir las acciones siguientes basado en las conclusiones a las que llegue
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el curso 3. Se seleccionan clasificación

- **Casos de uso – Escenario 4**

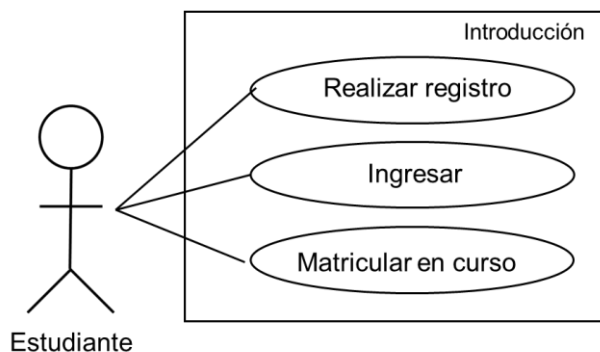


Figura 16. Diagrama de casos de uso para el escenario 4**Tabla 70.** Caso de uso CUE4.1

Nombre Identificador	e	Realizar registro – CUE4.1
Objetivo		Permitir el registro al estudiante para que ingrese al sistema
Actor		Estudiante
Precondiciones		-
Postcondiciones		El usuario puede ingresar el sistema
Escenario inicial		<ol style="list-style-type: none"> 1. Se accede a la aplicación web 2. Se selecciona registrar 3. Se proporciona la información de registro

Tabla 71. Caso de uso CUE4.2

Nombre Identificador	e	Ingresar – CUE4.2
Objetivo		Permitir el ingreso al sistema al estudiante que se encuentre registrado
Actor		Estudiante
Precondiciones		Estar registrado
Postcondiciones		El usuario puede emplear el sistema
Escenario inicial		<ol style="list-style-type: none"> 1. Se accede a la aplicación web 2. Se selecciona ingresar 3. Se proporciona el nombre de usuario y contraseña

Tabla 72. Caso de uso CUE4.3

Nombre Identificador	e	Matricular en curso – CUE4.3
Objetivo		Permitir al estudiante matricularse en el curso correspondiente
Actor		Estudiante

5. HERA-CT

Precondiciones	Haber ingresado
Postcondiciones	El usuario puede consultar la información del curso
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se selecciona el curso correspondiente 3. Se matricula

- Casos de uso – Escenario 5

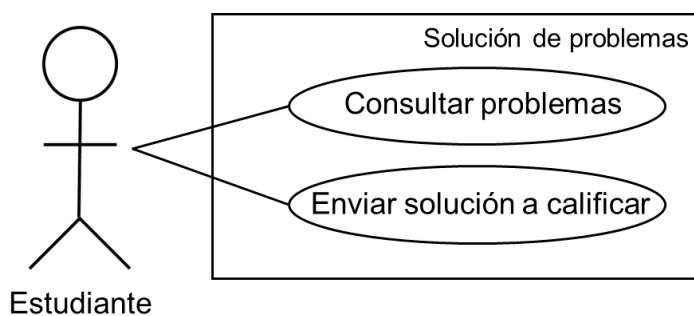


Figura 17. Diagrama de casos de uso para el escenario 5

Tabla 73. Caso de uso CUE5.1

Nombre Identificador	e	Consultar problemas – CUE5.1
Objetivo		Permitir el estudiante consultar la información de los problemas que se encuentran en el curso y debe resolver
Actor		Estudiante
Precondiciones		Estar matriculado en un curso
Postcondiciones		El usuario conoce el problema y puede empezar a resolverlo en Scratch
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se el curso 3. Se selecciona el problema a consultar

Tabla 74. Caso de uso CUE5.2

Nombre Identificador	e	Enviar solución a calificar – CUE5.2
-----------------------------	---	--------------------------------------

Objetivo	Permitir al estudiante enviar la solución del problema para que sea calificada
Actor	Estudiante
Precondiciones	Tener solución en Scratch
Postcondiciones	El usuario puede visualizar la realimentación
Escenario inicial	<ol style="list-style-type: none"> 1. Se ingresa al sistema 2. Se lee y entiende el problema a resolver 3. Se resuelve el problema a través de programación en Scratch 4. Se envía la solución a ser calificada

- **Casos de uso – Escenario 6**

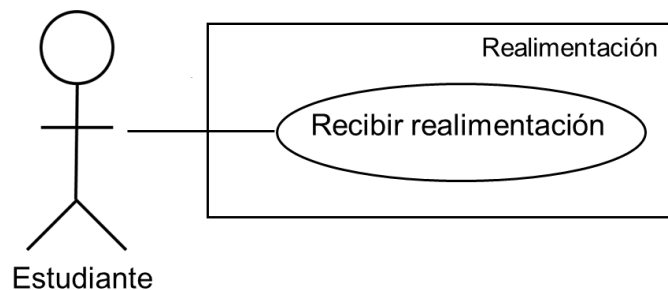


Figura 18. Diagrama de casos de uso para el escenario 6

Tabla 75. Caso de uso CUE6.1

Nombre Identificador	e	Recibir realimentación – CUE6.1
Objetivo		Facilitar la entrega de realimentación adecuada a la evidencia entregada por el estudiante.
Actor		Estudiante
Precondiciones		Haber enviado la solución a calificar
Postcondiciones		El estudiante puede continuar con otro problema
Escenario inicial		<ol style="list-style-type: none"> 1. Se ingresa a la aplicación web 2. Se lee y entiende el problema resolver 3. Se resuelve el problema a través de programación en Scratch

	4. Se envía la solución a ser calificada
--	--

5.2. Diseño del prototipo

Teniendo en cuenta los requisitos y demás información del análisis del prototipo, para el diseño se crean modelos de arquitectura de alto nivel que detallan los componentes que forman el sistema, con el fin de llevarlo a la implementación. Primero se brinda una visión general de la solución, seguido de la descripción de los procesos más importantes, después se presenta la arquitectura general del sistema y finalmente el modelo de entidad relación de la base de datos.

5.2.1. Visión general de la solución

La implementación que se propone, busca hacer una evaluación de tal manera que se consideren las actividades realizadas por el estudiante como evidencia para determinar su desempeño, y se contraste con los criterios establecidos por el docente para brindarle al estudiante una realimentación más cercana a sus necesidades. Además, ofrecerle al docente un seguimiento del desempeño de sus estudiantes.

Para esto, se ha determinado definir un repositorio de problemas de diferentes áreas del conocimiento, que se va nutriendo a medida de que los docentes interactúen con el sistema e incluyan diferentes problemas con los criterios mínimos que se desean evaluar. Se determina que dentro del sistema el nombre *problema* será reemplazado por *desafío*.

Los estudiantes y docentes, componen los clientes de la herramienta, ya que ellos van a interactuar con ella a través de la evaluación de los criterios del CT y el seguimiento de esta evaluación.

Los problemas que los estudiantes solucionan deben ser referenciados o incluidos en el sistema para evaluar los criterios definidos. Con el fin de realizar una evaluación para el aprendizaje, se ha determinado que cuando un estudiante termine una

actividad, se le debe brindar una realimentación acorde con su ejecución de la solución del problema., la realimentación del sistema será una respuesta ludificada que depende del resultado de su desempeño y los criterios que el docente consideró evaluar, la descripción de la ludificación se encuentra ampliamente descrita en el trabajo de grado: “Estrategias de ludificación para evaluar el pensamiento computacional a través de un modelo basado en evidencia”.

Los docentes podrán llevar un seguimiento de las estimaciones del desarrollo de CT de sus estudiantes, con el fin de determinar las tendencias, facilidades y dificultades que tienen durante en este desarrollo, detectar errores y determinar las mejoras apropiadas tanto en la enseñanza como en la evaluación.

5.2.2. Procesos principales de la solución

Los cuatro (4) procesos principales dentro del proceso de evaluación definido se presentan en la figura 19. El docente realiza un proceso de selección de desafíos, los cuales pueden ser creados por el mismo u otros docentes, cuando los docentes crean los desafíos deben incluir principalmente la descripción y los criterios mínimos de evaluación.

Una vez el docente selecciona los desafíos, el estudiante puede observar los desafíos a través del proceso de presentación. Cuando el desafío ha sido entendido, resuelto y enviado a calificar, la información llega al proceso de respuesta que se encarga de generar la realimentación al estudiante y enviar al proceso de estimación la información recolectada. El proceso de estimación genera los resultados generales y guarda la información recolectada para llevar un seguimiento del desempeño de los estudiantes.

5. HERA-CT

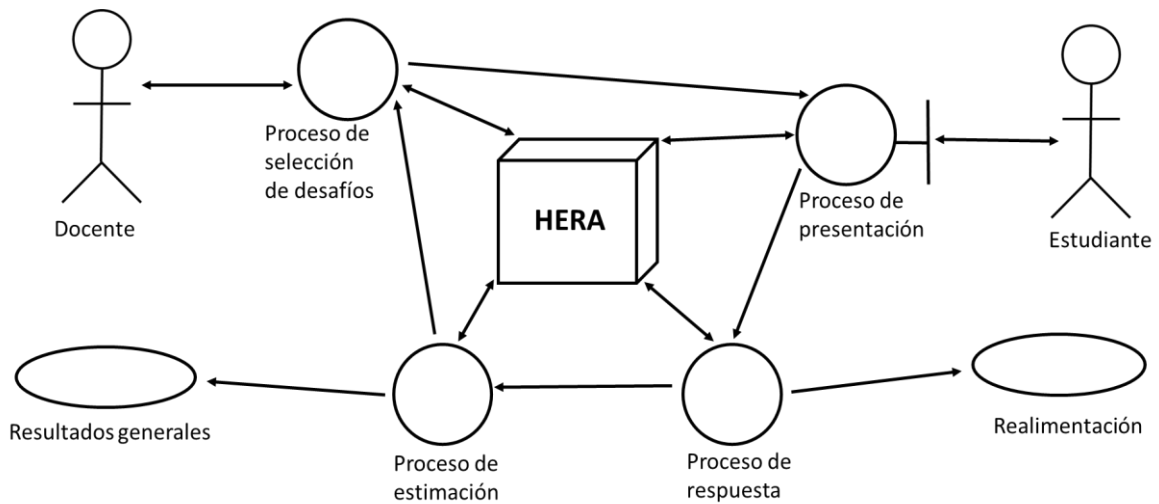


Figura 19. Procesos principales de HERA-CT

5.2.3. Arquitectura de la solución

El diseño de la aplicación web de evaluación para estimar el desarrollo de CT ha sido basado en el modelo de arquitectura Cliente/Servidor, el esquema puede ser observado en la figura 20. Cliente/Servidor basa su modelo en el procesamiento cooperativo de datos a través de un grupo de terminales llamados clientes, quienes solicitan a través de Internet a los servidores estos requerimientos. El sistema es centralizado, ya que los recursos se controlan desde el servidor, lo que contribuye con la seguridad del sistema, ya que posibles inconvenientes pueden ser generados por clientes

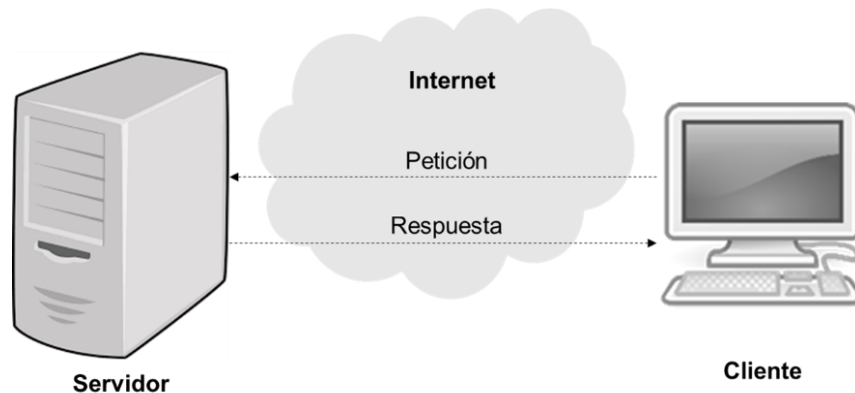


Figura 20. Esquema Cliente/Servidor

Considerando el modelo cliente servidor, el sistema se diseñó compuesto por tres (3) capas principales: capa de cliente, capa de lógica y capa de base de datos (Figura 21).

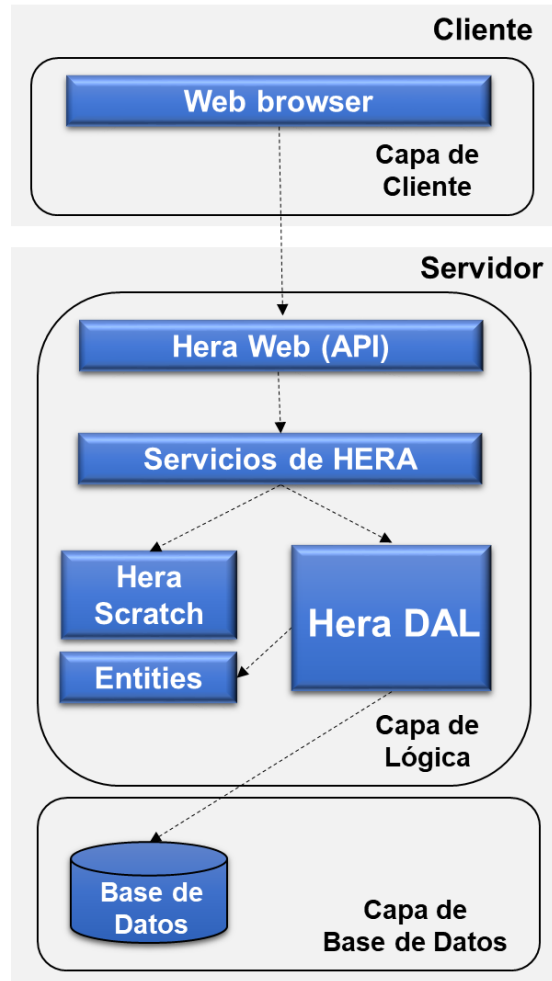


Figura 21. Diagrama de componentes del sistema

La capa de cliente es una aplicación Web que facilita la interacción de los usuarios con el sistema, se encuentra estructurada en siete (7) módulos que conforman el conjunto de componentes web. El módulo 1 tiene los componentes y servicios que son transversales en su aplicación y, son exportados en los módulos funcionales (feature modules) de la aplicación. El módulo 2 abarca los componentes del diseño gráfico general de la aplicación web. Los módulos 3 y 4 agrupan los requerimientos orientados a los usuarios, especializándose el 3 en los profesores y el 4 en estudiantes. El módulo 5 es el intermediario entre los módulos funcionales y el procesamiento de la

5. HERA-CT

información de los proyectos Scratch (la cual es realizada en la capa de lógica). El módulo 6 realiza los procesos de validación y seguridad. Y el módulo 7 manipula los elementos para la generación de animaciones en 2D en el navegador.

La capa de lógica realiza el análisis y procesamiento de código Scratch, y es el puente entre el cliente y la base de datos, se compone por seis (6) subcapas. La subcapa de Web API realiza la gestión de las peticiones HTTP realizadas al servidor de manera adecuada, lo que incluye: autenticación, validación de información de las peticiones, redirección de recursos HTTP, etc. La subcapa servicios de Hera proporciona a la Web API los recursos que necesita para realizar sus acciones. La subcapa Hera Scratch facilita el análisis y procesamiento de proyectos Scratch de acuerdo a los criterios establecidos, emplea una subcapa HttpClient que proporciona servicios cuando los requiere. Finalmente, la subcapa Hera DAL proporciona mecanismos para acceder a la información almacenada en la base de datos, empleando la subcapa Entities, encargada de definir los DTOs (Data Transfer Objects) del proyecto.

La capa de base de datos contiene la base de datos. La base de datos tiene un diseño que facilita la interoperabilidad entre los desafíos, cursos, y calificaciones de cada estudiante. Además, teniendo en cuenta entidades especializadas para capturar la información procesada por los análisis de código y facilitar la generación de estadísticas detalladas del desempeño de un estudiante o el curso en general. La estructura de la base de datos se puede apreciar en la figura 22, que muestra el modelo entidad relación. El modelo físico de la base de datos se puede apreciar en el Anexo C, tabla d2.

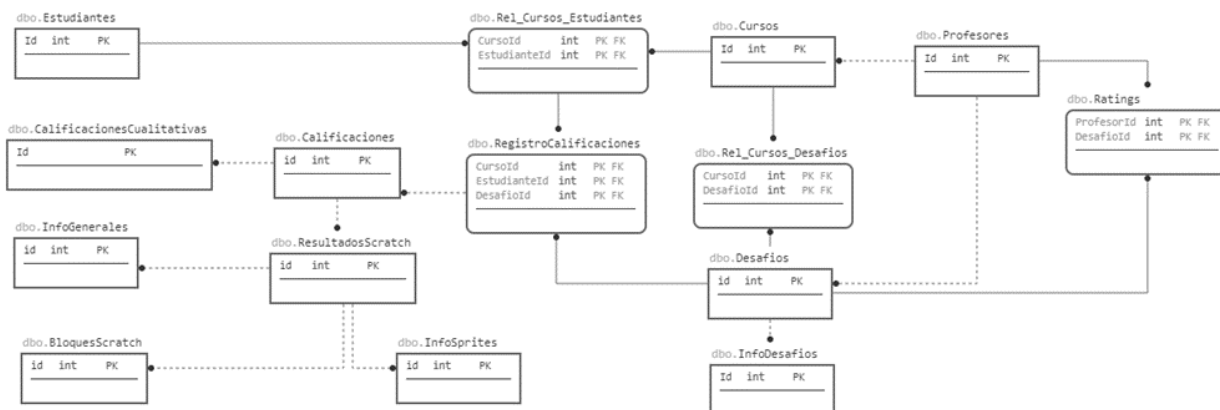


Figura 22. Modelo entidad relación de la base de datos

5.3. Implementación del prototipo

A pesar de que en un principio se inició modelando en UML (Unified Modeling Language), finalmente para implementación del prototipo la metodología de desarrollo se fundamentó en extraer las actividades y entregables estrictamente necesarios. Se priorizó en la rápida generación de pequeños entregables que brindan valor al usuario final. El proceso se conformó por la ejecución de múltiples Sprints, diseñados para implementar las diferentes funcionalidades al sistema.

La ejecución de cada Sprint siguió el siguiente proceso: (i) selección de requerimientos, su prioridad depende de la complejidad y el valor que aporte al usuario final, (ii) definición del plan entregable, después de que se establecen los requerimientos se determina un responsable y un tiempo prudente para su entrega, (iii) construcción, ejecución de pruebas e integración en producto, se construyen los requerimientos definidos, se procede con la ejecución de pruebas unitarias, lo que se hizo de manera automática a partir del uso de marcos de referencia en un proceso de integración continuo. Para la capa cliente se empleó Jasmine y para la capa servidor las pruebas unitarias se hicieron mediante xUnit, y (iv) inyección de requerimientos, finalmente se realiza una reunión para mostrar los avances integrados y continuar con la definición de los siguientes requerimientos, establecidos previamente o surgidos durante el proceso de desarrollo.

Para la implementación del prototipo se emplearon diferentes tecnologías y marcos de referencia, los cuales se describen a continuación.

- La capa de cliente está construida en Angular, un marco de referencia de código abierto, que cuenta con el soporte de Google para desarrollar de aplicaciones web basadas en componentes, empleando una arquitectura modelo-vista (MV: Model-View). También se emplean otras tecnologías como: Typescript, RXJS, etc.
- La capa de lógica está desarrollada en .NET Core, un marco de referencia de software de computadora gratuito y de código abierto, desarrollado por Microsoft con el soporte del MIT, empleando un servidor

5. HERA-CT

Azure Web que ofrece un servicio de computación en la nube creado por Microsoft para construir aplicaciones.

- La capa de base de datos se implementó en Azure SQL Server, un servicio de base de datos en la nube que proporciona gran compatibilidad.

5.3.1. Interfaz gráfica de usuario

Para presentar la herramienta de evaluación se ilustraran algunas de las interfaces gráficas de usuario (GUI: Graphical User Interface), para los usuarios del sistema: docentes y estudiantes.

- GUI del docente

La GUI del docente le permite realizar el registro e ingreso al sistema, gestionar cursos, gestionar desafíos y revisar las calificaciones. A continuación se presentan algunas de estas interfaces.

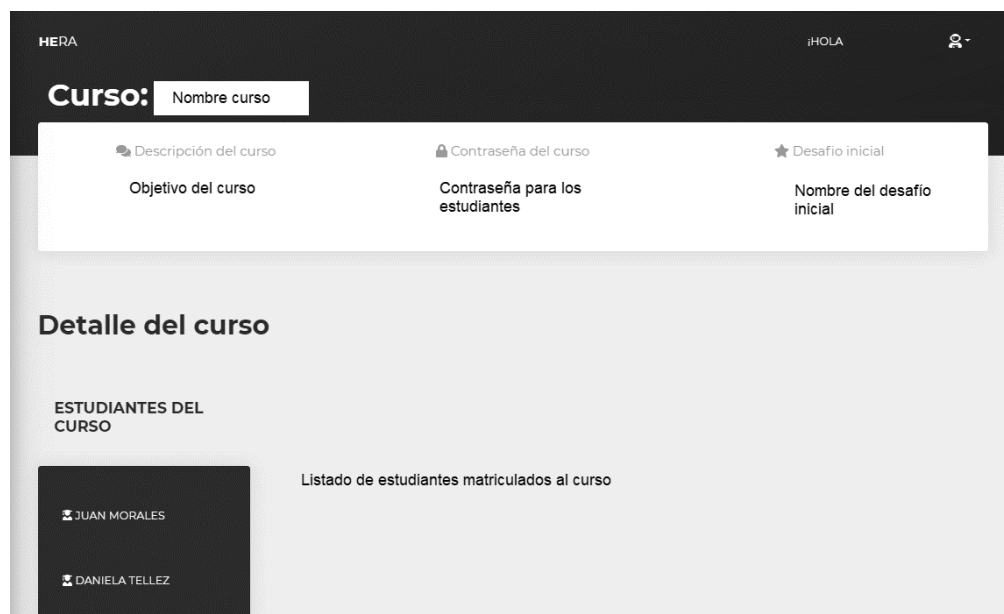


Figura 23. GUI información curso

La GUI representada en la figura 23 presenta la información del curso, donde se especifica (i) el nombre del curso, se recomienda que haga alusión al área del conocimiento para facilitar la búsqueda de desafíos a los docentes de otros cursos, (ii) descripción, se recomienda especificar el objetivo del curso, (iii) contraseña, necesaria para que los estudiantes puedan realizar la matrícula al curso, (iv) desafío inicial, todo curso es creado con un desafío inicial que se especifica en la definición del curso y (v) detalle del curso que incluye la lista de estudiantes, los desafíos del curso y las calificaciones que han obtenido los estudiantes de manera grupal e individual, más adelante se muestran en detalle las calificaciones. La relación de los criterios mínimos de evaluación y los bloques de scratch se encuentra en el Anexo C.

Crear desafío

Nombre del desafío
Nombre desafío

Descripción:
Aquí va la descripción

URL escenario inicial URL solución ID solución

Valoración de la solución

Más de un sprite tiene eventos Uso correcto de mensajes
 Uso, y creación, de variables compartidas Uso y creación de listas

Se usan todos los bloques Uso de bloques de entrada Uso de operadores básicos
 Creación de bloques propios Uso de variables no creadas Uso de operadores complejos
 Uso de clones Uso de sensores de sprite Uso de operadores anidados

Uso de bloques simples Uso de secuencias Dos hilos iniciando con banderas verde
 Uso de bloques complejos Uso más de un hilo por sprite Uso de más de un tipo de evento
 Uso de bloques anidados

Tipo de evaluación

Tipo evaluación: DetalleCars Param 1 Param 2 Param 3

Param 4

Crear desafío ✓

Figura 24. GUI creación de desafío

La figura 24 muestra la GUI de la creación del desafío, aquí los docentes especifican el nombre y descripción del desafío, brindan los enlaces a un escenario inicial, que facilita el trabajo al estudiante, y una posible solución, que sirve de guía para otros docentes. También hay un espacio para la valoración de la solución, donde se seleccionan los criterios mínimos a evaluar y finalmente se selecciona el tipo de

5. HERA-CT

evaluación que se va aplicar, este tipo de evaluación está relacionado con la realimentación ludificada que proporciona la aplicación.

En la figura 25 se puede apreciar el resultado individual de un estudiante con respecto a un desafío, el resultado brinda información acerca de cuándo fue resuelto el desafío, cuánto tiempo tardó el estudiante en realizarlo, cuántos scripts están duplicados o son código muerto, el total de bloques y sprites empleados, y los resultados en cada elemento de CT con respecto a los criterios mínimos seleccionados por el docente.



Figura 25. GUI calificación individual

La figura 26 presenta la interfaz gráfica de los resultados grupales de un ejercicio resuelto, en éste se indica la moda, mediana y media del uso de elementos en relación con los criterios mínimos.

DATOS GENERALES

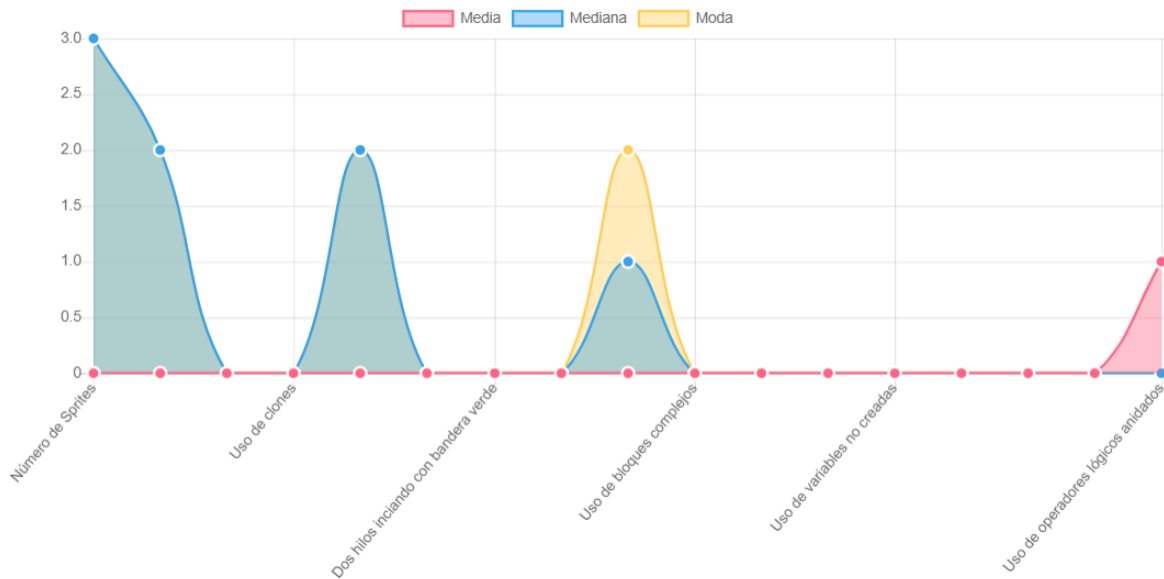


Figura 26. GUI calificación grupal

- GUI del estudiante

La GUI del estudiante proporciona opciones para realizar el registro e ingreso al sistema, matricularse a cursos, consultar desafíos, enviar a calificar los desafíos resueltos y revisar la realimentación. Se presentan en seguida algunas de estas interfaces.

5. HERA-CT



Figura 27. GUI consulta de desafíos

Cuando un estudiante ingresa a un curso al cual se encuentra matriculado, la GUI le presenta el desafío que debe resolver y los que ya están resueltos en caso de que sea así (Figura 27). Al dar clic en Iniciar desafío, el tiempo de realización del desafío empieza a correr y la GUI presentada en la figura 28 se muestra. El estudiante debe seleccionar los integrantes de su grupo de trabajo, en caso de que se cuente con un equipo de trabajo y dar clic en el botón de inicio para tener acceso al escenario.

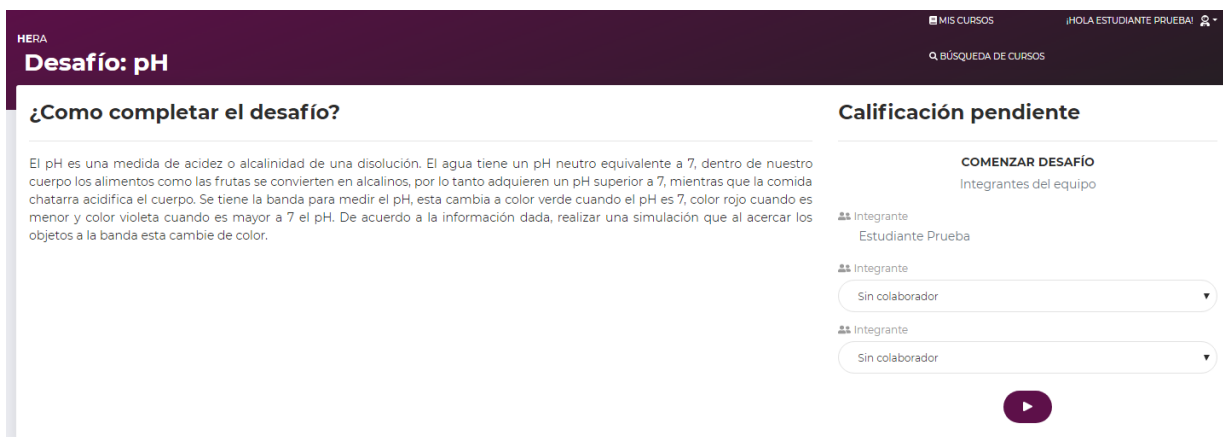


Figura 28. GUI inicio de desafío

Una vez el estudiante da clic en inicio, se pueden ver las opciones mostradas en la figura 29 al lado derecho de la interfaz. Ir al escenario es una opción que conduce al enlace proporcionado por el docente en escenario inicial durante la creación de desafíos. El campo Id proyecto scratch es para incluir el identificador del proyecto a que se envía a calificar.

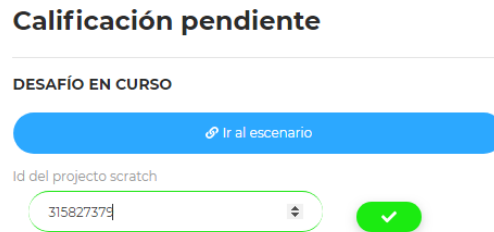


Figura 29. GUI opciones ir al escenario o enviar el proyecto a calificar

La figura 30 enseña una muestra de la realimentación que reciben los estudiantes tras completar un desafío. Se presenta una muestra, ya que las estrategias de ludificación para la realimentación constituyen la parte central de otro estudio.

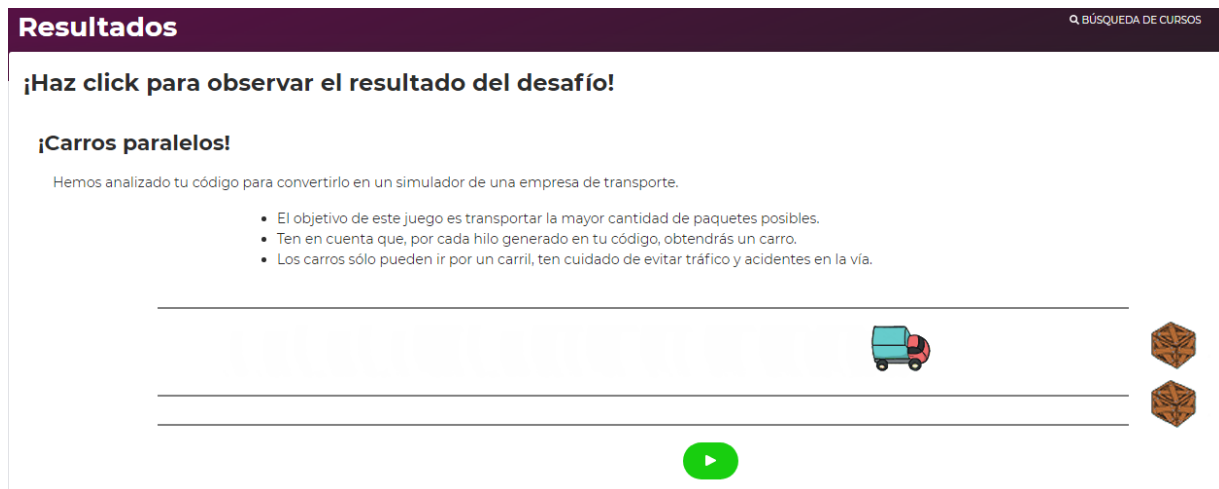


Figura 30. GUI visualización de resultados

La información de realimentación brindada en la figura X muestra la implementación de carros paralelos. El diseño de esta estrategia representa el uso adecuado de hilos y la programación en paralelo, busca evidenciar las consecuencias de un uso no adecuado de los recursos disponibles. El número de carriles representa los hilos

5. HERA-CT

sugeridos para completar el desafío; el número de camiones representa cada hilo creado por el usuario. Con lo que se busca mostrar a los camiones recorriendo los carriles de un lado al otro para transportar paquetes, uno por trayecto.

5.4.Despliegue del prototipo

Para la distribución física de los principales componentes que interactúan con el sistema, se identifican dos nodos que se comunican a través de Internet. El primer nodo es el cliente que representa los computadores de los usuarios del prototipo que emplean la aplicación web. El servidor representa las fuentes de datos de las que se consulta los datos y donde se analiza y procesa la información.

Cuando los usuarios envían peticiones al servidor ocurre la interacción entre el cliente y servidor, algunas de estas consultas son: (i) registro, (ii) autenticación, (iii) consultar curso, (iv) consultar desafío, (v) calificar evidencia, etc. En respuesta el servidor después de recibir una petición devuelve los datos solicitados por el cliente. En la figura 31 se describe la interacción entre los nodos descritos.

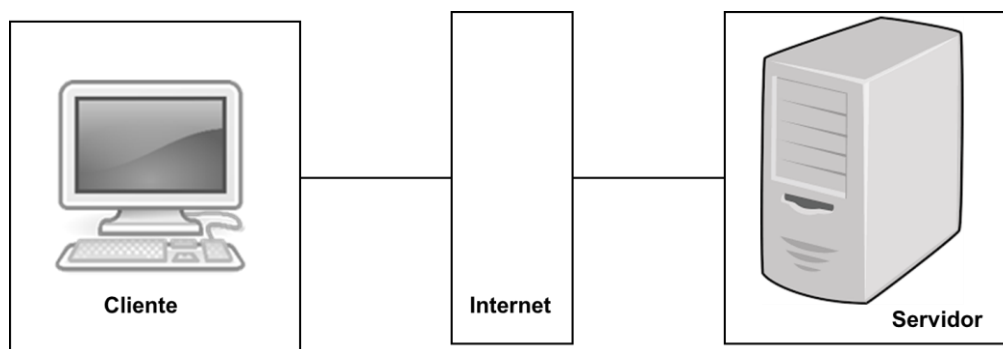


Figura 31. Vista de despliegue

Capítulo 6

6. Estudio de caso

Con el fin de validar el modelo propuesto, se presenta un estudio de caso. El objetivo es tener mayor conocimiento y comprensión acerca de las particularidades y especificaciones del proceso de evaluación para estimar el desarrollo de CT. El estudio se construye siguiendo como referencia la guía para el reporte de estudios de caso en Ingeniería de Software, expuesto por Runeson y Höst [142]. , que define: (i) el diseño y planeación del estudio de caso, (ii) la preparación y recolección de datos y (iii) análisis de los datos.

6.1. Diseño y planeación del estudio de caso

El presente estudio de caso se realiza con el propósito de evaluar el impacto del montaje del modelo de evaluación propuesto a lo largo del presente trabajo de investigación. Se evaluará el desempeño de los estudiantes con respecto a los criterios mínimos seleccionados por el docente y se comparará este resultado con el entregado por el docente en las sesiones de clase con el fin de determinar similitudes y diferencias.

6.1.1. Definición del caso

La definición del caso entrega una descripción general que incluye el objetivo, las preguntas de investigación, los métodos de recolección de datos y otra información relevante.

- **Objetivo:** Implementar un proceso de evaluación en el aula de clase para comprobar si con el modelo propuesto en el presente trabajo se logra estimar el desarrollo de CT en los estudiantes de educación básica.
- **Casos y unidades de análisis:** en la tabla 76 se pueden apreciar los tres (3) aspectos que se evalúan en este estudio y su justificación dentro del estudio de caso.

Tabla 76. Casos y unidades de análisis

Aspectos a evaluar	Justificación en el estudio de caso
Información personal y de autoeficacia acerca del uso del computador. Referencia: Anexo D, tabla d1.	Aunque no se trata de aspectos que se evalúen directamente, porque se trata de información personal y de autoeficacia, que pueden tener incidencia en el desarrollo de CT y se deben tener en cuenta para el análisis y toma de decisiones.
Criterios mínimos seleccionados por el docente. Referencia: Anexo C, tabla c1.	El análisis del código Scratch brinda los resultados de la evidencia entregada por el estudiante. La información es entregada en términos de los elementos de CT empleados por los estudiantes, en las soluciones propuestas para los problemas planteados, en relación con los criterios mínimos seleccionados por el docente.
Información acerca de la depuración,	La información considerada incide en la estimación del desarrollo de CT, ya que considera los elementos de CT

6. Estudio de caso

generalización y evaluación. Referencia: Tabla 41.	encontrados en el proceso de caracterización. Estos no pueden ser evaluados mediante el análisis del código, sino que deben ser preguntados u observados.
---	---

- **Marco de referencia teórico:** El marco de referencia teórico se encuentra consignado en el Capítulo 2.
- **Preguntas de investigación:** Para el desarrollo del estudio de caso se proponen las siguientes preguntas de investigación:
 - ¿Que influencia tiene la información personal en la estimación del desarrollo de CT del estudiante?
 - ¿Que influencia tiene la información de autoeficacia en la estimación del desarrollo de CT del estudiante?
 - ¿Que influencia tiene la información de prácticas de CT en la estimación del desarrollo de CT del estudiante?
 - ¿Que influencia tiene la evidencia entregada a través de prácticas de programación, en la en la estimación del desarrollo de CT del estudiante?
- **Métodos de recolección de datos:** en la tabla 77 se muestran los tres (3) aspectos que se evalúan en este estudio y los métodos de recolección de esos datos, los métodos corresponden a las mecánicas de evaluación que se pueden emplear para evaluar los aspectos.

Tabla 77. Métodos de recolección de datos

Aspectos a evaluar	Mecánicas de evaluación
Información personal y de autoeficacia acerca del uso del computador.	Encuesta: preguntas realizadas en el registro del prototipo
Criterios mínimos seleccionados por el docente.	Análisis automático: procesamiento de código Scratch a través de la aplicación Web Hera

Información acerca de la depuración, generalización y evaluación.	Entrevista: preguntas estructuradas acerca de las prácticas de CT, evaluación de pares y autoevaluación
---	--

- **Selección de datos:** los datos se recolectan mediante los métodos establecidos, durante las sesiones de la clase de informática. Se tendrán en cuenta los datos cuantitativos, provenientes del análisis de código Scratch y de las respuestas a preguntas con opciones limitadas, y los datos cualitativos, obtenidos en la entrevista dónde los estudiantes respondieron preguntas estructuradas sin opciones establecidas de respuesta, se discretizaran para un análisis cualitativo.
- **Selección de estrategias:** para este estudio de caso se realizarán pruebas con tres (3) grupos de estudiantes de educación básica del Instituto Técnico Industrial de la ciudad de Popayán, Colombia.

A pesar de que el cuarto objetivo específico menciona que el estudio de caso se realizaría en el aula Smart School, por razones administrativas no se pudo hacer con la población asistente al aula, sin embargo, la población del Instituto cumple con las características de la población general a la que va dirigido el modelo de evaluación basado en evidencia.

El Instituto Técnico Industrial es una institución educativa orientada al desarrollo de habilidades técnicas en los estudiantes. Particularmente, los estudiantes participantes en el estudio de caso, se encuentran matriculados en los énfasis de programación y mantenimiento de computadores.

6.1.2. Protocolo del estudio de caso

El protocolo de estudio de caso contiene las decisiones de diseño del estudio y los procedimientos de campo a llevar a cabo. La definición de estas decisiones de diseño van cambiando continuamente mientras el plan del estudio cambié, aquí se describe el protocolo definitivo. Contiene un preámbulo, información general del proyecto, instrumentos de investigación, como encuestas o cuestionarios, y guías para el análisis de información.

6. Estudio de caso

- **Preámbulo:** se describe el protocolo del estudio de caso con el fin de organizar las decisiones de diseño, tomadas para la realización del proceso de evaluación de criterios mínimos de CT.
- **Información general:** la información general del proyecto puede consultarse en el Capítulo 1, donde se encuentra ampliamente detallada.
- **Instrumentos de investigación:** para la investigación se emplearán: (i) encuesta, realizada junto con el registro, (ii) análisis automático de código Scratch, a través de HERA-CT después de la solución de cada problema, los problemas o desafíos empleados en el estudio de caso se encuentran consignados en el Anexo D, tabla d2, y (iii) entrevista, estructurada con el fin de obtener información de depuración, generalización y evaluación.
- **Guías de análisis de información:** para la información cuantitativa se realiza un análisis estadístico descriptivo, se tienen en cuenta la información de correlación de la evidencia entregada por el estudiante con respecto a los criterios mínimos seleccionados por el docente y se prueba la propuesta.
 - Con el fin de facilitar el proceso de análisis, la información recolectada de la encuesta ha sido discretizada teniendo en cuenta las opciones respondidas por los estudiantes, las cuales por sus características permitieron realizar esta discretización.

6.1.3. Consideraciones éticas

Los participantes del estudio son los estudiantes del Instituto Técnico Industrial de Popayán, ellos son informados del objetivo acerca de la estimación del desarrollo de CT mediante un modelo de evaluación, se les informa acerca del proceso al que se van a someter, este proceso se encuentra explicado en el Módulo del estudiante descrito en el Capítulo 4, también se les informa acerca de la intención de la investigación y se da un consentimiento informado con el fin de que la información obtenida en este proceso, sea analizada y publicada, respetando su privacidad.

De acuerdo a la reglamentación colombiana y a la luz de LA RESOLUCIÓN N° 008430 DE 1993 del Ministerio de Salud, existen los factores principales que proveen un buen marco para el desarrollo ético de las investigaciones en seres humanos:

- (i) **Valor:** la investigación busca validar el modelo de evaluación basado en evidencia definido para un entorno de educación básica.
- (ii) **Validez científica:** las pruebas se ejecutan con el objetivo de verificar la claridad del proceso de evaluación y de las prácticas implementadas. Se realizan seis sesiones mínimo por grupo, 12 sesiones máximo.
- (iii) **Proporción favorable de riesgo/ beneficio:** los riesgos a los que se exponen los participantes de la investigación son mínimos, dado que el estudio emplea técnicas y métodos de investigación documental retrospectivos; los beneficios potenciales para los individuos y los conocimientos ganados para la sociedad deben sobrepasar los riesgos.
- (iv) **Consentimiento informado:** los grupos de estudiantes son informados acerca de la investigación y dieron su consentimiento voluntario antes de convertirse en participantes de la investigación. El consentimiento informado se puede ver en el Anexo D, tabla d4.
- (v) **Respeto hacia los participantes:** se mantiene protegida la privacidad de los participantes, tienen la opción de dejar la investigación y tener un monitoreo de su bienestar.

6.2.Recolección de datos

Para recolectar los datos se proporcionaron espacios de trabajo durante sesiones en el aula de clase. Primero se consultó con los docentes y se acordaron las fechas para realizar las sesiones con estudiantes de los cursos de Programación e informática del Instituto Técnico Industrial de los grados octavo (8vo) y noveno (9no).

Los estudiantes participantes fueron inicialmente 57, de edades entre los 11 y 16 años, con un promedio de 14,5 años. La relación hombres mujeres es de 3:1.

- **Grupo 1:** noveno grado. 13 estudiantes. 6 sesiones.

6. Estudio de caso

- **Grupo 2:** noveno grado, 23 estudiantes. 12 sesiones.
- **Grupo 3:** octavo grado, 21 estudiantes. 12 sesiones.

Sin embargo, únicamente 40 estudiantes completaron las 12 sesiones e interactuaron con los tres (3) mecanismos de evaluación propuestos en este estudio, como los otros 13 estudiantes no completaron el número de sesiones son descartados del estudio de caso.

Después de explicar a los participantes los objetivos y condiciones de la investigación, y recibir la autorización escrita del consentimiento informado de cada uno de ellos, se realiza una introducción acerca de las herramientas que se van a emplear para facilitar la recolección de datos, en este caso las herramientas son Scratch y HERA-CT.

Durante el proceso, tanto el docente del grupo como el investigador se encontraron presentes siempre.

6.2.1. Fuentes de datos

- **Información de la encuesta inicial:** esta información se recolecta en la introducción de los estudiantes, cuando realizan el registro a la aplicación HERA-CT. Se trata de datos referentes a la información personal de los participantes y algunas preguntas de autoeficacia con respecto al manejo del computador y conocimientos generales de este. La tabla d1 del anexo D contiene la información descrita.
- **Resultados del análisis automático de código Scratch:** se proponen 13 desafíos para ser resueltos por los estudiantes durante sus prácticas de programación. Los desafíos corresponden a problemas de ciencias naturales y matemáticas, relacionados con temáticas encontradas en los formularios de preguntas de las pruebas saber 9, los problemas planteados y los criterios mínimos seleccionados en cada problema se encuentran en el anexo D, tablas d2 y d3.
- **Información de la entrevista:** se realizó una entrevista con preguntas estructuradas y respuestas abiertas, sin embargo los resultados de estas fueron discretizados con el fin de facilitar el análisis de los datos. La

entrevista se realizó personalmente con cada estudiantes, no obstante debido a que se requiere de mucho tiempo para la realización de las entrevistas, se propone una encuesta de análisis automático, incorporada a HERA-CT, con las opciones encontradas en la discretización de las variables. La información se puede consultar en el anexo D, tabla d6.

6.3. Análisis de los datos recolectados

Para realizar el análisis de los datos, es necesario reunir toda la información disponible. Como la información es tan extensa se muestra la que ha sido considerada la más relevante y pertinente para este estudio.

Como se mencionó anteriormente los datos recolectados corresponden a:

- La información completa de los datos personales y respuestas de autoeficacia se pueden encontrar en la tabla d5 del anexo D, aunque se puede apreciar un resumen de la información en la tabla 78.
- La información de desempeño de los estudiantes para lo que se muestran tres gráficas que presentan los resultados de los desafíos seleccionados (Gráficas 32 a 34).
- La tabla 79 contiene la clasificación de los estudiantes.
- La información de la entrevista se puede encontrar en la tabla d6 del anexo D, aunque se puede apreciar un resumen de la información en la tabla 78.

Se puede apreciar que la mayoría de los participantes son hombres, correspondiente a un 63% de la población estudiada.

La edad promedio es de 14, 5 años, con edades entre los 13 y 17 años.

La mayoría de los estudiantes están cursando noveno grado, aunque algunos se encuentran en octavo.

Tabla 78. Medidas estadísticas aplicadas

Variable	Sumatoria	Promedio	Moda
Genero	36	0,63157895	1

6. Estudio de caso

Edad		14,5087719	15
Grado			9
# sesiones			12
Frecuencia Pc			0
Conocimiento Computador			10
Manejo Computador			5
Actividad Favorita PC			0
Ejercicio1	34	0,59649123	1
Ejercicio2	57	1	1
Ejercicio3	57	1	1
Ejercicio4	57	1	1
Ejercicio5	56	0,98245614	1
Ejercicio6	48	0,84210526	1
Ejercicio7	37	0,64912281	1
Ejercicio8	32	0,56140351	1
Ejercicio9	30	0,52631579	0
Ejercicio10	21	0,36842105	0
Ejercicio11	13	0,22807018	0
Ejercicio12	6	0,10526316	0
Ejercicio13	4	0,07017544	0
AutoEval			4
Generalizacion			0
Depuracion			3
Valoración docente		3,76491228	3,8

La tabla 38 nos arroja información acerca de las preferencias de los estudiantes por ejemplo la actividad que más realizan en el computador es interactuar por redes sociales, lo que nos muestra de que les interesa la comunicación, lo que podría tenerse en cuenta para otro estudio. Otra característica importante, es que la gran mayoría interactúan con un computador todos los días, lo que les permite tener acceso a información de diferente tipo.

La mayoría de los estudiantes creen tener un conocimiento del computador alto, lo que depende de las razones para que lo usen, lo que se encuentra muy relacionado con la

autoevaluación ya que se demuestra que los estudiantes tienen un alto concepto acerca de su desempeño, a pesar de que los resultado muestran que los estudiantes presentan falencias.

No todos los estudiantes completaron la totalidad de los desafíos propuestos, por lo que para la evaluación del desempeño de los desafíos finales solo se contaba con poca información.

Casi ningún estudiante realiza prácticas referentes a la generalización de patrones y muy pocos depuran su código.

La figura 32 presenta los resultados de desempeño de los estudiantes del desafío 2, se pueden ver que emplean los bloques más básicos para resolver el desafío, además siendo un desafío de poca dificultad no se requiere emplear adicionales.

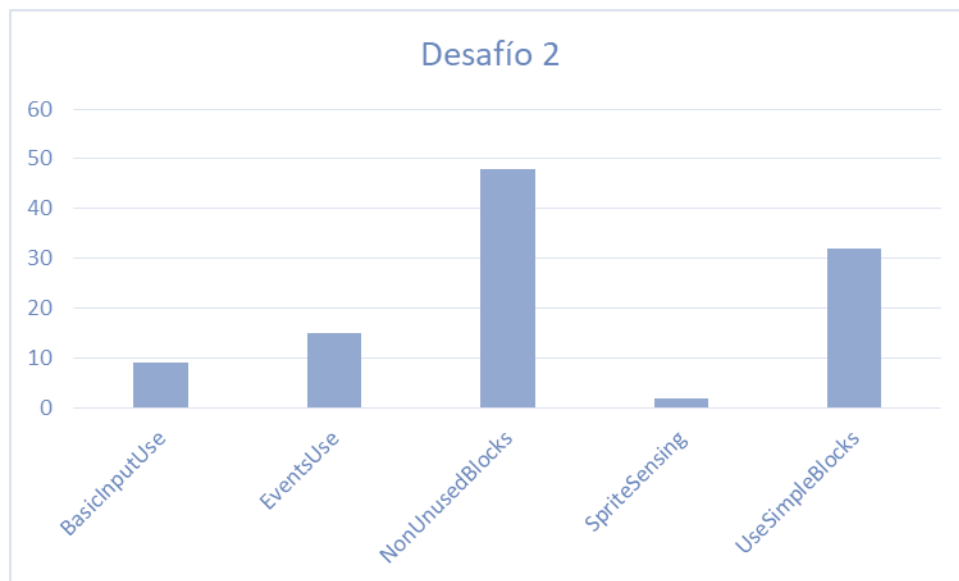


Figura 32. Resultados de desempeño desafío 2

Se puede apreciar que la figura 33 muestra que se han empleado más bloques y más variados, lo que demuestra que si están empleando más elementos de CT y por lo tanto su desarrollo se evidencia, eso se puede estimar.

6. Estudio de caso

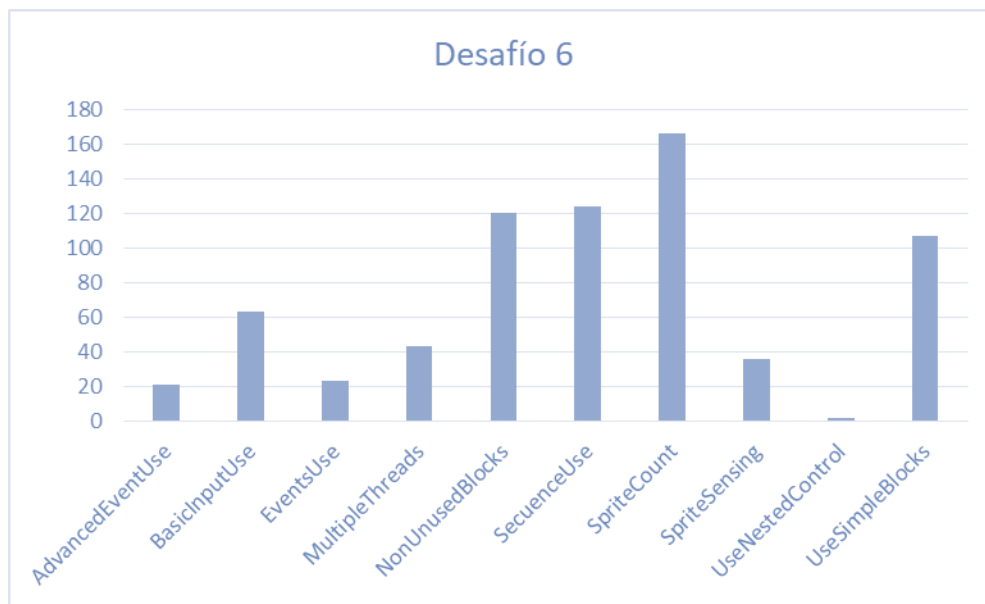


Figura 33. Resultados de desempeño desafío 6

En la figura 34 se muestran los resultados del desafío 13, este desafío fue resuelto por 4 participantes, sin embargo no todos emplearon los bloques requeridos, se puede notar que cuando se incorporan por primera vez bloques como clones, que son más complejos en su uso, los estudiantes tardan en entenderlos y usarlos.

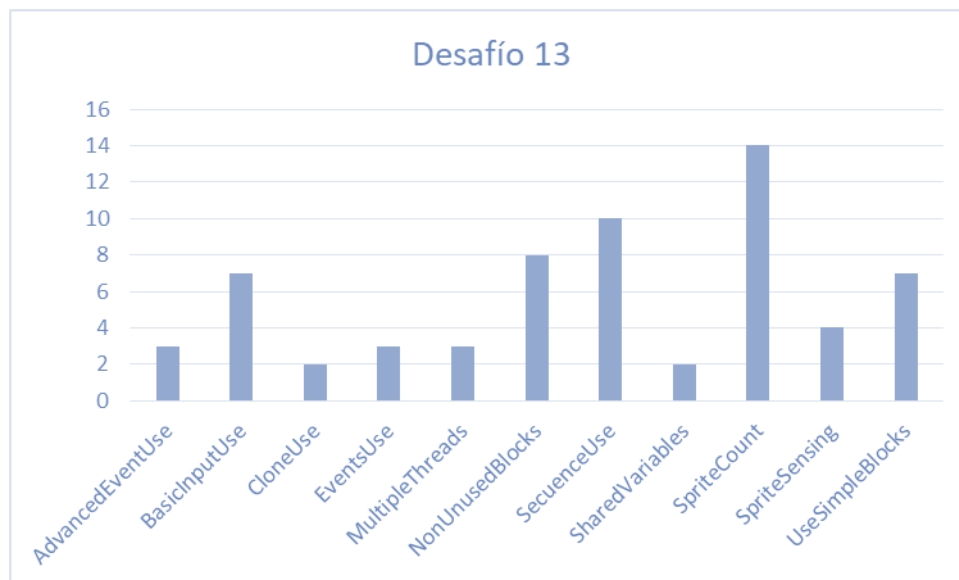


Figura 34. Resultados de desempeño desafío 13

La información de clasificación se puede ver en la tabla 79, donde 8 estudiantes tuvieron una calificación Excelente debido a la correlación de sus ejercicios con respecto a los criterios seleccionados para su evaluación. 14 estudiantes Buenos, 14 aceptables y 4 deficientes, de acuerdo a la clasificación planteada, lo que nos demuestra que la gran mayoría se encuentra en un desempeño medio. No se presentaron clasificaciones nulas, puede ser debido a que el docente estuvo pendiente todo el tiempo y no permitió que ningún estudiante dejara de realizar los desafíos.

Tabla 79. Clasificación de estudiantes

Estudiante	Clasificación	Estudiante	Clasificación
Estudiante 1	E	Estudiante 21	B
Estudiante 2	B	Estudiante 22	B
Estudiante 3	E	Estudiante 23	E
Estudiante 4	B	Estudiante 24	E
Estudiante 5	B	Estudiante 25	A
Estudiante 6	B	Estudiante 26	A
Estudiante 7	E	Estudiante 27	B
Estudiante 8	A	Estudiante 28	B
Estudiante 9	B	Estudiante 29	A
Estudiante 10	D	Estudiante 30	B
Estudiante 11	B	Estudiante 31	A
Estudiante 12	D	Estudiante 32	A
Estudiante 13	B	Estudiante 33	E
Estudiante 14	A	Estudiante 34	E
Estudiante 15	E	Estudiante 35	B
Estudiante 16	A	Estudiante 36	A
Estudiante 17	A	Estudiante 37	A
Estudiante 18	D	Estudiante 38	A
Estudiante 19	D	Estudiante 39	A
Estudiante 20	B	Estudiante 40	A

6.4. Conclusiones

El modelo planteado proporciona los pasos necesarios para llevar a cabo un proceso de evaluación continuo y apropiado. Los resultados presentados revelan que si se puede estimar un desarrollo de CT en los estudiantes teniendo en cuenta las características del modelo planteadas. Es importante considerar, desafíos acorde con el nivel de desempeño de los estudiantes en cuanto a CT y a su nivel de conocimiento del área de conocimiento seleccionada.

A pesar de que los estudiantes se encuentran diferente nivel de escolaridad (octavo y noveno), no se apreciaron diferencias significativas entre los resultados de desempeño de los grupos.

Es difícil apreciar las diferencias de los desempeños entre los géneros, ya que la relación entre hombres y mujeres es de 3:1. Sin embargo, esto nos arroja una información muy importante y es que, debido a que la institución donde se realizó el estudio de caso es una institución que instruye a los alumnos en competencias técnicas y se evidencia que la mayoría son hombres, se puede notar que hay una mayor incidencia para que los hombres se interesen por este tipo de prácticas como la programación.

Se revisaron los resultados de los estudiantes que obtuvieron una clasificación alta y se encontró que muchos de ellos emplean el computador principalmente para jugar videojuegos, a diferencia de sus compañeros que prefieren las redes sociales.

La clasificación brinda información útil para conocer los estudiantes que están codificando en relación con lo esperado y aquellos que no.

Al mejorar las herramientas de evaluación, los docentes tienen la capacidad de evaluar las soluciones de sus estudiantes de manera válida y fiable.

Al seleccionar los factores a evaluar y definir el nivel de complejidad de los desafíos, se logra que la evaluación a los estudiantes este acorde con los objetivos planteados.

Se requiere de varias sesiones para lograr estimar todos los criterios mínimos de CT, los estudiantes toman tiempo resolviendo los desafíos incluso cuando se proporciona un escenario.

Capítulo 7

7. Conclusiones y trabajo futuro

7.1. Conclusiones

- El pensamiento computacional es una habilidad esencial, por esta razón es importante que se desarrolle desde edades tempranas, para esto se requiere que los docentes tengan un conocimiento adecuado para incorporarlo en su enseñanza e incluirlo de manera transversal a diferentes áreas del conocimiento, así se fortalece la competencia y se adquiere un aprendizaje en estas áreas.
- Para la evaluación de CT, es necesario soportar al docente con herramientas que le permitan determinar las actividades a realizar por los estudiantes y tener un seguimiento de su progreso. El rol del docente debe tener la posibilidad de determinar y especificar lo que sus estudiantes van evidenciar, el qué y el cómo.
- El proceso de evaluación iterativa y embebida en las sesiones de clase ofrecen características de evaluación formativa, al igual que la autoevaluación y realimentación continúa, lo que facilita la detección temprana de dificultades en los estudiantes, de acuerdo con los conocimientos y habilidades.
- La evaluación de los criterios mínimos debe hacerse considerando aquellos que el docente desea evaluar, ya que depende del nivel de escolaridad y la actividad específica a desarrollar, el docente debe tener el control de las decisiones de evaluación.

- La evaluación del aprendizaje no se debe centrar en una tarea específica, debe considerar el proceso que lleva el estudiante, debe garantizar continuidad y una realimentación apropiada y acorde con los objetivos de aprendizaje.
- Incorporar prácticas de programación en las sesiones de clase que ofrezcan una alternativa adicional para evaluar los conocimientos de los estudiantes, contribuir con el fortalecimiento de habilidades como CT y preparar a los estudiantes para ser parte de la sociedad actual, la cual exige la comprensión y el desarrollo de habilidades computacionales en las personas.
- La caracterización de los criterios mínimos brinda información útil para conocer los factores a evaluar. Lo que potencialmente facilita la incorporación de CT en los procesos de evaluación, ya que permite el entendimiento de la competencia.
- Para la estimación del desarrollo de CT se deben considerar la mayoría posible de las diferentes características que lo componen, es decir de todos los elementos que lo definen, ya que al no considerarlos se pierde información valiosa acerca del desarrollo de CT en los estudiantes.
- Se recomiendan evaluaciones continuas, ordenadas, sistemáticas e integradas a las sesiones de clase, con el fin de detectar las falencias y logros que permitan realimentación oportuna y apropiada.
- El constante análisis de los resultados propuesto en la fase de análisis, facilita la detección temprana de dificultades y fortalezas en los estudiantes.
- Debido a la variedad de elementos que componen CT, es pertinente garantizar la utilización de diversos mecanismos de evaluación que permitan la recolección de diferentes tipos de datos.

7.2.Trabajos futuros

- Proponer y seguir un plan de mantenimiento y mejora de la aplicación, con el objetivo de optimizar y adaptar la aplicación a diferentes necesidades.
- Incluir en la aplicación todas las vistas propuestas en el anexo A., para la visualización de resultados.
- Evaluar las interacciones de los estudiantes en la comunidad Scratch, lo que brinda información acerca de la comunicación, colaboración y trabajo en equipo, potencialmente también brinda información acerca de prácticas y perspectivas de CT.

7. Conclusiones y trabajos futuros

- Incluir un mecanismo que permita a los estudiantes llevar registros anecdóticos para el desarrollo de proyectos de larga duración. Se propone, incorporar palabras clave las cuales pueden ser arrastradas por los estudiantes y pegadas en un tablero de tareas dividido en tres partes: Por realizar, En progreso, Listo, con el objetivo de facilitar su revisión de una manera automatizada.
- Incluir mecanismos como “*eye tracking*” u observación indirecta por videograbación que permiten revisar detalladamente las prácticas de programación, lo que brinda una opción que requiere de tiempo y esfuerzo pero elimina la subjetividad de la evaluación, brindando mayor validez.
- Emplear el modelo de evaluación propuesto para estimar el desarrollo de habilidades diferentes a CT, como creatividad que se puede evaluar en el diseño de escenarios.
- Ampliar el alcance de la solución propuesta, con el fin de abarcar diferentes plataformas para el desarrollo de soluciones.

Bibliografía

- [1] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [2] S. Grover and R. Pea, "Computational thinking in K-12: A review of the state of the field," *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.
- [3] J. M. Wing, "Computational thinking: What and why?," *theLink - The Magazine of the Varnege Mellon University School of Computer Science*, 2010.
- [4] J. Wing, "Computational thinking benefits society," *J. Comput. Sci. Coll.*, vol. 24, no. 6, pp. 6–7, 2014.
- [5] D. Riley and K. A. Hunt, *Computational thinking for the modern problem solver*. Chapman and Hall/CRC, 2014.
- [6] ISTE and CSTA, "Operational Definition of Computational Thinking," *Report*, p. 1030054, 2011.
- [7] P. Felicia, *Developments in Current Game-Based Learning Design and Deployment*. IGI Global, 2012.
- [8] S. Grover, S. Cooper, and R. Pea, "Assessing computational learning in K-12," *Innov. Technol. Comput. Sci. Educ. - ITiCSE*, pp. 57–62, 2014.
- [9] F. Kalelioğlu, Y. Gülbahar, and V. Kukul, "A framework for computational thinking based on a systematic research review," *Balt. J. Mod. Comput.*, vol. 4, no. 3, pp. 583–596, 2016.
- [10] S. Atmatzidou and S. Demetriadis, "Advancing Students' Computational Thinking Skills Through Educational Robotics," *Robot. Auton. Syst.*, vol. 75, no. PB, pp. 661–670, 2016.
- [11] D. Weintrop *et al.*, "Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 136 LNICST, pp. 22–25, 2014.
- [12] M. Román-gonzález, "Computational Thinking Test: Design Guidelines and Content Validation Computational Thinking Test : Design Guidelines and Content Validation," *Proc. EDULEARN15 Conf.*, no. March, pp. 2436–2444, 2016.
- [13] B. Wolz, U., Hallberg, C., & Taylor, "Scrape: A tool for visualizing the code of Scratch programs -," in *SIGCSE '11 - Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, 2011.
- [14] B. Boe, C. Hill, M. Len, G. Dreschler, P. Conrad, and D. Franklin, "Hairball: Lint-inspired Static Analysis of Scratch Projects," *Proceeding 44th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 215–220, 2013.
- [15] J. Moreno-León, G. Robles, and M. Román-González, "Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking," *Rev.*

- Educ. a Distancia*, vol. 46, no. 46, 2015.
- [16] K. H. Koh, A. Basawapatna, H. Nickerson, and A. Repenning, "Real time assessment of computational thinking," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, pp. 49–52, 2014.
- [17] J. M. Wing, "Computational Thinking: What and Why?," *thelink - Magazne Varnege Mellon Univ. Sch. Comput. Sci.*, no. March 2006, pp. 1–6, 2010.
- [18] L. Werner, J. Denner, and S. Campe, "The Fairy Performance Assessment: Measuring Computational Thinking in Middle School," *Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ. - SIGCSE '12*, pp. 215–220, 2012.
- [19] R. Federation, "ICT in Primary Education," vol. 1, no. i, pp. 1–136.
- [20] Ministerio de Educación Nacional (MEN), "Educación pertinente para un ciudadano del siglo XXI," *Al tablero*, no. 52, 2009.
- [21] C. Ministerio de Tecnologías de la Información y las Comunicaciones, "Cauca Vive Digital: Acercando las TIC a los caucanos," *Ministerio TIC*, 2014. [Online]. Available: <https://www.mintic.gov.co/portal/604/w3-article-7403.html>. [Accessed: 02-Feb-2019].
- [22] J. M. Wing, "Computational thinking and thinking about computing," *ppt*, vol. 366, no. 1881. p. 46, 2008.
- [23] G. Exploring Computational Thinking, "Computational thinking concepts guide." [Online]. Available: g.co/exploringCT. [Accessed: 02-Feb-2019].
- [24] The Royal Society, "Shut down or restart? The way forward for computing in UK schools," *Technology*, no. January, pp. 1–122, 2012.
- [25] Computer Science Teachers Association (CSTA) and I. S. for T. E. (ISTE), *Computational thinking in K–12 education leadership toolkit*, 1st. ed. 2011.
- [26] Computer Science Teachers Association (CSTA) and International Society for Technology Education (ISTE), *Computational thinking - Teacher resources*, 2nd. ed. 2011.
- [27] National Science Foundation (NSF), "Computer Science is for All Students." [Online]. Available: https://www.nsf.gov/news/special_reports/csed/index.jsp. [Accessed: 02-Feb-2019].
- [28] Y. Kafai and Q. Burke, "Computer programming goes back to school," *Phi Delta Kappan*, vol. 95, no. 1, pp. 61–65, 2013.
- [29] M. Resnick *et al.*, "Scratch: Programming for All," *Commun. ACM*, vol. 52, no. 11, p. 60, 2009.
- [30] I. Lee *et al.*, "Computational thinking for youth in practice," *ACM Inroads*, vol. 2, no. 1, p. 32, 2011.
- [31] M. Scriven, "The methodology of evaluation," in *Perspectives of curriculum evaluation*, R. W. Tyler, R. . Gagne, and M. Scriven, Eds. Chicago, IL: Rand McNally, 1967, pp. 39–83.
- [32] J. M. Álvarez Méndez, *Evaluar para aprender, examinar para excluir*. Madrid: Morata, 2001.
- [33] R. E. Bennett, "Formative assessment: A critical review," *Assess. Educ. Princ. Policy Pract.*, vol. 18, no. 1, pp. 5–25, 2011.
- [34] S. C. Arredondo and J. C. Diago, *Evaluación educativa de aprendizajes y competencias*. Pearson Hall, 2010.
- [35] J. Dolin, P. Black, W. Harlen, and A. Tiberghien, "Exploring Relations Between

Bibliografía

- Formative and Summative Assessment,” in *The Wiley Handbook of Personality Assessment*, S. Johar and U. Kumar, Eds. 2018, pp. 53–80.
- [36] M. de E. Nacional, “Evaluación en el aula,” 2012. [Online]. Available: <http://www.colombiaaprende.edu.co/html/home/1592/article-237450.html>. [Accessed: 02-Feb-2019].
- [37] Ministerio de Educación Nacional, *Decreto No. 1290*, no. 1290. Colombia, 2009, p. 5.
- [38] D. D. Dixson and F. C. Worrell, “Formative and Summative Assessment in the Classroom,” *Theory Pract.*, vol. 55, no. 2, pp. 153–159, 2016.
- [39] D. Sadler, “Formative assessment and the design of instructional systems,” *Instr. Sci.*, vol. 18, pp. 119–144, 1989.
- [40] D. Nicol and D. MacFarlane-Dick, “Formative assessment and selfregulated learning: A model and seven principles of good feedback practice,” *Stud. High. Educ.*, vol. 31, no. 2, pp. 199–218, 2006.
- [41] R. J. Mislevy, R. G. Almond, and J. F. Lukas, “A Brief Introduction to Evidence-Centered Design,” *Russell J. Bertrand Russell Arch.*, vol. 1522, no. July, pp. 1–37, 2003.
- [42] V. J. Shute, “Stealth Assessment in Computer-Based Games To Support Learning,” *Comput. Games Instr.*, pp. 503–524, 2011.
- [43] M. Bienkowski, E. Snow, D. Rutstein, S. Grover, and S. R. I. International, “Assessment Design Patterns for Computational Thinking Practices in Secondary Computer Science: Authors Acknowledgments,” no. December, 2015.
- [44] V. J. Shute and F. Ke, “Assessment in Game-Based Learning,” pp. 43–58, 2012.
- [45] E. Snow, C. Tate, D. Rutstein, and M. Bienkowski, “Assessment design patterns for computational thinking practices in exploring computer science,” Menlo Park, CA, 2017.
- [46] S. Papert, *Mindstorms: Children, computers and powerful ideas*, 1st ed. New York: Basic Books, Inc., 1980.
- [47] C. M. Lewis, “How Programming Environment Shapes Perception, Learning and Goals: Logo vs . Scratch,” *Sigcse '10*, pp. 346–350, 2010.
- [48] Lifelong Kindergarten. MIT Media Lab, “Scratch - Imagine, Program, Share,” *Scratch statistics*. [Online]. Available: <https://scratch.mit.edu/statistics/>. [Accessed: 13-Sep-2018].
- [49] J. Denner, L. Werner, and E. Ortiz, “Computer Games Created by Middle School Girls: Can They Be Used to Measure Understanding of Computer Science Concepts?,” *Comput. Educ.*, vol. 58, no. 1, pp. 240–249, 2012.
- [50] C. W. Herbert, *An introduction to programming using Alice 2.2*, 2nd. ed. Boston, Massachusetts, USA, 2013.
- [51] J. Solís-Martínez, J. P. Espada, N. García-Menéndez, B. C. Pelayo G-Bustelo, and J. M. Cueva Lovelle, “VGPM: Using business process modeling for videogame modeling and code generation in multiple platforms,” *Comput. Stand. Interfaces*, vol. 42, pp. 42–52, 2015.
- [52] D. Bau, D. A. Bau, M. Dawson, and C. S. Pickens, “Pencil Code : Block Code for a Text World,” *14th Int. Conf. Interact. Des. Child.*, pp. 445–448, 2015.
- [53] K. Stolee and T. Fristoe, “Expressing computer science concepts through Kodu

- game lab,” *Sigcse’11*, pp. 99–104, 2011.
- [54] A. R. Basawapatna, A. Repenning, and C. H. Lewis, “The Simulation Creation Toolkit: An Initial Exploration into Making Programming Accessible While Preserving Computational Thinking,” in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 501–506.
- [55] K. Howland and J. Good, “Learning to communicate computationally with Flip: A bi-modal programming language for game creation,” *Comput. Educ.*, vol. 80, pp. 224–240, 2015.
- [56] D. S. Touretzky, D. Marghitu, S. Ludi, D. Bernstein, and L. Ni, “Accelerating K-12 Computational Thinking Using Scaffolding, Staging, and Abstraction,” *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ. - SIGCSE ’13*, p. 609, 2013.
- [57] T. Turchi and A. Malizia, “Fostering computational thinking skills with a tangible blocks programming environment,” in *IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2016, vol. 2016-Novem, pp. 232–233.
- [58] Massachusetts Institute of Technology, “App Inventor web site,” *Center for Mobile Learning*. [Online]. Available: <http://appinventor.mit.edu>. [Accessed: 02-Feb-2019].
- [59] “Code.org.” [Online]. Available: <https://code.org>. [Accessed: 02-Feb-2019].
- [60] S. Grover and R. Pea, “Using a discourse-intensive pedagogy and android’s app inventor for introducing computational concepts to middle school students,” *SIGCSE 2013 - Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 723–728, 2013.
- [61] A. Repenning, “Programming goes back to school,” *Commun. ACM*, vol. 55, no. 5, p. 38, 2012.
- [62] C. M. Lewis, “How Programming Environment Shapes Perception, Learning and Goals: Logo vs . Scratch,” in *Sigcse*, 2010, pp. 346–350.
- [63] H. Swanson, G. Anton, C. Bain, M. Horn, and U. Wilensky, “Computational thinking in the science classroom,” in *Conference Proceedings of International Conference on Computational Thinking Education*, 2017.
- [64] U. Wilensky, “NetLogo,” *Center for Connected Learning and Computer-Based Modeling, Northwestern University*. [Online]. Available: <http://ccl.northwestern.edu/netlogo/>. [Accessed: 02-Feb-2019].
- [65] J. Denner, L. Werner, and E. Ortiz, “Computer games created by middle school girls: can they be used to measure understanding of computer science concepts?,” *Comput. Educ.*, vol. 58, no. 1, pp. 240–249, 2012.
- [66] J. Denner and L. Werner, “Measuring computational thinking in middle school using game programming,” in *AERA*, 2011.
- [67] D. S. Touretzky, D. Marghitu, S. Ludi, D. Bernstein, and L. Ni, “Accelerating K-12 computational thinking using scaffolding, staging, and abstraction,” *ACM Tech. Symp. Comput. Sci. Educ. - SIGCSE*, p. 609, 2013.
- [68] L. Benotti, M. C. Martinez, and F. Schapachnik, “A tool for introducing computer science with automatic formative assessment,” *IEEE Trans. Learn. Technol.*, vol. XX, no. X, pp. 1–1, 2017.
- [69] A. Ioannidou, V. Bennett, A. Repenning, K. H. Koh, and A. Basawapatna, “Computational thinking patterns,” in *Annual Meeting of the American Educational Research Association (AERA)*, 2011, pp. 1–15.

Bibliografía

- [70] K. S. Marshall, "Was that CT? Assessing computational thinking patterns through video-based prompts," in *Annual Meeting of the American Educational Research Association*, 2011, p. 12.
- [71] K. Howland and J. Good, "Learning to communicate computationally with flip: a bi-modal programming language for game creation," *Comput. Educ.*, vol. 80, pp. 224–240, 2015.
- [72] V. Chaudhary, V. Agarwal, P. Sureka, and A. Sureka, "An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit," *Int. Conf. Technol. Educ.*, pp. 1–4, 2016.
- [73] G. Chen, J. Shen, L. Barth-Cohen, S. Jiang, X. Huang, and M. Eltoukhy, "Assessing elementary students' computational thinking in everyday reasoning and robotics programming," *Comput. Educ.*, vol. 109, pp. 162–175, 2017.
- [74] I. Fronza, N. El Ioini, and L. Corral, "Computational thinking through mobile programming," in *13th International Conference, MobiWIS*, 2016, p. 448.
- [75] R. Gopalan, "An Assessment Tool to Analyze Code Written in App Inventor," *All Grad. Plan B other Reports*, p. 1324, 2018.
- [76] S. Y. Lye and J. H. L. Koh, "Case Studies of Elementary Children's Engagement in Computational Thinking Through Scratch Programming," in *Computational Thinking in the STEM Disciplines*, Cham: Springer International Publishing, 2018, pp. 227–251.
- [77] S. Papavlasopoulou, M. N. Giannakos, and L. Jaccheri, "Discovering children's competences in coding through the analysis of Scratch projects," in *IEEE Global Engineering Education Conference (EDUCON)*, 2018, pp. 1133–1139.
- [78] M. P. Palencia, "El pensamiento computacional y la resolución de problemas: una apuesta pedagógica en el siglo XXI," *Sist. Universidades estatales del Caribe*, pp. 64–73, 2017.
- [79] N. van Es and J. Jeuring, "Designing and comparing two Scratch-based teaching approaches for students aged 10-12 years," in *Proceedings of Koli Calling*, 2017.
- [80] A. K. Hoover *et al.*, "Assessing computational thinking in students' game designs," in *CHI Play*, 2016, pp. 173–179.
- [81] J. Moreno-León, G. Robles, and M. Román-González, "Code to Learn: Where Does It Belong in the K-12 Curriculum?," *Jite.Org*, vol. 15, pp. 283–303, 2016.
- [82] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," *Annu. Am. Educ. Res. Assoc. Meet. Vancouver, BC, Canada*, pp. 1–25, 2012.
- [83] A. Ioannidou, V. Bennett, A. Repenning, K. H. Koh, and A. Basawapatna, "Computational Thinking Patterns.," *Proc. 18th ACM Conf. Innov. Technol. Comput. Sci. Educ. ACM*, 2013., vol. 1, no. 2, pp. 38–43, 2011.
- [84] T. Karvounidis, I. Argyriou, A. Ladas, and C. Douligeris, "A design and evaluation framework for visual programming codes," in *Global Engineering Education Conference*, 2017.
- [85] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, "A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming," *Procedia - Soc. Behav. Sci.*, vol. 47, pp. 1991–1999, 2012.
- [86] P. Gouws, L. A., Bradshaw, K., & Wentworth, "Computational thinking in educational activities: An evaluation of the educational game light-bot," *Proc.*

- 18th ACM Conf. Innov. Technol. Comput. Sci. Educ. 10-15., 2013.
- [87] P. Gouws, L., Bradshaw, K., & Wentworth, “First year student performance in a test for computational thinking.,” 2013.
- [88] M. Román-González, J. C. Pérez-González, and C. Jiménez-Fernández, “Test de Pensamiento Computacional : diseño y psicometría general,” *III Congr. Int. sobre Aprendizaje, Innovación y Compet. (CINAIC 2015)*, no. Cinaic, 2015.
- [89] U. L. Ling, T. C. Saibin, N. Naharu, J. Labadin, and N. A. Aziz, “An evaluation tool to measure computational thinking skills: Pilot investigation,” *Natl. Acad. Manag. Staff Cult. ARTS Her.*, no. 1, pp. 606–614, 2018.
- [90] G. Ota, Y. Morimoto, and H. Kato, “Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts,” *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, vol. November, pp. 238–239, 2016.
- [91] Z. Chang and Y. Sun, “Scratch Analysis Tool (SAT): A modern scratch project analysis tool based on antlr to assess computational thinking skills,” *14th Int. Wirel. Commun. Mob. Comput. Conf.*, pp. 950–955, 2018.
- [92] A. E. Tew and M. Guzdial, “The FCS1 : A Language Independent Assessment of CS1 Knowledge,” in *ACM technical symposium on computer science education*, 2011, pp. 111–116.
- [93] D. Weintrop *et al.*, “Defining computational thinking for mathematics and science classrooms,” *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2015.
- [94] C. E. Serrano, *Modelo integral para el profesional en ingeniería*. Universidad del Cauca, 2005.
- [95] M. Román-González, J. Moreno-León, and G. Robles, “Complementary tools for computational thinking assessment,” *Int. Conf. Comput. Think. Educ.*, no. July, p. in press, 2017.
- [96] V. J. Shute, C. Sun, and J. Asbell-Clarke, “Demystifying computational thinking,” *Educ. Res. Rev.*, vol. 22, pp. 142–158, 2017.
- [97] M. U. Bers, *Coding as a playground: Programming and computational thinking in the early childhood classroom*. 2017.
- [98] T.-C. Hsu, S.-C. Chang, and Y.-T. Hung, “How to learn and how to teach computational thinking: Suggestions based on a review of academic publications,” *Comput. Educ.*, 2018.
- [99] V. Barr and C. Stephenson, “Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?,” *ACM Inroads*, vol. 2, no. 1, pp. 48–54, 2011.
- [100] Computing At School, “Computational thinking,” 2011.
- [101] A. Csizmadia *et al.*, “Computational thinking - A guide for teachers,” 2015.
- [102] C. B. A. P. Program, “AP Computer Science Principles, Course and Exam Description,” New York, NY, USA, 2017.
- [103] Queen Mary - University of London, “Computer science for fun.” [Online]. Available: <http://www.cs4fn.org/computationalthinking/>. [Accessed: 02-Feb-2019].
- [104] E. Hunsaker, “Teaching computational thinking - Badge guide,” Provo, Utah, EE. UU, 2018.
- [105] Microsoft, “Computational Thinking and its importance in education,” 2018.

Bibliografía

- [Online]. Available: <https://education.microsoft.com/courses-and-resources/courses/computational-thinking-and-its-importance-in-education>. [Accessed: 02-Feb-2019].
- [106] P. Curzon, M. Dorling, T. Ng, P. Curzon, C. Selby, and J. Woollard, "Developing computational thinking in the classroom: A framework," 2014.
- [107] C. Selby, M. Dorling, and J. Woollard, "Evidence of assessing computational thinking," *IFIP TC3 Work. Conf. a New Cult. Learn. Comput. Next Gener.*, pp. 232–242, 2015.
- [108] D. Weintrop *et al.*, "Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology," *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2016.
- [109] J. Moreno-León, G. Robles, and M. Román-González, "Dr. Scratch: Análisis automático de proyectos scratch para evaluar y fomentar el pensamiento computacional," *Rev. Educ. a Distancia*, vol. 46, no. 10, pp. 1–23, 2015.
- [110] V. Dagiene and G. Futschek, "Bebras - International challenge on informatics and computational thinking." [Online]. Available: <https://www.bebas.org>. [Accessed: 02-Feb-2019].
- [111] V. Dagiene and G. Futschek, "Bebras international contest on informatics and computer literacy: Criteria for good tasks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5090 LNCS, pp. 19–30, 2008.
- [112] T. Y. Lee, M. L. Mauriello, J. Ahn, and B. B. Bederson, "CTArcade: Computational thinking with games in school age children," *Int. J. Child-Computer Interact.*, vol. 2, no. 1, pp. 26–33, 2014.
- [113] S. Atmatzidou and S. Demetriadis, "How to support students' computational thinking skills in educational robotics activities," *Proc. 4th Int. Work. Teach. Robot. Teach. with Robot. 5th Int. Conf. Robot. Educ.*, pp. 43–50, 2014.
- [114] C. Kazimoglu, M. Kiernan, L. Bacon, and L. MacKinnon, "Learning programming at the computational thinking level via digital game-play," *Procedia Comput. Sci.*, vol. 9, no. 0, pp. 522–531, 2012.
- [115] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, "A serious game for developing computational thinking and learning introductory computer programming," *Procedia - Soc. Behav. Sci.*, vol. 47, pp. 1991–1999, 2012.
- [116] K. Boom, M. Bower, J. Siemon, and A. Scholkmann, "Relationship between computational thinking and a measure of intelligence as a general problem-solving ability," *23rd Annu. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, pp. 206–211, 2018.
- [117] M. L. Wu, "Educational game design as gateway for operationalizing computational thinking skills among middle school students," *Ccsenet.Org*, vol. 11, no. 4, pp. 15–28, 2018.
- [118] J. L. Weese and R. Feldhausen, "STEM outreach: Assessing computational thinking and problem solving," *Am. Soc. Eng. Educ.*, p. 11, 2017.
- [119] E. Rowe, J. Asbell-Clarke, S. Gasca, and K. Cunningham, "Assessing implicit computational thinking in zoombinis gameplay," *Proc. Int. Conf. Found. Digit. Games - FDG '17*, no. Figure 1, pp. 1–4, 2017.
- [120] V. KUKUL and S. KARATAŞ, "Computational Thinking Self-Efficacy Scale:

- Development, Validity and Reliability,” *Informatics Educ.*, vol. 18, no. 1, pp. 151–164, 2019.
- [121] E. Relkin, “Assessing young children’s computational thinking abilities,” Tufts University, 2018.
- [122] T. Djambong and V. Freiman, “Task-based assessment of students’ computational thinking skills developed through visual programming or tangible coding environments,” in *13th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2016)*, 2016, pp. 41–52.
- [123] S. Basu *et al.*, “A principled approach to designing assessments that integrate science and computational thinking evidence-centered design,” no. June, pp. 384–391, 2018.
- [124] Y. Allsop, “Assessing computational thinking process using a multiple evaluation approach,” *Int. J. Child-Computer Interact.*, vol. 19, pp. 30–55, 2019.
- [125] B. Zhong, Q. Wang, J. Chen, and Y. Li, “An exploration of three-dimensional integrated assessment for computational thinking,” *J. Educ. Comput. Res.*, vol. 53, no. 4, pp. 562–590, 2016.
- [126] D. Weintrop *et al.*, “Interactive assessment tools for computational thinking in high school stem classrooms,” in *6th International Conference, INTETAIN 2014 Chicago, IL, USA, July 9–11, 2014*, 2014, pp. 22–25.
- [127] H. M. Jamil, “Smart assessment of and tutoring for computational thinking MOOC assignments using MindReader,” *arXiv Prepr.*, 2017.
- [128] R. Zúñiga and J. Hurtado, “Discovering the mechanisms of abstraction in the performance of work teams in children to solve computational problems,” *Sist. Telemática*, vol. 14, no. 36, pp. 69–87, 2016.
- [129] J. M. Wing, “Computational thinking and thinking about computing,” *Philos. Trans. R. Soc.*, vol. 366, no. July, pp. 3717–3725, 2008.
- [130] M. Romero, A. Lepage, and B. Lille, “Computational thinking development through creative programming in higher education,” *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, 2017.
- [131] M. Resnick and N. Rusk, “New pathways into data science extending the scratch programming language to enable youth to analyze and visualize their own learning,” 2013. [Online]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwit9qTAlrjAhVEtVMKHfrACoQFggnMAA&url=https%3A%2F%2Fwww.hub.ki%2Fgroups%2Fresources%2Fwiki%2FNewPathways2%3Faction%3Dpdf&usg=AOvVaw2OEBYeZFr-JZSZpEqUemcE>.
- [132] T. S. Barcelos, I. F. De Educação, T. De São, S. Paulo, and R. Villarroel, “Mathematics learning through computational thinking activities: a systematic literature review,” vol. 24, no. 7, pp. 815–845, 2018.
- [133] G. Gadanidis, “Artificial intelligence , computational thinking , and mathematics education,” in *ICICTE*, 2016, pp. 83–91.
- [134] D. Moursund, *Computational thinking and math maturity: Improving math education in k-8 schools*. Eugene, Oregon: University of Oregon, 2006.
- [135] P. C. Ncr *et al.*, “Step-by-step data mining guide,” *SPSS inc*, vol. 78, pp. 1–78, 2000.
- [136] R. G. Almond, R. J. Mislevy, L. S. Steinberg, D. Yan, and D. M. Williamson, “An

Bibliografía

- Introduction to Evidence-Centered Design,” in *Science+Business*, 2015, pp. 19–40.
- [137] M. Benson, D. Fay, K. L. Kunze, R. J. Mislevy, and J. Behrens, “Putting ECD into Practice: The Interplay of Theory and Data in Evidence Models within a Digital Learning Environment,” *J. Educ. Data Min.*, vol. 4, no. 1, pp. 49–110, 2012.
- [138] V. J. Shute and M. Ventura, *Stealth assessment - Measuring and supporting learning in video games*. MIT Press books, 2013.
- [139] M. G. De Oliveira, P. Marques Ciarelli, and E. Oliveira, “Recommendation of programming activities by multi-label classification for a formative assessment of students,” *Expert Syst. Appl.*, vol. 40, pp. 6641–6651, 2013.
- [140] Waikato University, “Weka 3: Machine Learning Software in Java.” [Online]. Available: <https://www.cs.waikato.ac.nz/ml/weka/>.
- [141] U. Fayyad, “Bayesian Networks for Data Mining,” *Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 79–119, 1997.
- [142] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009.

Anexos

Anexo A

Tabla a 1. Lista de elementos CT – sin seleccionar	211
Tabla a 2. Mapeo de términos de los elementos CT por núcleos temáticos.....	212
Tabla a 3. Caracterización de criterios mínimos- A nivel de marcos de referencia	213
Tabla a 4. Caracterización de criterios mínimos- A nivel de herramientas.....	214
Tabla a 5. Caracterización de criterios mínimos- A nivel de implementaciones.....	215
Tabla a 6. Caracterización de mecanismos de evaluación.....	216
Tabla a 7. Glosario de definiciones de los elementos de CT	217

Anexo B

Tabla b 1. Bloques de Scratch	221
Tabla b 2. Información de los estudiantes recolectada en la introducción	224
Tabla b 3. Propuesta de las opciones de visualización de resultados para los docentes	224
Tabla b 4. Ejemplo 2. Tabla de presentación de calificación	226
Tabla b 5. Ejemplo 1. Tabla de presentación de calificación	226
Tabla b 6. Plan del proyecto de minería de datos	226
Tabla b 7. Descripción de los datos entregados por el estudiante– Fase Comprensión de la información	227
Tabla b 8. Variables seleccionadas por el docente– Fase Comprensión de la información. 229	
Tabla b 9. Dataset final	230
Tabla b 10. Resultado - Redes Bayesianas	230
Tabla b 11. Resultado – Redes neuronales	231
Tabla b 12. Resultado Tasa nula	231
Figura b 1. Grafo generado - Red Bayesiana.....	231
Figura b 2. Errores de clasificación - Red Bayesiana	231
Figura b 3. Errores de clasificación - Red Neuronal	232

Anexo C

Tabla c 1. Relación de los nombres de las variables definición de criterios y base de datos	233
Tabla c 2. Modelo lógico de la base de datos.....	234

Anexo D

Tabla d 1. Información recolectada por HERA-CT en el registro de los estudiantes al sistema	235
Tabla d 2. Problemas o desafíos propuestos en el estudio de caso	236
Tabla d 3. Relación de los desafíos propuestos en el estudio de caso y los elementos de CT	237
Tabla d 4. Consentimiento informado	237
Tabla d 5. Datos recolectados durante el registro al prototipo y empleado en el estudio de caso	239
Tabla d 6. Discretización de los elementos de CT relacionados con las preguntas en la entrevista.....	240

Anexo A

A. Tablas de apoyo en el proceso de caracterización – capítulo 3

En este Anexo están consignadas algunas de las tablas que soportaron el proceso de Caracterización de elementos CT.

Tabla a 1. Lista de elementos CT – sin seleccionar

No.	Elemento CT
1	Abstracción
2	Generalización de patrones
3	Reconocimiento de patrones
4	Modularización, Modularidad
5	Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos
6	Modelado
7	Simulación
8	Formulación o representación de problemas
9	Descomposición, Descomposición de problemas
10	Solución y resolución de problemas
11	Declaraciones condicionales, Lógica condicional
12	Razonamiento lógico, Lógica, Pensamiento lógico
13	Paralelismo
14	Recolección de información
15	Análisis de información
16	Representación de información
17	Automatización
18	Evaluar
19	Probar
20	Depuración
21	Experimentar
22	Crear
23	Perseverar
24	Colaborar
25	Reflexionar
26	Codificar
27	Diseñar

28	Analizar
29	Aplicar
30	Secuencias
31	Bucles o ciclos
32	Eventos
33	Condicionales
34	Operadores
35	Incremental e iterativo, Iteración
36	Reutilización y remezclado
37	Expresar
38	Conectar
39	Cuestionar
40	Conexión de la computación
41	Crear artefactos computacionales
42	Analizar problemas y artefactos
43	Comunicar
44	Soluciones eficientes
45	Pensamiento científico
46	Pensamientos innovadores
47	Control de flujo
48	Interactividad del usuario, Interfaz de usuario
49	Sincronización
50	Procedimiento, Procedimiento común
51	Socialización
52	Computación distribuida
53	Instancias no articuladas
54	Pensamiento sistémico
55	Tratamiento sistemático de la información
56	Sistemas de símbolos y representaciones
57	Pensamiento iterativo, recursivo y paralelo
58	Limitaciones de eficiencia y rendimiento
59	Estructuras de Control
60	Hardware software
61	Proceso de diseño
62	Seguro, confiado
63	Ser flexible
64	Experimentar
65	Ser Perseverante
66	Transformar
67	Conectar con otras áreas
68	Planear
69	Visualizar
70	Capacidad de hacer inferencias y predicciones
71	Eficiencia y desempeño
72	Efectividad
73	Inicialización y actualización de variables

Tabla a 2. Mapeo de términos de los elementos CT por núcleos temáticos

Anexo A

No.	Elementos	Autores	Educativas	Industrias	Marcos	Herramientas	Implementaciones	Total
1	Abstracción	1	1	1	1	1	1	6
2	Generalización de patrones	1	0	1	1	1	1	5
3	Reconocimiento de patrones	0	0	1	0	1	0	2
4	Modularización, Modularidad	1	0	0	1	0	0	2
5	Algoritmos, Algoritmos y procedimientos, Diseño de algoritmos	1	1	1	1	1	1	6
6	Modelado	0	0	0	1	0	0	1
7	Simulación	0	0	1	1	0	0	2
8	Formulación o representación de problemas	1	0	0	0	0	0	1
9	Descomposición, Descomposición de problemas	1	1	1	1	1	1	6
10	Solución y resolución de problemas	0	0	0	1	0	0	1
11	Declaraciones condicionales, Lógica condicional	1	0	0	0	1	0	2
12	Razonamiento lógico, Lógica, Pensamiento lógico	0	0	0	0	1	0	1
13	Paralelismo	0	0	1	1	1	0	3
14	Recolección de información	0	0	1	1	0	1	3
15	Análisis de información	0	0	1	1	0	1	3
16	Representación de información	0	0	1	1	1	0	3
17	Automatización	0	0	1	0	0	0	1
18	Evaluar	0	1	0	1	0	0	2
19	Probar	0	0	0	1	0	0	1
20	Depuración	1	1	0	1	0	1	4
21	Experimentar	0	1	0	0	0	0	1
22	Crear	0	1	0	0	0	0	1
24	Colaborar	0	1	0	0	0	0	1
28	Analizar	0	1	0	0	0	0	1
30	Secuencias	0	0	0	1	0	0	1
31	Bucles o ciclos	0	0	0	1	0	0	1
32	Eventos	0	0	0	1	0	0	1
33	Condicionales	0	0	0	1	0	0	1
34	Operadores	0	0	0	1	0	0	1
35	Incremental e iterativo, Iteración	0	0	0	1	0	0	1
36	Reutilización y remezclado	0	0	0	1	0	0	1
37	Expresar	0	0	0	1	0	0	1
38	Conectar	0	0	0	1	0	0	1
39	Cuestionar	0	0	0	1	0	0	1
43	Comunicar	0	1	0	0	0	0	1
47	Control de flujo	0	0	0	0	1	0	1
48	Interactividad del usuario, Interfaz de usuario	0	0	0	0	1	0	1
49	Sincronización	0	0	0	0	1	0	1
54	Pensamiento sistémico	0	0	0	1	0	0	1

Tabla a 3. Caracterización de criterios mínimos- A nivel de marcos de referencia

Elementos	Criterios	Referencia
Abstracción	Reconoce que existen diferentes soluciones para el mismo problema	Referencia 2
Abstracción	Utiliza una variable y operadores relacionales dentro de un bucle para manejarlo	Referencia 2
Abstracción	Sabe que se puede utilizar un procedimiento para ocultar los detalles con una sub-solución	Referencia 2

Abstracción	Emplea correctamente las declaraciones condicionales	Referencia 2
Abstracción y modularización	Puede construir algo grande y complejo, partiendo de pequeñas partes	Referencia 1
Abstracción y modularización	Puede construir algo grande y complejo, partiendo de pequeñas partes	Referencia 3
Bucles	Repite pasos mediante el uso de bloques	Referencia 1
Bucles	Repite pasos mediante el uso de bloques	Referencia 3
Condicionales	Ejecuta acciones como consecuencia de un acontecimiento	Referencia 1
Condicionales	Ejecuta acciones como consecuencia de un acontecimiento	Referencia 3
Conectarse	Comparte con otros desarrolladores y buscar entre los diferentes proyectos	Referencia 1
Conectarse	Comparte con otros desarrolladores y buscar entre los diferentes proyectos	Referencia 3
Cuestionarse	Pregunta por lo que lo rodea y en algunos casos responder a través del diseño	Referencia 1
Cuestionarse	Pregunta por lo que lo rodea y en algunos casos responder a través del diseño	Referencia 3
Descomposición	Diseña soluciones descomponiendo problemas y soluciona cada parte	Referencia 2
Descomposición	Realiza búsquedas de información	Referencia 2
Evaluación	Analiza y evalúa la información	Referencia 2
Evaluación	Utiliza criterios para evaluar la calidad de las soluciones	Referencia 2
Evaluación	Hace juicios sobre el contenido digital cuando lo evalúa y lo reutiliza	Referencia 2
Evaluación	Reconoce a la audiencia al diseñar y crear contenido digital	Referencia 2
Evaluación	Comprende el potencial de la tecnología de la información para la colaboración	Referencia 2
Eventos	Muestra como una tarea se efectúa como una consecuencia de un acontecimiento	Referencia 1
Eventos	Muestra como una tarea se efectúa como una consecuencia de un acontecimiento	Referencia 3
Expresar	Comunica a sus compañeros o docente información de lo que está desarrollando y como lo hace	Referencia 1
Expresar	Comunica a sus compañeros o docente información de lo que está desarrollando y como lo hace	Referencia 3
Generalización	Diseña, escribe y depura programas modulares utilizando procedimientos	Referencia 2
Incremental e iterativo	Construye la solución por pequeños pasos	Referencia 1
Incremental e iterativo	Construye la solución por pequeños pasos	Referencia 3
Información	Almacenamiento, recuperación y actualización de la información	Referencia 1
Información	Almacenamiento, recuperación y actualización de la información	Referencia 3
Operadores	Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres	Referencia 1
Operadores	Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres	Referencia 3
Paralelismo	Usa secuencias ejecutándose al mismo tiempo	Referencia 1
Paralelismo	Usa secuencias ejecutándose al mismo tiempo	Referencia 3
Pruebas y depuración	Prueba y reconoce errores en las secuencias de programas y es capaz de arreglarlos	Referencia 1
Pruebas y depuración	Prueba y reconoce errores en las secuencias de programas y es capaz de arreglarlos	Referencia 3
Reutilización y remezcla	Emplea código existente en un nuevo proyecto	Referencia 1
Reutilización y remezcla	Emplea código existente en un nuevo proyecto	Referencia 3
Secuencias	Representa las soluciones como una serie de pasos	Referencia 1
Secuencias	Representa las soluciones como una serie de pasos	Referencia 3

Tabla a 4. Caracterización de criterios mínimos- A nivel de herramientas

Elementos	Criterios	Referencia
Abstracción	Descompone el problema en partes pequeñas	Referencia 6
Abstracción y descomposición del problema	Emplea más de un hilo y más de un sprite, define bloques, usa clones	Referencia 4
Abstracción y descomposición del problema	Dispone de dos o más hilos inician con el mismo bloque	Referencia 4
Adaptación al cambio	Facilita la realización cambios	Referencia 6
Análisis de información	Toma la información para manipularla, analizarla y visualizarla	Referencia 8
Bucles	Emplea secuencias de repetición	Referencia 7
Condicionales	Usa condicionales y operadores lógicos	Referencia 7
Control de flujo	Emplea bloques como repetir, por siempre y otros bucles.	Referencia 4

Anexo A

Efectividad	Cumple con lo requerido para que el programa opera apropiadamente	Referencia 6
Eventos	Determina eventos a través del uso del teclado, ratón y recursos multimedia.	Referencia 7
Facilidad de pruebas	Permite realizar corrección de los errores	Referencia 6
Información	Define variables o listas	Referencia 7
Interactividad con el usuario	Realiza una adecuada interacción con el usuario	Referencia 6
Interactividad del usuario	Brinda herramientas al usuario para interactuar con el sistema, a través del teclado y ratón.	Referencia 4
Interfaz con el usuario	Permite la interacción con el usuario a través del uso del teclado, ratón y recursos multimedia	Referencia 7
Legibilidad	Programa de tal manera que el código se entiende con facilidad.	Referencia 6
Modelado y simulación	Emplea modelos computacionales para entender conceptos y/o solucionar problemas, entiende el funcionamiento de los modelos, construye modelos o los modifica	Referencia 8
Optimización	Considero cambiar los nombres de los objetos con nombres significativos.	Referencia 6
Optimización	Dispone de los recursos correctamente, por ejemplo no hay redundancia, no hay presencia de código muerto.	Referencia 6
Paralelismo	Hace uso de condicionales y operadores lógicos	Referencia 4
Paralelismo	Reconoce la diferencia entre procesos paralelos y seriales	Referencia 6
Paralelismo	Maneja correctamente los mensajes, sincroniza el fondo	Referencia 7
Pensamiento lógico	Utiliza bloques como esperar y parar, maneja mensajes correctamente	Referencia 4
Pensamiento sistémico	Comprende las relaciones entre los sistemas, identifica, entiende y maneja la complejidad de los problemas, investiga un sistema como una unidad	Referencia 8
Procedimientos	Codifica más de un script, define bloques, usa clones	Referencia 7
Procedimientos comunes	Define scripts que comparten bloques, recursividad del uso de bloques	Referencia 7
Representación de la información	Emplea correctamente las variables, emplea listas	Referencia 6
Representación de los datos	Modifica las propiedades de los sprites, usa operaciones con variables o listas	Referencia 4
Seguridad	Inicializa las variables, termina la interacción después de un tiempo	Referencia 6
Sincronización	Emplea una comunicación entre los objetos correcta, usa comandos disponibles para ayudar al flujo del programa	Referencia 6
Solución de problemas computacionales	Selecciona herramientas computacionales efectivas, evalúa diferentes enfoques de un problema, brinda soluciones informáticas modulares a los problemas, emplea estrategias de resolución de problemas y creación de abstracciones	Referencia 8

Tabla a 5. Caracterización de criterios mínimos- A nivel de implementaciones

Elementos	Criterios	Referencia
Abstracción	Hace comentarios sobre los datos utilizados para la solución del problema.	Referencia 17
Abstracción	Es capaz de generalizar por atributos	Referencia 12
Abstracción	Intenta simplificar un problema ignorando detalles innecesarios, busca patrones	Referencia 11
Abstracción y descomposición del problema	Emplea más de un hilo y más de un sprite, define bloques, usa clones	Referencia 16
Abstracción y descomposición del problema	Dispone de dos o más hilos inician con el mismo bloque	Referencia 16
Algoritmos	Descompone un problema en partes más pequeñas.	Referencia 13
Algoritmos	Utiliza grupos de bloques como unidades en programas más grandes.	Referencia 13
Bucles	Repite pasos mediante el uso de bloques	Referencia 18
Comportamientos de aprendizaje	Trabaja con colaboración, piensa creativamente y críticamente, depura errores, modifica ideas y comunica estas ideas utilizando diferentes modos de intercambio conversacional	Referencia 18
Condicionales	Ejecuta acciones como consecuencia de un acontecimiento	Referencia 18
Control de flujo	Emplea bloques como repetir, por siempre y otros bucles.	Referencia 16
Control de flujo	Emplea bucles correctamente, responde a eventos, los comandos se ejecutan al cumplir con las condiciones esperadas	Referencia 11
Depuración	Reconoce errores en las secuencias de programas y es capaz de arreglarlos.	Referencia 13
Descomposición	Si hay problemas secundarios en el problema, puede administrar los procesos de solución de estos	Referencia 17
Descomposición de problemas	Aísla los atributos	Referencia 12

Descomposición de problemas	Fracciona un problema en varias partes y resuelve una a una	Referencia 11
Diseño de algoritmos	Puede generar secuencias de comportamientos que se repiten	Referencia 12
Estructura de control	Integra parámetros y bloques que controlan comportamientos complejos con realimentación de sensores.	Referencia 13
Eventos	Muestra como una tarea se efectúa como una consecuencia de un acontecimiento	Referencia 18
Formulación y solución de problemas	Formula problemas de forma que el computador puede ayudar a resolverlos	Referencia 10
Generalización	Realiza conexiones entre el problema actual y los problemas encontrados anteriormente	Referencia 17
Generar, revisar y evaluar soluciones	Transfiere conocimiento en la solución de otros problemas	Referencia 10
Hardware	Hay conocimiento de las partes que componen el robot y sus funciones.	Referencia 13
Implementación de una solución	Ha completado la solución	Referencia 12
Incremental e iterativo	Realiza mejoras paulatinas ampliando la solución	Referencia 11
Información	Puede guardar, actualizar y recuperar información	Referencia 11
Información	Procesa información de una manera lógica, representa la información de manera abstracta	Referencia 10
Información	Almacenamiento, recuperación y actualización de la información	Referencia 18
Interacciones del jugador	Facilita la interacción con el jugador	Referencia 16
Interactividad del usuario	Brinda herramientas al usuario para interactuar con el sistema, a través del teclado y ratón.	Referencia 16
Meta juego	La narrativa, las instrucciones y demás elementos del juego son acordes a lo esperado	Referencia 16
Operadores	Maneja datos numéricos, operaciones matemáticas y cadenas de caracteres	Referencia 18
Paralelismo	Hace uso de condicionales y operadores lógicos	Referencia 16
Paralelismo	Trabaja con otros para resolver diferentes partes del problema al mismo tiempo, realiza más de una acción al mismo tiempo	Referencia 11
Paralelismo	Usa secuencias ejecutándose al mismo tiempo	Referencia 18
Pensamiento algorítmico	Crea una lista de pasos para resolver los problemas, usa las matemáticas	Referencia 11
Pensamiento lógico	Utiliza bloques como esperar y parar, maneja mensajes correctamente	Referencia 16
Prácticas meta cognitivas	Planea con anterioridad lo que va a realizar identifica errores y explica como los resolvió. (planeación, evaluación y monitoreo)	Referencia 18
Proceso de diseño	Evidencia un proceso iterativo	Referencia 13
Pruebas sistemáticas	Prueba un atributo al tiempo	Referencia 12
Pruebas y depuración	Corre el programa para verificar que la solución sea correcta y soluciona los errores	Referencia 11
Razonamiento	Decide si los datos que se utilizarán para la solución del problema son adecuados o no	Referencia 17
Reconocimiento de patrones	Observa los elementos comunes y los reconoce	Referencia 12
Representación de la información	Identifica los símbolos de los bloques y cartas de acción.	Referencia 13
Representación de los datos	Modifica las propiedades de los sprites, usa operaciones con variables o listas	Referencia 16
Representación y solución de problemas	Resuelve problemas de una manera eficiente	Referencia 10
Reutilización, conexión y cuestionamiento	Comparte sus programas y revisa los de sus compañeros para mejorar el propio	Referencia 11
Secuencias	Representa las soluciones como una serie de pasos	Referencia 18
Software	Programa secuencias sintácticamente correctas.	Referencia 13
Tema de mitigación	Aborda correctamente el tema	Referencia 16

Tabla a 6. Caracterización de mecanismos de evaluación

Mecanismos de evaluación	Referencia	Grupo
Analizar automáticamente del código Scratch.	Referencia 4	Análisis automático
Analizar automáticamente el código de los estudiantes empleando Dr. Scratch	Referencia 16	Análisis automático
Analizar automáticamente el código escrito por los estudiantes por medio del prototipo online MindReader.	Referencia 9	Análisis automático
Analizar automáticamente las actividades realizadas a través de herramientas computacionales definidas en el marco de referencia propuesto.	Referencia 8	Análisis automático

Anexo A

Analizar automáticamente las habilidades de los estudiantes en el desarrollo de las tareas.	Referencia 15	Análisis automático
Analizar automáticamente los proyectos entregados por los estudiantes.	Referencia 7	Análisis automático
Analizar automáticamente y en tiempo real de los ejercicios de los estudiantes.	Referencia 5	Análisis automático
Analizar cualitativamente los elementos CT.	Referencia 16	Análisis manual
Analizar del cuadro entregado por codOrama, que relaciona la solución con los elementos definidos	Referencia 6	Análisis manual
Analizar los juegos terminados empleando una guía.	Referencia 18	Análisis manual
Analizar los proyectos, registrando los bloques de los proyectos y su frecuencia de uso.	Referencia 1	Análisis manual
Calificar de un instrumento de 23 ítems de respuesta múltiple y respuestas abiertas	Referencia 10	Análisis manual
Entrevistar a los estudiantes a través de preguntas basadas en artefactos-objetos.	Referencia 1	Entrevista
Entrevistar a los estudiantes para conocer sus prácticas y perspectivas.	Referencia 18	Entrevista
Estimar el grado de cumplimiento en el desarrollo de los retos propuestos en el juego.	Referencia 12	Análisis automático
Examinar el informe de diseño creativo.	Referencia 3	Revisión de registros anecdóticos
Indagar los resultados del desarrollo de actividades interactuando con KIBO.	Referencia 13	Entrevista
Inspeccionar el informe de reflexión.	Referencia 3	Revisión de registros anecdóticos
Observar a los participantes y tener conversaciones informales con ellos.	Referencia 18	Observación
Preguntar acerca del diseño de escenarios.	Referencia 1	Entrevista
Revisar el cumplimiento de cada criterio de acuerdo a tareas realizadas por sus estudiantes.	Referencia 2	Análisis manual
Revisar el desarrollo de tareas diseñadas de acuerdo al marco conceptual definido.	Referencia 3	Análisis manual
Revisar las bitácoras y hojas de resolución de problemas.	Referencia 18	Revisión de registros anecdóticos
Valorar el resultado de las tareas propuestas.	Referencia 14	Análisis manual
Valorar las respuestas de los estudiantes de la encuesta de autoeficacia.	Referencia 17	Encuesta
Valorar una encuesta de autoeficacia.	Referencia 11	Encuesta
Verificar las prácticas mediante una grabación de video de los estudiantes programando.	Referencia 15	Observación
Verificar un video grabación de los estudiantes interactuando con la herramienta.	Referencia 13	Observación
Verificar un video grabación de los estudiantes mientras ellos desarrollan las tareas diseñadas.	Referencia 12	Observación

Tabla a 7. Glosario de definiciones de los elementos de CT

Elemento	Definición
Automatización	Automatización implica la necesidad de tener un computador para que interprete las abstracciones. [129]. Tener computadores o maquinas realizando tareas repetitivas [23] y tediosas. [26]
Diseño de algoritmos (y procedimientos)	Un algoritmo es una abstracción de un procedimiento paso a paso para tomar una entrada y producir un resultado deseado. [129] Crear y ordenar una serie de instrucciones o pasos para resolver problemas o hacer tareas [23], [26], [116], [117] y otros similares a ese [104], [113] Hacer pasos y reglas[100]
Abstracción	Construir algo grande uniendo pequeñas partes que lo componen. [82] Reducir la complejidad hasta llegar a la idea principal [26]. Identificar y extraer la información relevante para definir la idea principal[23] Remover detalles innecesarios [100] Proceso de crear algo simple a partir de algo complicado al omitir los detalles irrelevantes, al encontrar los patrones relevantes y al separar las ideas de los detalles tangibles. La abstracción se utiliza en la definición de patrones, generalización de instancias y parametrización. [3] Habilidad de generalizar un patrón y priorizar que es pertinente y que no en relación a un problema. [105] Y generalización de patrones: es la habilidad de filtrar información no necesaria para resolver problemas, generalizar información y extrapolar principios y conceptos desde la información presente [116], [117] Eliminar los detalles específicos de un conjunto de problemas para encontrar la solución genérica que se pueda utilizar para todos ellos, o representando un sistema complicado con un modelo simple reflejando la capacidad de dividir un problema en partes más pequeñas que sean más fáciles de entender, programar

	<p>y depurar[109]</p> <p>Desarrollar modelos o simulaciones de un fenómeno natural o artificial usarlo para hacer predicciones acerca del mundo y analizar su eficacia y validez [102] Crear un modelo o simulación visual de un problema que incorpore solo los partes más importantes. [104]</p> <p>Sub categorías de abstracción: recolección y análisis de información, reconocimiento de patrones y modelado. [96]</p> <p>Mecanismos de abstracción: recursión, generalización; ocultamiento de la información y encapsulamiento [128]</p>
Recolección de información	Reunir información apropiada [26], [23].
Análisis de información	<p>Encontrar el sentido de la información, reconocer patrones y trazar conclusiones [26], [23]</p> <p>Organizar y analizar información [116]</p> <p>Y la visualización: información resumida en forma esquemática, que incluye atributos o variables para unidades de información o la capacidad de representar redes o sistemas complejos en representaciones visuales [117]</p>
Representación de información	Representar y organizar datos en gráficos, cuadros, palabras o imágenes apropiados [23], [26]
Descomposición de problemas	<p>Desglosar tareas en partes más pequeñas y manejables [23], [26], [100], [104], [113], [116] que se pueden resolver de forma independiente. [118]</p> <p>Las partes divididas deben elementos funcionales que en conjunto comprende todo el sistema o problema. [96]</p> <p>Pequeñas piezas que forman un gran problema. [105]</p> <p>Capacidad de descomponer un sistema en componentes o deconstruir una tarea en procedimientos para que uno pueda explicar los elementos de un proceso a otra persona o una computadora. [117]</p> <p>Explicación de una acción en detalle [112]</p>
Simulación (modelado)	<p>Representación o modelo de un proceso [26]</p> <p>Desarrollar modelos para imitar los procesos del mundo real [23]</p> <p>Construir modelos para representar como opera un sistema o como lo hará en el futuro [96]</p> <p>Hacer predicciones de resultados a partir del flujo de información de un problema planteado. [98] Resolver los problemas actuales a través de un modelo o desarrollar un nuevo sistema. [98]</p> <p>Implementar un algoritmo como bancos de pruebas para identificar qué circunstancias y abstracciones considerar. [114], [115]</p>
Paralelismo	<p>El paralelismo se refiere al procesamiento de tareas o secuencias de instrucciones que se realizan al mismo tiempo [74], [82], [109] con el fin de alcanzar un objetivo común [23].</p> <p>Organizar recursos para llevar a cabo tareas simultáneamente con el fin de alcanzar un objetivo común [26], [102].</p>
Reconocimiento de patrones	<p>Observar patrones, tendencias y regularidades en la información [23], [104], ayuda a los estudiantes a predecir con mayor cercanía que ocurrirá después [105] o que resultados se obtendrán [112].</p> <p>Capacidad de detectar puntos en común o disparidades que lo ayudarán a hacer predicciones o conducir a atajos [117].</p>
Generalización de patrones	<p>Crear reglas, principios o teorías de observación de patrones para probar resultados predichos [23]</p> <p>Reconocer y usar similitudes [100]</p> <p>Transferir el proceso de solución de un problema a una gran variedad de situaciones.[113] para resolver problemas efectivamente y eficientemente [96]</p> <p>Y abstracción: resolver problemas de tipos similares porque en pasadas experiencias se han resuelto. [112]</p>
Lógica – Razonamiento lógico	<p>Predecir y analizar [100].</p> <p>Se refiere a las consecuencias de un valor dada una declaración (falso o verdadero) [115].</p> <p>Capacidad del proyecto para comportarse de manera diferente dependiendo de la situación [109].</p> <p>Encontrar la asociación de patrones entre diferentes eventos [98]</p> <p>Habilidad para tomar decisiones en determinada circunstancia, soportando muchos resultados [82]</p>
Evaluar	Generar juicios [100]. Asegurar que la solución está bien [104].
Experimentar	Experimentar y jugar [100], [104]
Crear	Diseñar y hacer [100], [104]
Perseverar	Continuar haciendo el trabajo [100], [104]
Colaborar	Trabajar en conjunto [100] [104],[102]
Modularidad	Desarrollo de procesos autónomos, los cuales están encapsulados en un conjunto de comandos usados con frecuencia que desempeñan una función específica y podrían ser usado en el mismo o en diferente problemas [113].
Reflexionar	Reflexionar es la habilidad de hacer juicios que son justos y honestos en situaciones complejas [101].
Codificar	Es la traducción del diseño al código y su evaluación para garantizar que funcione correctamente en todas las condiciones previstas [101].

Anexo A

Diseñar	Implica crear representaciones del diseño, incluidas representaciones legibles por humanos, como diagramas de flujo, guiones gráficos, pseudocódigo, diagramas de sistemas, etc. [101].
Analizar	Incluye descomponer un partes (descomposición), reducir los detalles innecesarios (abstracción), identificar procesos (algoritmos) y encontrar similitudes en patrones (generalización) [101].
Aplicar	Es la adopción de soluciones preexistentes para cumplir con los requisitos de otro contexto [101]. Relacionado con generalización.
Secuencias	Serie de pasos individuales que son ejecutados por un computador [82].
Bucles	Repetición de las acciones [82].
Eventos	Una acción causa que otra ocurra [82].
Condicionales	Habilidad para tomar decisiones en determinada circunstancias, soportando muchos resultados [82]
Operadores	Proporciona soporte matemático, lógico y expresión de cadenas de caracteres, facilitando al programador manipular y operar la información [82]
Incremental e iterativo	Construir un programa por pequeñas partes en lugar de todo el programa a la vez [118], adaptándose a los posibles cambios que se puedan presentar [82]. Repetir los procesos de diseño para refinar las soluciones, hasta lograr el resultado ideal [96].
Probar y depurar	Detectar, identificar y solucionar errores [104], [100] cuando una solución no funciona como se supone que debería [96] , mientras se está desarrollando [118] Analizar problemas y errores en lógica o en actividades[114] Depuración se refiere al acto de determinar problemas en un algoritmo [115]
Reutilizar y remezclar	Soportar el desarrollo de las capacidades de lectura crítica de código y provocar importantes preguntas acerca de propiedad y autoría [82].
Expresar	La computación es algo que puede ser usado para diseñar y expresarse. [82]
Conectar	Interactuar con otras personas [82]
Cuestionar	Incluye interrogar lo que se da por sentado y en algunos casos responder al interrogante a través del diseño. [82] Entender cada parte del código en lugar de usar algún código sin entenderlo bien [118].
Conexión de la computación	Se refiere al estudio de los efectos que los desarrollos producen en los individuos y la sociedad, y al establecimiento de conexiones entre diferentes conceptos informáticos [102].
Crear artefactos computacionales	Son los aspectos creativos de la computación al diseñar y desarrollar artefactos computacionales interesantes, así como a la aplicación de técnicas computacionales para resolver problemas de manera creativa. Un artefacto computacional es una visualización, un gráfico, un programa, un video o un audio que los estudiantes crean usando una computadora. [102]
Analizar problemas y artefactos	Evaluar una solución propuesta a un problema. Localizar y corregir errores. Explicar cómo funciona un artefacto. [102]
Comunicar	Se compone de incluye descripciones escritas y orales respaldadas por gráficos, visualizaciones y análisis computacional. [102] Estar dispuesto y ser capaz de comunicarse efectivamente con los demás. [104]
Pensamiento lógico	Consiste en deducir la mayor cantidad de información posible de la poca que ya tiene, pero (y esto es lo más importante) no saltando a conclusiones. [103]
Pensamiento algorítmico	Es la capacidad de pensar en términos de tales algoritmos como una forma de resolver problemas. Es una habilidad básica que las personas desarrollan cuando aprenden a escribir sus propios programas de computadora [103] Pasos lógicos requeridos para construir una solución a un problema dado. Es más evidente cuando los participantes discuten reglas, procesos de pensamiento o prioridades relacionadas con las estrategias. [112] El pensamiento algorítmico es la práctica de crear un conjunto paso a paso de instrucciones, pautas o reglas para resolver un problema. [105] [118] Construir algoritmos: Implica la construcción de procedimientos paso a paso para resolver un problema en particular. [114] La construcción de algoritmos contiene un conjunto de lógica condicional y presentan instrucciones para resolver un problema complejo en un enfoque paso a paso. [115] Diseño de algoritmos: cree una serie de pasos ordenados para resolver un problema. [96]
Soluciones eficientes	Se refiere a los resultados de buena calidad y hechos rigurosamente [103].
Pensamiento científico	Es la capacidad de desarrollar nuevos conocimientos, ya sea que se trate de hechos cotidianos o de descubrimientos de vanguardia [103]
Pensamientos innovadores	Es la capacidad de desarrollar nuevas ideas y ser perseverante para alcanzarlas. [103]

Solución de problemas	Se refiere a resolver problemas con el pensamiento lógico mediante el uso de varios modelos computacionales [114] El paso final del pensamiento lógico [98]
Socializar	Se refiere al aspecto social de la CT, que implica coordinación, cooperación y / o competencia [114]
Computación distribuida	Se refiere a los aspectos sociales de CT e incluye muchas partes al desarrollar abstracciones [115]
Instancias no articuladas	Realizar una acción dentro del juego como resultado de un proceso de pensamiento que no pudo ser articulado [112]
Control de flujo	Está relacionado con el comportamiento del sprite controlado. Implica ciertos bloques que se repiten varias veces o hasta que surge la situación. [109], dirige los pasos de un algoritmo para completar. [118]
Interactividad con el usuario	Significa que, la persona que está ejecutando el proyecto pueda realizar acciones que provocan nuevas situaciones en el proyecto.[109]
Sincronización	Se refiere un concepto que permite a los sprites organizar las cosas para que sucedan en el orden que queremos. [109]
Hardware y Software	Componen los objetos inteligentes tanto su parte física como las instrucciones y programas que lo manejan, son obtenidos mediante la ingeniería humana. [97]
Proceso de diseño	Este proceso incluye la resolución de problemas, edición y revisión.[121]
Estar seguro de sí mismo	Crear en las propias habilidades para resolver problemas [104]
Ser flexible	Ser capaz de manejar los cambios y problemas. [104]
Transformación	Conversión de la información [98]
Lógica condicional	Encontrar la asociación de patrones entre diferentes eventos. [98]
Conexión con otras áreas	Encontrar la relación entre la información [98].
Visualización	Ver el contenido de una manera fácil de entender [98]
Eficiencia y restricciones de desempeño	Analizar la eficiencia de los resultados finales para alcanzar un objetivo más perfecto. [2], [98]

Anexo B

B. Tablas de apoyo de la definición del modelo– capítulo 4

Tabla b 1. Bloques de Scratch

Bloques Scratch					
Tipo	Nombre	Variables	Tipo	Nombre	Variables
	Movimiento			Sensores	
	mover	#	(pregunta)	¿tocando ?	puntero del ratón
	girar	#			borde
	apuntar en dirección	#			Sprite
	apuntar hacia	puntero ratón	(pregunta)	¿tocando el color?	color
		un sprite	(pregunta)	¿color tocando color?	color, color
	ir a	x, y	(valor-)	distancia a	puntero del ratón
	ir a	puntero ratón			Sprite
		posición aleatoria		preguntar pregunta y esperar	pregunta
		Objeto 1	(valor)	respuesta	
	deslizar en seg a	#, x, y	(pregunta)	tecla presionada	tecla
	cambiar x por	#	(pregunta)	¿ratón presionado?	
	fijar x a	#	(valor-)	posición x del ratón	
	cambiar y por	#	(valor-)	posición y del ratón	
	fijar y a	#	(valor)	volumen del sonido	
	rebotar si toca un borde		(valor)	movimiento del video en	este objeto
	fijar estilo de rotación	izquierda - derecha			escenario
		no rotar		video	apagado
		en todas direcciones			encendido
(valor)	posición en x	Valor para mostrar en pantalla o usar como variable			encendido e invertido
(valor)	posición en y			fijar transparencia de video a %	#
(valor)	dirección		(valor)	cronometro	
	Apariencia			reiniciar cronometro	
	decir por seg	palabra, #	(valor-)	posición de Sprite u escenario	posición en x
	decir	palabra			posición en y

	pensar por seg	palabra, #			dirección
	pensar	palabra			# de disfraz
	mostrar				nombre del disfraz
	esconder				tamaño
	cambiar disfraz a	opciones de disfraz			volumen
	siguiente disfraz		(valor)	opción actual	año
	cambiar fondo a	opciones de fondo			mes
	cambiar efecto por seg	color			fecha
		ojo de pez			día de la semana
		remolino			hora
		pixelar			minuto
		mosaico			segundo
		brillantez	(valor-)	días desde el 2000	
		desvaneces	(valor-)	nombre de usuario	
	establecer efecto a #	opción de efecto, #		Operadores	
	quitar efectos gráficos		(valor-)	a+b	
	cambiar tamaño por	#	(valor-)	a-b	
	fijar tamaño a %	#	(valor-)	a*b	
	enviar al frente		(valor-)	a/b	
	ir capas hacia a atrás	#	(valor-)	número al azar entre y	
(valor)	# de disfraz		(pregunta)	menor que	
(valor)	nombre de fondo		(pregunta)	igual	
(valor)	tamaño		(pregunta)	mayor que	
	Sonido		(pregunta)	y	
	tocar sonido	pop, grabar	(pregunta)	o	
	tocar sonido y esperar	pop, grabar #	(pregunta)	no	
	detener todos los sonidos		(valor-)	unir algo1 algo2	
	tocar tambor durante pulsos	#, #	(valor-)	letra # de algo	
	silencio por pulsos	#	(valor-)	longitud de algo	
	tocar nota durante pulsos	nota, #	(valor-)	a mod b	
	fijar instrumento a	21 opciones de instrumento (tambor, guitarra, etc.)	(valor-)	redondear	
	cambiar volumen por	#	(valor-)	operación de #	abs
	fijar volumen a %	#			piso
(valor)	volumen				piso
	cambiar tempo por	#			raíz cuadrada
	fijar tempo a ppm	#			sin
(valor)	tempo				cos
	Lápiz				tan
	borrar				asin
	sellar				acos
	bajar lápiz				atan
	subir lápiz				En
	fijar color de lápiz a	color			log
	cambiar color del lápiz por	#			potencia de e
	fijar color de lápiz a	#			potencia de 10
	cambiar intensidad del lápiz por	#		Más bloques	

Anexo B

	fijar intensidad del lápiz a	#		Crear un bloque	
	cambiar tamaño de lápiz por	#		Añadir una extensión	PicoBoard
	fijar tamaño de lápiz a	#			LEGO WeDo 1.0
	Datos				LEGO WeDo 2.0
crear una variable (valor)	fijar variable a	#			
	variable			Disfraces / Fondos	
	cambiar variable por			Agregar disfraz / fondo	
	mostrar variable	var		Agregar uno de Scratch	
	esconder variable	var		Agregar uno propio	
crear una lista (lista)	añade algo a lista	algo, lista		Agregar uno de un archivo	
	lista			Agregar uno de una fotografía que se tome al instante	
	borrar posición de lista	posición, lista		Editar disfraz/fondo	
	insertar algo en posición lista	algo, posición, lista		cambiar nombre	
	reemplazar elemento posición de lista con algo	posición, lista, algo		edición gráfica	
(valor-)	elemento posición de lista				
(valor-)	longitud de lista			Sonidos	
(pregunta)	¿la lista contiene algo?	lista, algo		Agregar sonido a escenario o sprite	
	mostrar lista	lista		Agregar sonidos	
	esconder lista	lista		Agregar un sonido grabado	
	Eventos			Agregar un sonido de una carpeta	
	al presionar bandera verde			Editar sonido	
	al presionar tecla	tecla		reproducir	
	al hacer clic en este objeto			parar	
	cuando el fondo cambie a	fondos		grabar	
	cuando ser mayor a	volumen del sonido o cronometro o movimiento del video, #		editar	deshacer
	al recibir mensaje	mensaje			rehacer
	enviar mensaje	mensaje			cortar
	enviar mensaje y esperar	mensaje			copiar
	Control				pegar
	esperar segundos	#			borrar
	repetir	#			seleccionar todo
	por siempre			Efectos	aparecer
	si entonces	condición			desvanecer
	si entonces, si no	condición			mas fuerte
	esperar hasta que	condición			más suave
	repetir hasta que	condición			silencio
	detener	todos			reversa

		este programa		Volumen del micrófono	
		otros programas en el objeto	Objetos i (Sprite)	nombre del objeto	editable
	al comenzar como clon			posición x y y	
	crear clon de	sprite		dirección	editable
	borrar este clon			estilo de rotación	circular
Objetos i	puede ser arrastrado	(opción)			horizontal
	mostrar	(opción)			punto
	Nuevo objeto				

Tabla b 2. Información de los estudiantes recolectada en la introducción

Variable	Tipo de dato	Rango	Tipo de información
Nombre y apellido	Cadena de caracteres	N.A.	Información personal
E-mail	Cadena de caracteres	N.A.	Información de identificación
Contraseña	Cadena de caracteres	N.A.	Información de identificación
Usuario Scratch	Cadena de caracteres	N.A.	Información de identificación
Edad	Númérico	8-18	Información personal
Genero	Binario	Femenino (0) o masculino (1)	Información personal
Grado	Númérico	3-11	Información personal
Materia favorita	Opción - numérico	0-8	Información personal
Actividad favorita del PC	Opción - numérico	0-8	Información personal
Conocimiento del PC	Númérico	0-10	Pregunta de autoeficacia
Manejo PC	Númérico	0-10	Pregunta de autoeficacia
Frecuencia Uso PC	Númérico	0-3	Pregunta de autoeficacia

Tabla b 3. Propuesta de las opciones de visualización de resultados para los docentes

Número de Criterios	Número de Estudiantes	Número de Ejercicios	Gráfica resultante	Información de la gráfica	Información adicional fuera de la grafica	Observaciones
Un criterio (seleccionado por el docente para mejor visualización)	Un estudiante	Todos los ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio	Solo valor por ejercicio en el tiempo	Valores: Media, moda, mediana, desviación estándar	
		Un grupo de ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio	Solo valor por ejercicio en el tiempo	Valores: Media, moda, mediana, desviación estándar	
	Un grupo de estudiantes (que el docente seleccione)	Un ejercicio	Estudiante Vs Valor obtenido en ese ejercicio	Solo valor por estudiante	Valores: Media, moda, mediana, desviación estándar	
		Un grupo de ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio, grafica por estudiante	Una gráfica por estudiante, ejercicio vs valor por criterio		Ejemplo tabla b5
		Todos los ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio, grafica por estudiante	Graficas superpuestas de los estudiantes		Ejemplo tabla b5
	Todos los estudiantes	Un ejercicio	Estudiante Vs Valor obtenido en ese ejercicio	Solo valor por estudiante	Valores: Media, moda, mediana, desviación estándar	
		Un grupo de ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio, grafica por estudiante	Una gráfica por estudiante, ejercicio vs valor por criterio		Ejemplo tabla b5

Anexo B

		Todos los ejercicios	Ejercicio Vs Valor obtenido en cada ejercicio por ese criterio, grafica por estudiante	Graficas superpuestas de los estudiantes		Ejemplo tabla b5
Un grupo de criterios - únicamente los criterios seleccionados por el docente al crear el desafío (ya sea por medio de la solución o manualmente)	Un estudiante	Un ejercicio	Criterio Vs Valor	Solo debería mostrar los valores fijos, ya que es un valor por criterio, 1 grafica		
		Un grupo de ejercicios	Criterio Vs Valores (media, moda, mediana de los ejercicios)	3 gráficas: Media, moda, mediana		
		Todos los ejercicios	Criterio Vs Valores (media, moda, mediana de los ejercicios)	3 gráficas: Media, moda, mediana		
	Un grupo de estudiantes (que el docente seleccione)	Un ejercicio	Criterio Vs Valor (media, moda, mediana de los estudiantes)	3 gráficas: Media, moda, mediana		
		Un grupo de ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
		Todos los ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
	Todos los estudiantes	Un ejercicio	Criterio Vs Valor (media, moda, mediana de los estudiantes)	3 gráficas: Media, moda, mediana		
		Un grupo de ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
Todos los criterios (puede seleccionar la opción el docente)		Todos los ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
	Un estudiante	Un ejercicio	Criterio Vs Valor	Solo debería mostrar los valores fijos, ya que es un valor por criterio, 1 grafica		
		Un grupo de ejercicios	Criterio Vs Valores (media, moda, mediana de los ejercicios)	3 gráficas: Media, moda, mediana		
		Todos los ejercicios	Criterio Vs Valores (media, moda, mediana de los ejercicios)	3 gráficas: Media, moda, mediana		
	Un grupo de estudiantes (que el docente seleccione)	Un ejercicio	Criterio Vs Valor (media, moda, mediana de los estudiantes)	3 gráficas: Media, moda, mediana		
		Un grupo de ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		
		Todos los ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
	Todos los estudiantes	Un ejercicio	Criterio Vs Valor (media, moda,	3 gráficas: Media, moda, mediana		

			mediana de los estudiantes)			
		Un grupo de ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		
		Todos los ejercicios	Criterio vs Valor promedio de los ejercicios	Una gráfica por estudiante, criterio vs valor promedio de los ejercicios		Ejemplo tabla b4
Clasificación	Un estudiante	Un ejercicio			Clasificación	
		Un grupo de ejercicios			Clasificación	
		Todos los ejercicios			Clasificación	
	Un grupo de estudiantes (que el docente seleccione)	Un ejercicio			Clasificación	
		Un grupo de ejercicios			Clasificación	
		Todos los ejercicios			Clasificación	
	Todos los estudiantes	Un ejercicio			Clasificación	
		Un grupo de ejercicios			Clasificación	
		Todos los ejercicios			Clasificación	

Tabla b 4. Ejemplo 2. Tabla de presentación de calificación

	Estudiante 1	Estudiante 2	...	Media	Mediana	Moda
Criterio 1	Valor promedio en los ejercicios					
Criterio 2						
Criterio 3						
...						

Tabla b 5. Ejemplo 1. Tabla de presentación de calificación

	Estudiante 1	Estudiante 2	...	Media	Mediana	Moda
Ejercicio 1	Valor					
Ejercicio 2						
Ejercicio 3						
...						
Media						
Mediana						
Moda						

Tabla b 6. Plan del proyecto de minería de datos

Fase	Actividad	Requerimientos	Resultados
Comprensión de la información	Recopilación inicial de datos	Tabla b2 Anexo B, Tabla 40, Tabla 41.	Listado de variables, primera columna tablas b7 y b8 del Anexo B.
	Descripción de los datos	Resultados actividad anterior	Tablas b7 y b8 del Anexo B.
	Exploración de los datos	Resultados actividad anterior	Tabla 42.
	Verificación de calidad de datos	Resultados actividad anterior	Tabla 43.
	Selección de los datos	Criterios de inclusión y exclusión	Tabla 44

Anexo B

Preparación de la información	Limpieza de los datos	Información de la actividad verificación de la calidad de datos	Data set (se hicieron varios, únicamente se adjunta el definitivo en la tabla X)
	Construcción de la información	Datos seleccionados y limpios	
	Integración de los datos	Información de todas las tablas	
	Formateo de los datos.	Datos integrados	
Modelado	Selección de la técnica de modelado		Modelo sección 4.4.6
	Diseño de la evaluación		
	Construcción del modelo		
	Evaluación del modelo		
Evaluación	Evaluación de resultados		Resultado de la evaluación sección 4.4.7
	Revisar el proceso		
	Establecimiento de los siguientes pasos o acciones		
Despliegue	Planificación de despliegue		
	Planificación de la monitorización y del mantenimiento		
	Generación de informe final		
	Revisión del proyecto		

Tabla b 7. Descripción de los datos entregados por el estudiante– Fase Comprensión de la información

Variable	Tipo de dato	Rango	Opciones	Tipo de información	Descripción
Nombre y Apellido	Cadena de caracteres	N.A.	N.A.	Información personal	Nombre completo del estudiante
E-mail	Cadena de caracteres	N.A.	N.A.	Información de identificación	Correo electrónico
Contraseña	Cadena de caracteres	N.A.	N.A.	Información de identificación	Contraseña de ingreso
Usuario Scratch	Cadena de caracteres	N.A.	N.A.	Información de identificación	Usuario en la plataforma Scratch del MIT
Edad	Númérico	8-18	Valor numérico del 8-18	Información personal	Edad del estudiante
Genero	Binario	0-1	Femenino (0) o masculino (1)	Información personal	Genero del estudiante
Grado	Númérico	3-11	Valor numérico del 3-11	Información personal	Grado del estudiante
Materia favorita	Númérico	0-8	Matemáticas (0), Historia (1), Geografía (2), Español (3), Inglés (4), Artes (5), Informática (6), Educación física (7), Música (8)	Información personal	Materia favorita del estudiante
Actividad favorita del PC	Númérico	0-8	Redes sociales (0), ver videos (1), ver películas/series (2), buscar información (3), aprender más (4), hacer tareas (5), jugar video juegos (6), comunicarme con amigos o familiares (7), otras actividades (8)	Información personal	Actividad favorita del computador del estudiante
Conocimiento del PC	Númérico	0-10	Valor numérico del 0-10	Pregunta de autoeficacia	Grado de conocimiento que el estudiante considera tiene en cuanto al computador
Manejo PC	Númérico	0-10	Valor numérico del 0-10	Pregunta de autoeficacia	Grado de habilidad que el estudiante considera tiene en cuanto al manejo de computadores

Frecuencia Uso PC	Numérico	0-3	Todos los días (0), una vez por semana (1), rara vez (2), nunca (3)	Pregunta de autoeficacia	Frecuencia con la que el estudiante interactúa con computadores
Elabora secuencias	Binario	0-1	No elabora secuencias (0), elabora secuencias (1)	Evidencia entregada por el estudiante	Uso de bloques Scratch formando secuencias
No hay código muerto	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Número de bloques Scratch sin usar
No hay código duplicado	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Número de secuencias iguales
Define bloques	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Emplea clones	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
# hilos por sprite (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
# sprite tiene eventos (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
# hilos iniciando en bandera verde (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
# programas en iniciando en clic mouse a sprite o clic any key (2 o más),	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Envío y recepción de mensajes	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
# programas en los otros eventos (2 o más)	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Envío y recepción de mensajes a varios sprites	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Sensores (entrada ratón y teclado)	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Uso de variables Scratch	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Sensores de sprite	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Creación de variables	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Variables compartidas con todos los sprites	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Creación de listas	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Operadores matemáticos básicos, unir palabras	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Operadores matemáticos poco comunes, número al azar, mayor, menor, igual	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable

Anexo B

Operadores lógicos, operadores anidados	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Esperar, repetir, repetir por siempre, si, detener este	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Esperar hasta, repetir hasta, si, si no, detener todos los programas del sprite	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Condicionales anidados	Numérico	0-n	Número de ocurrencias de la variable	Evidencia entregada por el estudiante	Uso de bloques Scratch correspondientes con la variable
Evaluación1	Numérico	0-4	No (0), Casi nada (1), Parcialmente (2), Casi todo (3), Completamente (4)	Respuesta en encuesta	Evaluación de la solución
Depuración	Numérico	0-3	No probó el proyecto (0), Encontró errores y no hizo nada (1), Encontró errores y preguntó (2), Encontró errores y revisó (3)	Respuesta en encuesta	Pregunta acerca de prácticas de depuración
Generalización	Numérico	0-3	No reutilizo código alguno (0), Reutilizó código encontrado en Google o de un compañero (1), Reutilizó código encontrado en el repositorio de Scratch(2), Reutilizó código de un proyecto propio (3)	Respuesta en encuesta	Pregunta acerca de prácticas de generalización
Evaluación2	Numérico	0-4	Nulo(0), Malo (1), Regular (2), Bueno (3), Excelente (4)	Respuesta en encuesta	Autoevaluación
Evaluación3	Numérico	0-4	Nulo(0), Malo (1), Regular (2), Bueno (3), Excelente (4)	Respuesta en encuesta	Evaluación de pares

Tabla b 8. Variables seleccionadas por el docente– Fase Comprensión de la información

Variable	Tipo de dato	Rango	Opciones	Tipo de información	Descripción
Elabora secuencias	Binario	0-1	No elabora secuencias (0), elabora secuencias (1)	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch formando secuencias
No hay código muerto	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante deja el número de bloques Scratch sin usar
No hay código duplicado	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante repite el número de secuencias iguales
Define bloques	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Emplea clones	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
# hilos por sprite (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
# sprite tiene eventos (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
# hilos iniciando en bandera verde (dos o más)	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
# programas en iniciando en clic mouse a sprite o	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable

clic any key (2 o más),					
Envío y recepción de mensajes	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
# programas en los otros eventos (2 o más)	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Envío y recepción de mensajes a varios sprites	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Sensores (entrada ratón y teclado)	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Uso de variables Scratch	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Sensores de sprite	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Creación de variables	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Variables compartidas con todos los sprites	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Creación de listas	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Operadores matemáticos básicos, unir palabras	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Operadores matemáticos poco comunes, número al azar, mayor, menor, igual	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Operadores lógicos, operadores anidados	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Esperar, repetir, repetir por siempre, si, detener este	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Esperar hasta, repetir hasta, si, si no, detener todos los programas del sprite	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Condicionales anidados	Numérico	0-n	Número de ocurrencias de la variable	Criterio de evaluación seleccionado por el docente	El estudiante usa los bloques Scratch correspondientes con la variable
Valoración docente	Numérico	0-5	Nulo(0), Nulo(1), Deficiente(2), Aceptable(3), Bien(4), Excelente(5)	Calificación del docente	El estudiante cumple con los objetivos propuestos.

Tabla b 9. Dataset final

Anexo digital

Tabla b 10. Resultado - Redes Bayesianas

Anexo digital

Anexo B

Tabla b 11. Resultado – Redes neuronales

Anexo digital

Tabla b 12.Resultado Tasa nula

Anexo digital

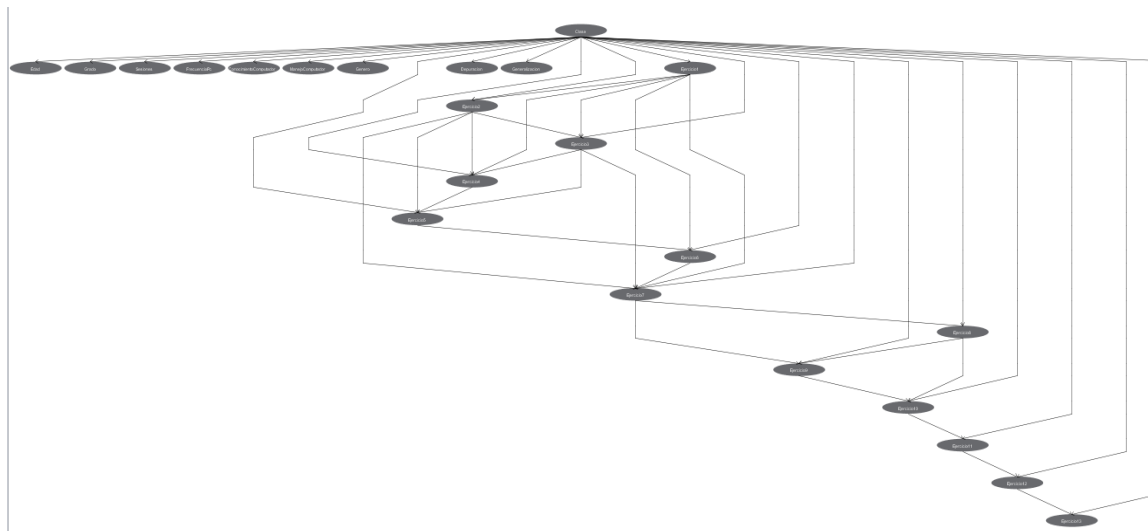


Figura b 4. Grafo generado - Red Bayesiana

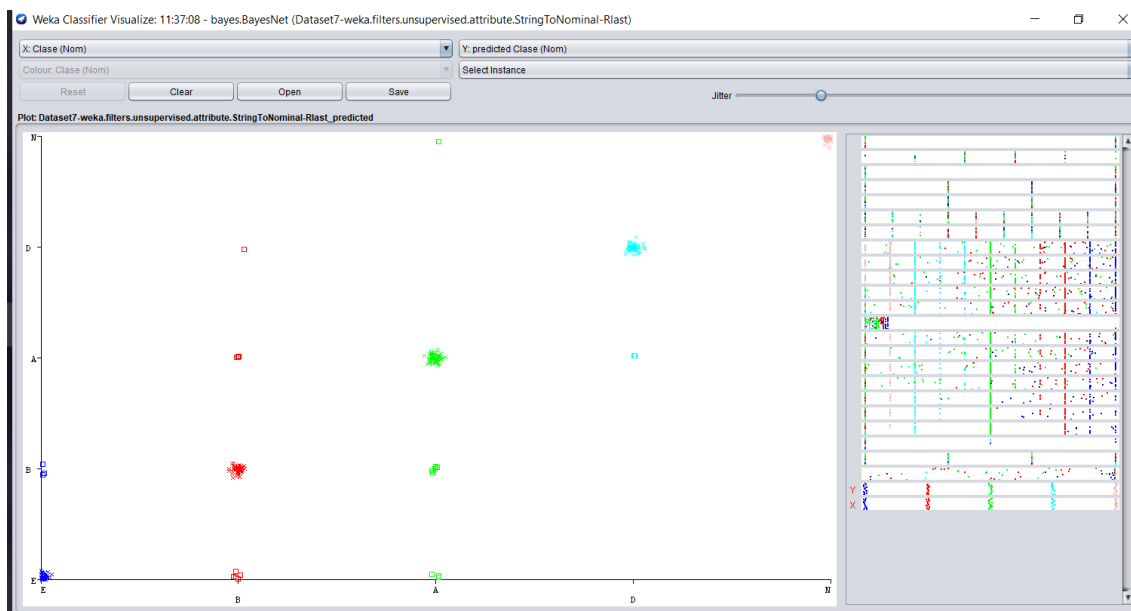


Figura b 5. Errores de clasificación - Red Bayesiana



Figura b 6. Errores de clasificación - Red Neuronal

Anexo C

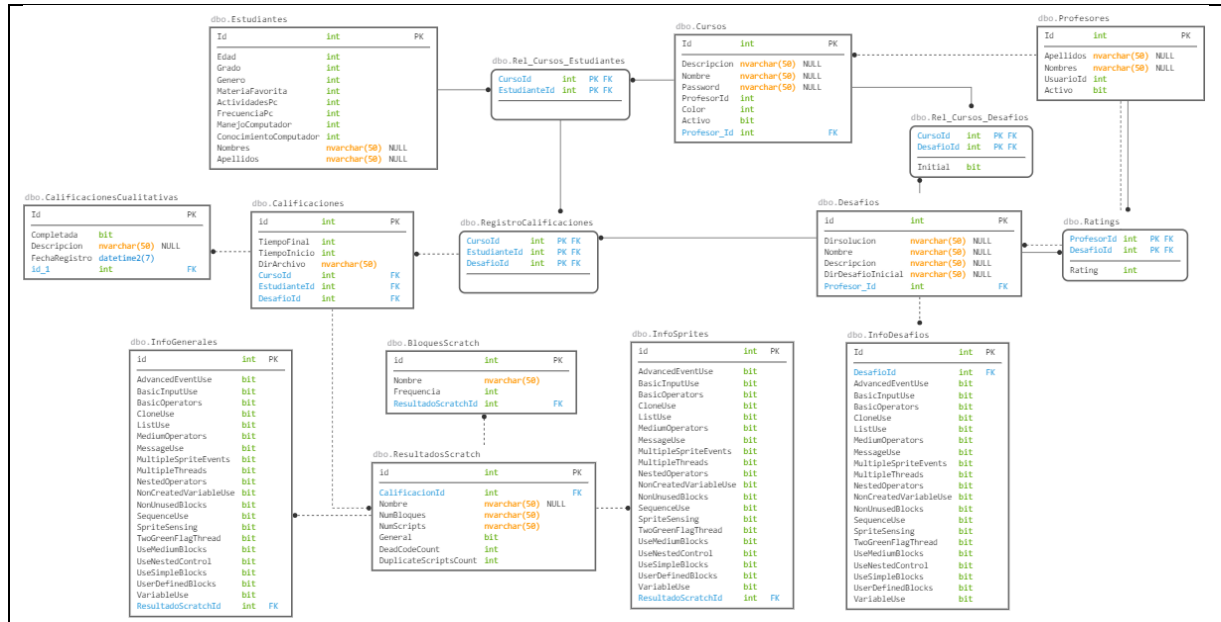
C. Tablas de apoyo en la definición de la herramienta HERA-CT capítulo 5

Tabla c 1. Relación de los nombres de las variables definición de criterios y base de datos

Elementos de CT		Nombre en base de datos	Variable	Nombre en base de datos	Variable	Nombre en base de datos	Variable
Diseño de algoritmos		SecuenceUse	Elabora secuencias	DeadCodeCount	No hay código muerto	DuplicateScriptsCount	No hay código duplicado
Abstracción		NonUnusedBlocks	No hay código muerto				
	Reconocimiento de patrones	UserDefinedBlocks	Define uno o más bloques	CloneUse	Emplea clones		
	Descomposición de problemas	MultipleThreads	# hilos por sprite (dos o más)	MultipleSpriteEvents	# sprite tiene eventos (dos o más)		
Paralelismo			# hilos iniciando en bandera verde (dos o más)	EventsUse	# programas en iniciando en clic mouse a sprite o clic any key (2 o más).	AdvancedEventUse	# programas en los otros eventos (2 o más)
		TwoGreenFlagThread		MessageUse	Envío y recepción de mensajes	MessageUse	Envío, espera y recepción de mensajes
Información	Recolección	SpriteSensing	Sensores (entrada ratón y teclado)	BasicInputUse	Uso de variables Scratch (no creadas, Ej.: la respuesta a una pregunta, posición x, etc.)		Sensores de sprite
	Representación	VariableCreation	Creación de variables	SharedVariables	Variables compartidas con todos los sprites	ListUse	Creación de listas
	Análisis	BasicOperators	Operadores matemáticos	MediumOperators	Operadores matemáticos poco	AdvancedOperators	Operadores lógicos,

			cos básicos, unir palabras		comunes, número al azar, mayor, menor, igual		operadores anidados
Lógica condicional		UseSimpleBlocks	Esperar, repetir, repetir por siempre, si, detener este	UseMediumBlocks	Esperar hasta, repetir hasta, si, si no, detener todos los programas del sprite	UseNestedControl	Condiciones anidados

Tabla c 2. Modelo lógico de la base de datos



Anexo D

D. Información de soporte del estudio de caso- capítulo 6

Tabla d 1. Información recolectada por HERA-CT en el registro de los estudiantes al sistema

Variable	Tipo de dato	Rango	Opciones	Tipo de información	Descripción
Nombre y Apellido	Cadena de caracteres	N.A.	N.A.	Información personal	Nombre completo del estudiante
E-mail	Cadena de caracteres	N.A.	N.A.	Información de identificación	Correo electrónico
Contraseña	Cadena de caracteres	N.A.	N.A.	Información de identificación	Contraseña de ingreso
Usuario Scratch	Cadena de caracteres	N.A.	N.A.	Información de identificación	Usuario en la plataforma Scratch
Edad	Numérico	8-18	Valor numérico del 8-18	Información personal	Edad del estudiante
Genero	Binario	0-1	Femenino (0) o masculino (1)	Información personal	Genero del estudiante
Grado	Numérico	3-11	Valor numérico del 3-11	Información personal	Grado del estudiante
Materia favorita	Numérico	0-8	Matemáticas (0), Historia (1), Geografía (2), Español (3), Inglés (4), Artes (5), Informática (6), Educación física (7), Música (8)	Información personal	Materia favorita del estudiante
Actividad favorita del PC	Numérico	0-8	Redes sociales (0), ver videos (1), ver películas/series (2), buscar información (3), aprender más (4), hacer tareas (5), jugar video juegos (6), comunicarme con amigos o familiares (7), otras actividades (8)	Información personal	Actividad favorita del computador del estudiante
Conocimiento del PC	Numérico	0-10	Valor numérico del 0-10	Pregunta de autoeficacia	Grado de conocimiento que el estudiante considera tiene en cuanto al computador
Manejo PC	Numérico	0-10	Valor numérico del 0-10	Pregunta de autoeficacia	Grado de habilidad que el estudiante considera tiene en cuanto al manejo de computadores
Frecuencia Uso PC	Numérico	0-3	Todos los días (0), una vez por semana (1), rara vez (2), nunca (3)	Pregunta de autoeficacia	Frecuencia con la que el estudiante interactúa con computadores

Tabla d 2. Problemas o desafíos propuestos en el estudio de caso

Desafío No.	Nombre	Área del conocimiento	Descripción
Desafío 1	Inicial		Ejercicio diagnóstico
Desafío 2	Orbitando	Ciencias naturales	¿Cómo le explicarías a un niño como giran los planetas alrededor del Sol? Desafío: Cuando se dé inicio al programa, el planeta debe iniciar una órbita alrededor del Sol.
Desafío 3	Fotosíntesis	Ciencias naturales	Durante la fotosíntesis las plantas absorben luz solar, agua (H ₂ O) y dióxido de carbono (CO ₂), lo que transforman en oxígeno (O ₂) y materia orgánica (C ₆ H ₁₂ O ₆). Desafío: Al iniciar el programa, todos elementos deben tomar una posición de acuerdo al proceso de fotosíntesis.
Desafío 4	Hierro	Ciencias naturales	Cuando existe falta de hierro (Fe) en las plantas, también conocido como clorosis férrica, las hojas se vuelven pálidas y su crecimiento es reducido. Desafío: Muestra el efecto que produce la falta de hierro en una planta, empleando una comparación en el escenario propuesto.
Desafío 5	Conversación ambiental	Ciencias naturales	El tiempo de descomposición de una botella plástica es superior a los 300 años, sin embargo muchas personas usan de estas botellas diariamente y las desechan después de un solo uso. Desafío: Mediante una conversación entre personajes, explica cómo se puede disminuir el consumo de productos en botellas plásticas.
Desafío 6	PH	Ciencias naturales	El pH es una medida de acidez o alcalinidad de una disolución. El agua tiene un pH neutro equivalente a 7, dentro de nuestro cuerpo los alimentos como las frutas se convierten en alcalinos, por lo tanto adquieren un pH superior a 7, mientras que la comida chatarra acidifica el cuerpo. Se tiene la banda para medir el pH, esta cambia a color verde cuando el pH es 7, color rojo cuando es menor y color violeta cuando es mayor a 7 el pH. Desafío: Mostrar cómo al realizar una simulación que al acercar los objetos a la banda esta cambia de color de acuerdo a su PH.
Desafío 7	Cadena trófica	Ciencias naturales	La cadena trófica es la corriente de nutrientes que se establece entre las especies que comparten un ecosistema. En el mar los tiburones se consideran consumidores heterótrofos ya que requieren materia orgánica procedente de otros seres vivos para producir sus componentes, de la misma forma algunos peces grandes consumen a los más pequeños. Desafío: Muestre como se desarrolla la cadena trófica en el fondo del mar.
Desafío 8	Musgos	Ciencias naturales	Los musgos son plantas que no tienen sistema vascular ni raíces como los arbustos y árboles, y no pueden almacenar agua ni absorberla del subsuelo. Por lo tanto se encuentran adaptadas a ambientes húmedos y tropicales porque pueden disponer de abundante agua durante todo el año. Desafío: permitir mover el musgo de un entorno a otro y mostrar como el tamaño de este variara de acuerdo a las características del ambiente.
Desafío 9	Ebullición del agua	Ciencias naturales	El punto de ebullición del agua es de 100 grados centígrados. Desafío: muestre como varía la temperatura cuando se enciende el fuego, iniciando en los 32 grados centígrados, que es la temperatura ambiente, hasta lograr su punto de ebullición, en este momento deben aparecer las burbujas sobre el agua.
Desafío 10	Figuras geométricas	Matemáticas	Desafío: realice un programa que grafique cada una de las figuras geométricas propuestas.
Desafío 11	Laberinto	Matemáticas	Los cisnes buscan una pareja para toda la vida, su búsqueda comienza a los 2-3 años de vida y cuando la encuentran suelen permanecer junta a ella hasta que muere, aunque en algunas ocasiones hasta que se pierde. Desafío: proporcione el control del cisne para que este pueda superar el laberinto y llegar hasta su pareja perdida.
Desafío 12	Puntaje	Matemáticas	Desafío: mueve el balón en todos los sentidos y cuente las veces que toca la varilla.
Desafío 13	Amenaza en el hormiguero	Ciencias naturales	Los animales mirmecófagos son los que están especializados en comer hormigas y termitas. Desafío: incluya en el escenario un mirmecófago que se coma todos los clones de hormigas que puedan salir.

Anexo D

Tabla d 3. Relación de los desafíos propuestos en el estudio de caso y los elementos de CT

Desafío No.	Nombre	Diseño de algoritmos	Abstracción	Paralelismo	Información	Lógica condicional
Desafío 1	Inicial	DA 1, DA 2, DA 3	A1			
Desafío 2	Orbitando	DA 1, DA 2, DA 3	A1		R1	2LC1
Desafío 3	Fotosíntesis	DA 1, DA 2, DA 3	A1	P1	R1, R2	
Desafío 4	Hierro	DA 1, DA 2, DA 3	A1, D2	P1		2LC1
Desafío 5	Conversación ambiental	DA 1, DA 2, DA 3	A1, D1, D2	P2		LC1
Desafío 6	PH	DA 1, DA 2, DA 3	A1		R3	LC1, LC2, LC3
Desafío 7	Cadena trófica	DA 1, DA 2, DA 3	A1, D1	P1	R1	LC2
Desafío 8	Musgos	DA 1, DA 2, DA 3	A1	P3	R2, R3	2LC1
Desafío 9	Ebullición del agua	DA 1, DA 2, DA 3	A1, D1	P2	Rp1, A1	2LC2
Desafío 10	Figuras geométricas	DA 1, DA 2, DA 3	A1, R1		A2	LC1
Desafío 11	Laberinto	DA 1, DA 2, DA 3	A1, D1	P2	R3	LC1, LC2, LC3
Desafío 12	Puntaje	DA 1, DA 2, DA 3	A1, D2	P3	R3, Rp1, Rp2, A1	LC1
Desafío 13	Amenaza en el hormiguero	DA 1, DA 2, DA 3	A1, R2, D2	P3	R3	LC2

Tabla d 4. Consentimiento informado

Consentimiento Informado

Proyecto: Modelo de evaluación formativa para estimar el desarrollo del Pensamiento Computacional. Lugar: Popayán.

Estimados Padres y Madres de Familia:

El Parque Informático Carlos Albán y la Universidad del Cauca apoyan la práctica de obtener el consentimiento informado de, y proteger a, los sujetos humanos que participen en investigaciones. La siguiente información tiene por objeto ayudarlo a decidir si permitirá que su hijo (a) participe en el presente estudio. Su (s) hijo (s) está (n) cordialmente invitados a formar parte de este proyecto de investigación por asistir al aula SmartSchool en el Parque Informático Carlos Albán.

El objetivo es proponer un modelo para la evaluación del aprendizaje de las habilidades mínimas del pensamiento computacional, en estudiantes de educación básica y media. Para esto, los estudiantes asistirán a sesiones donde se desarrollaran soluciones a problemas mediante la programación, estas se evaluarán por la herramienta que incluye la propuesta para determinar con la evidencia recolectada si se han adquirido o no las habilidades mínimas de la competencia.

Nos gustaría que su hijo (a) participara, pero tal participación es estrictamente voluntaria. Le aseguramos que el nombre de su hijo (a) no se mencionara en los hallazgos de la investigación. La información solo se identificará mediante un código numérico.

Usted no recibirá incentivo de índole económico (no se le dará dinero) por participar en el estudio, pero los resultados del desempeño de su hijo durante la actividad le serán compartidos si son de su interés. Si le gustaría tener información adicional acerca del estudio antes o después que finalice, puede ponerse en contacto con cualquiera de nosotros por teléfono o correo. Agradecemos su atención y apreciamos su interés y cooperación.

La información sobre su hijo(a) será vista únicamente por nuestro grupo de investigación. La información será guardada en cajón bajo llave en la oficina de la profesora Carolina González en la facultad de Ingeniería

Electrónica de la Universidad del Cauca. Nadie fuera de nuestro equipo de investigación verá la información sobre su hijo (a). Solo el equipo de investigación sabrá cuál es el código único de su hijo (a). De este modo, podremos comunicarle si encontramos algo que requiera su atención con respecto al proceso de aprendizaje que adelanta el niño (a). Cuando expliquemos nuestra investigación a otras personas, no usaremos su nombre o nada que permita que otra gente sepa quién es usted o su hijo(a). La información será guardada cumpliendo los criterios de confidencialidad y respeto. Cabe resaltar que todos los datos personales utilizados en este proyecto solo se manejarán en el mismo y no serán utilizados en otras investigaciones.

Se construirá una base de datos, ésta estará en un computador del grupo de investigación, en custodia del investigador principal, Carolina González Serrano y al a cual solo tendrán acceso los aliados e investigadores del proyecto. Los resultados serán divulgados a través de medios de comunicación y eventos académicos, sin embargo, no se mencionaran los nombres de los niños y niñas, de modo que otras personas puedan aprender de nuestra investigación, sin conocer su identidad. La información obtenida será de utilidad para la decisión acerca de la manera de mejorar los procesos de enseñanza.

Usted tendrá derecho a retirar a su hijo (a) cuando lo desee, su participación es voluntaria, y en caso de retirarse los datos de su hijo (a) serán eliminados. Esta investigación contiene los elementos éticos que la ley y la doctrina exigen (código Helsinki – código Nuremberg resolución 08430 de 1993) que rigen la ética en la investigación científica en Colombia. Se garantiza total confidencialidad con los datos recolectados.

Si tiene preguntas, puede contactar a la profesora Carolina González Serrano, oficina 424 en la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Teléfono 8209800 Ext. 2117 o al correo cgonzals@unicauca.edu.co.

Esta propuesta ha sido evaluada y aprobada por el comité de ética de la Vicerrectoría de Investigaciones de la Universidad del Cauca, Popayán.

Cláusulas estándar:

- Mi decisión de participar en el estudio es completamente voluntaria.
- Los procedimientos principales, han sido expuestos en un lenguaje comprensible, logrando total entendimiento del mismo.
- La información obtenida en este estudio, será mantenida con estricta confidencialidad por el grupo de investigadores, y solo para el propósito del protocolo descrito en este documento.
- Me han explicado los beneficios de este estudio.
- Me han explicado que me podré retirar en cualquier momento del estudio, sin que ello acarree perjuicio en este estudio.
- Los resultados de este estudio pueden ser divulgados en eventos nacionales y/o internacionales o ser publicados en revistas científicas son identificar el nombre de mi hijo(a). Además tendré derecho a conocer los resultados. Si tengo una pregunta durante o después del procedimiento puedo contactar a los investigadores.

Certificado de consentimiento

Nombre del participante: _____

Nombre legible del padre, madre o tutor: _____

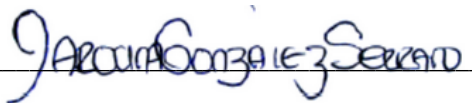
Firma del tutor: _____ Fecha: _____

He leído exactamente o he sido testigo de la lectura correcta del consentimiento al participante potencial, y éste ha tenido la posibilidad de hacer preguntas. Confirmando que el participante ha dado consentimiento libremente.

Anexo D

Nombre legible del investigador: Carolina González Serrano.

Firma del Investigador: _____



Se da una copia de éste consentimiento informado al participante _CGS (Iniciales del investigador)

Tabla d 5. Datos recolectados durante el registro al prototipo y empleado en el estudio de caso

Id	Actividad Pc	Conocimiento PC	Edad	Frecuencia Pc	Genero	Grado	Manejo PC	Materia Fav
1	7	10	14	0	1	8	7	4
2	0	5	14	0	0	8	5	3
3	6	8	14	0	1	8	5	0
4	0	8	15	2	1	8	5	6
5	2	8	14	2	0	8	5	7
6	6	4	13	3	1	8	10	8
7	1	8	12	2	0	8	4	7
8	6	9	14	1	1	8	6	0
9	0	6	14	0	1	8	4	1
10	8	2	14	1	1	8	5	0
11	0	6	14	0	0	8	4	7
12	8	6	13	3	1	8	6	7
13	6	3	14	0	1	8	5	7
14	0	2	13	2	1	8	5	0
15	8	6	15	2	1	8	3	7
16	0	10	14	0	1	8	8	3
17	4	7	13	0	1	8	6	3
18	8	2	14	1	1	8	7	5
19	0	2	14	3	1	8	2	7
20	5	6	15	2	1	8	8	6
21	3	9	15	2	1	9	6	3
22	0	9	15	0	0	9	7	3
23	6	10	13	0	1	9	9	0
24	3	10	14	0	1	9	6	6
25	0	4	13	1	0	9	5	4
26	0	4	15	2	1	9	6	7
27	0	7	15	2	1	9	5	7
28	0	5	15	0	1	9	4	7
29	0	10	13	0	0	9	10	4
30	1	8	15	0	1	9	8	1
31	6	7	14	0	1	9	7	0
32	6	10	15	0	1	9	8	3
33	6	10	15	0	0	9	9	6
34	2	10	15	0	1	9	8	4
35	3	9	16	0	0	9	8	3
36	3	7	15	0	0	9	7	3
37	2	5	17	1	0	9	7	4
38	6	10	14	0	1	9	10	0
39	3	4	14	0	1	9	8	1
40	1	5	14	2	0	9	4	4
media	3,15	6,775	14,2	0,85	0,7	8,5	6,3	3,95

moda	0	10	14	0	8	5	7
sumatoria					28		

Tabla d 6. Discretización de los elementos de CT relacionados con las preguntas en la entrevista

Estudiante	1 (información)		2 Solución de problemas				3 Evaluación		4 Generalización de patrones		5 Depuración		6 Depuración		7 Evaluación		8 Evaluación											
	Preguntaste a otra persona para recibir ayuda en el proyecto?		Describe el proceso de desarrollo				Si empleaste Estrategia		El producto terminado, cumple con los requerimientos planteados?		Empleaste programación de otro proyecto?		Probaste el proyecto antes de enviarlo?		Si lo probaste, encontraste errores?		Como fue tu compromiso en el desarrollo del proyecto?		Como fue el compromiso de tu compañero?									
	Si	No	A quien?	Ensayo y error	Plan estrategia	Otro	Cual?	Si funciona	Si, con cambios	No funciona	Si	No	Si	No	Cual?	Si	No	Si	No	Que hiciste?	Bueno	Normal	Malo	Bueno	Normal	Malo	Igual al mio	
1	1	0	al profesor	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	analice de nuevo	0	1	0	0	0	1	0		
2	1	0	al profesor	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	preguntaba	0	1	0	0	1	0	0		
3	1	0	un compañero	0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0	mover objetos	1	0	0	1	0	0	0	
4	1	0	un compañero	0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0	analice de nuevo	1	0	0	1	0	0	0	
5	0	1		0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	lo probe y funciona	1	0	0	0	0	0	0		
6	1	0	un compañero	1	1	0	ambos	0	1	0	1	0	0	0	0	0	1	1	0	analice de nuevo	1	0	0	1	0	0	0	
7	1	0	un compañero	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1		0	1	0	0	1	0	0	
8	1	0	al profesor	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0		0	1	0	0	0	0	0	1	
9	1	0	un compañero	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	mover objetos	1	0	0	1	0	0	0	
10	1	0	a la ingeniera y al profesor	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	analice de nuevo	1	0	0	1	0	0	0	
11	1	0	un compañero	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	scrata	1	0	0	1	0	0	0	
12	1	0	un compañero	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	analice de nuevo	1	0	0	1	0	0	0	
13	1	0	a la ingeniera	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	mover objetos	1	0	0	1	0	0	0	
14	1	0	un compañero	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1		0	0	1	0	0	0	1	
15	1	0	un compañero	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0		0	1	0	0	1	0	0	
16	1	0	un compañero	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	mover objetos	1	0	0	0	1	0	0	
17	1	0	a la ingeniera y al profesor	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	mi co	1	0	1	0		0	0	
18	1	0	a la ingeniera	1	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	corregirlos	0	0	0	0	0	1	0
19	1	0	un compañero	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	animaciones	1	0	0	1	0	0	0	
20	1	0	al profesor	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1		0	0	1	0	1	0	0	
21	1	0	un compañero	0	1	0	0	0	1	0	1	0	0	1	0	1	0	0	0		0	1	0	0	1	0	0	
22	1	0	a la profesora y un compañero	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	corregirlos	1	0	0	1	0	0	0
23	1	0	un compañero	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1			0	1	0	0	1	0	0	
24	0	1		0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	
25	1	0	un compañero	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	corregirlos	1	0	0	1	0	0	0
26	1	0	a la ingeniera y a un compañero	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	corregirlos	1	0	0	1	0	0	0
27	1	0	a la ingeniera	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	scrac	0	0	1	0		0	0	
28	1	0	a la ingeniera	0	1	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	corregirlos	1	0	0	0	0	0	1
29	1	0	un compañero	0	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0		0	1	0	0	1	0	0	
30	1	0	al profesor	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	preguntar	1	0	0	1	0	0	0
31	1	0	a la profesora y un compañero	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	pedi ayuda	0	1	0	1	0	0	0
32	1	0	un compañero	0	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	analice de nuevo	0	0	0	0	0	0	0
33	1	0	un compañero	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	ejercicios	0	1	0	1	0	0	0
34	1	0	a la profesora	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	corregirlos	1	0	0	1	0	0	0
35	1	0	un compañero	1	1	0	ambos	0	0	1	1	0	0	0	0	1	0	1	0	0	corregirlos	1	0	0	1	0	0	0
36	1	0	a la profesora	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	corregirlos	1	0	0	1	0	0	0
37	1	0	un compañero	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	corregirlos	1	0	0	1	0	0	0
38	1	0	a la profesora	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0		0	1	0	0	1	0	0	
39	0	1		0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0	corregirlos	1	0	0	0	0	0	1
40	1	0	a la profesora	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	corregirlos	1	0	0	0	0	0	0