

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“FRAMEWORK PARA LA ENSEÑANZA DE HABILIDADES DEL PENSAMIENTO COMPUTACIONAL EN EDADES TEMPRANAS EN BASE A ESTRATEGIAS UNPLUGGED”

SEBASTIÁN PATRICIO PEÑA COFRÉ

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Federico Meza
Profesora Correferente: Andrea Vásquez

Noviembre - 2024

DEDICATORIA

Dedicado a mi familia, amigos y a todos los que estuvieron presentes durante mi formación académica.

AGRADECIMIENTOS

En primer lugar quisiera agradecer a mi familia, en especial a mis padres Justiniano Peña y Ana Cofré y hermanos Joselyn Peña y Cristopher Peña, los cuales han estado ahí para brindarme apoyo durante todo este proceso y durante toda mi formación académica, aconsejándome, festejando mis logros y siendo un pilar para mi.

También me gustaría agradecer a mis compañeros de carrera y amigos, que han estado conmigo durante este proceso y durante la carrera en general siendo mis compañeros de proyectos, trabajos, dificultades y risas, aquellos que influenciaron en mi y me ayudaron a llegar a este punto. Una mención especial a Julio González, Matías Herrera, Benjamín Espinoza, Simón Ahumada, Mauricio Gaueca, Jeremmy Moya, Matías Bustamante y Claudio Lopez.

Finalmente me gustaría extender este agradecimiento a mi Profesor guía Federico Meza, por apoyarme durante todo este proceso, revisando el trabajo, proponiendo mejoras y aconsejándome para llegar al mejor resultado posible, sin esto y su paciencia durante el tiempo de desarrollo del trabajo, esta memoria no hubiese sido posible.

RESUMEN

Resumen—

Este trabajo propone un framework para la enseñanza de las habilidades del pensamiento computacional a través de una selección de estrategias pedagógicas unplugged, junto con una pauta de buenas prácticas para facilitar al educador su aplicación en contextos reales con un enfoque en la educación en edades tempranas. Para abordar esta propuesta se realizó una investigación profunda para seleccionar las mejores estrategias unplugged, diseñar el framework y recopilar actividades de ejemplo. Finalmente para validar la efectividad del framework se realizó un caso de uso en un colegio donde se capacitó a profesores y se desarrolló una actividad práctica junto con una pequeña encuesta. Los resultados muestran que estrategias como el storytelling y actividades kinestésicas pueden desarrollar estas habilidades en niños pequeños, y que el framework facilita a los educadores en Chile implementar el pensamiento computacional.

Palabras Clave— Pensamiento Computacional, Actividades Unplugged, Educación en edades tempranas.

ABSTRACT

Abstract— This work proposes a framework for teaching computational thinking skills through a selection of unplugged pedagogical strategies, along with a set of best practices to assist educators in applying it in real-world contexts, with a focus on early childhood education. To address this proposal, an in-depth investigation was carried out to select the best unplugged strategies, design the framework, and compile example activities. Finally, to validate the effectiveness of the framework, a case study was conducted in a school where teachers received training and carried out a practical activity along with a brief survey. The results show that strategies such as storytelling and kinesthetic activities can develop these skills in young children and that the framework helps educators in Chile implement computational thinking.

Keywords— Computational Thinking, Unplugged Activities, Early Childhood Education.

GLOSARIO

K-12: Designación de sistemas educativos de escolarización primaria y secundaria.

CT: Pensamiento computacional.

STEM: Science, Technology, Engineering and Mathematics.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Causas del problema	3
1.1.1 Dificultad en la enseñanza del CT sin programación y dispositivos electrónicos en edades tempranas	3
1.1.2 Falta de claridad sobre la enseñanza del CT en edades tempranas y su reciente incorporación a diferentes currículos educativos	3
1.2 Efectos del problema	4
1.2.1 Bajo rendimiento académico en asignaturas asociadas a programación y rechazo a estas	4
1.2.2 Falta de desarrollo de habilidades asociadas al CT y Falta de habilidades para resolver problemas	4
1.3 Objetivos	5
1.4 Impacto de la solución	5
CAPÍTULO 2: MARCO CONCEPTUAL	6
2.1 Pensamiento Computacional	6
2.1.1 Definición	6
2.2 Pensamiento Computacional en la educación	7
2.3 Educación en edades tempranas	10
2.4 Pensamiento Computacional en edades tempranas	10
2.5 Enseñanza del Pensamiento Computacional	11
2.6 Estrategias Unplugged	12
2.6.1 Beneficios de las estrategias unplugged en la educación del CT	12
2.6.2 Metáforas	13
2.6.3 Graph Paper Programming	13
2.6.4 StoryTelling	13
2.6.5 Juegos de Clasificación y Ordenación	14
2.6.6 Actividades de Role-Playing	14
2.6.7 Juegos de Estrategia y Resolución de Problemas	14
2.6.8 Simulación de Programación	15
2.6.9 Actividades Kinestésicas	15

2.6.10	Estrategias Mixtas	15
2.6.11	CS Unplugged	15
2.7	Framework Educativo	17
2.7.1	Constructivismo	17
2.7.2	El Enfoque Basado en Competencias	17
2.7.3	El Aprendizaje Cooperativo	18
2.7.4	Taxonomía de Bloom	18
2.7.5	Taxonomía de Bloom Original	19
2.7.6	Revisión de la Taxonomía	19
2.7.7	Zona de desarrollo próximo (ZPD)	20
2.8	Frameworks en la educación del pensamiento computacional	21
2.8.1	The Three Dimensional Framework	21
2.8.2	The Three Dimensional Framework Redeveloped for ECE	22
2.8.3	The CT-cube	24
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN		27
3.1	Selección de estrategias unplugged	28
3.1.1	Criterios de selección:	28
3.2	Análisis de frameworks existentes	32
3.3	Diseño de Framework en base a las estrategias seleccionadas	34
3.4	Diseño de pauta de buenas practicas para cada estrategia	36
3.4.1	Introducción de conceptos y evaluación inicial	37
3.4.2	Pauta de buenas prácticas para cada habilidad	37
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		47
4.1	Capacitación	47
4.2	Actividad Práctica	48
4.3	Resultados	49
CAPÍTULO 5: CONCLUSIONES		54
5.1	Alcances y Aprendizajes	54
5.2	Trabajo a futuro y posibles mejoras	55
ANEXOS		57
5.1	Imágenes de documentación sobre el caso de uso: Resultados y Guía entregada.	57
REFERENCIAS BIBLIOGRÁFICAS		72

ÍNDICE DE FIGURAS

1	Árbol del Problema.	5
2	Los 4 pilares del pensamiento computacional.	7
3	A five-step research plan for CT Education.	9
4	CT curriculum framework for early childhood education: A three-dimensional model.	24
5	The CT-cube	25
6	Framework para la enseñanza de las habilidades del pensamiento computacional en base a estrategias unplugged.	36
7	Ejemplo de mapa.	38
8	Ejemplo de actividad de descomposición.	39
9	Ilustración visual del algoritmo por selección.	41
10	Ilustración visual de los pasos de la cueca.	45
11	Hoja de Respuestas 1.	57
12	Hoja de Respuestas 2.	58
13	Hoja de Respuestas 3.	59
14	Hoja de Respuestas 4.	60
15	Hoja de Respuestas 5.	61
16	Hoja de Respuestas 6.	62
17	Hoja de Respuestas 7.	63
18	Hoja de Respuestas 8.	64
19	Hoja de Respuestas 9.	65
20	Guía entregada a participantes Hoja 1.	66
21	Guía entregada a participantes Hoja 2.	67
22	Guía entregada a participantes Hoja 3.	68
23	Guía entregada a participantes Hoja 4.	69
24	Guía entregada a participantes Hoja 5.	70
25	Guía entregada a participantes Hoja 6.	71

ÍNDICE DE TABLAS

1	Fuerzas y limitaciones de los acercamientos para evaluación.	22
2	Tabla resultados preguntas	53

INTRODUCCIÓN

En los últimos años, el pensamiento computacional ha ganado importancia dado su reconocimiento como una habilidad clave del siglo XXI. Es por esto que su enseñanza se ha convertido en un tema de estudio transversal en todos los niveles educativos. En este contexto, se aborda la problemática de enseñar el pensamiento computacional, específicamente, desarrollar las habilidades asociadas a este desde edades tempranas, tales como el reconocimiento de patrones, pensamiento algorítmico, abstracción y descomposición.

Es en este contexto donde surgen diversas metodologías, estrategias pedagógicas y marcos de trabajo que buscan contribuir a la educación del pensamiento computacional en edades tempranas. En este trabajo, se aborda un tipo de estrategias en particular: las unplugged, es decir, aquellas que no requieren el uso de dispositivos electrónicos.

En primer lugar, se realiza un estudio exhaustivo de la literatura asociada al tema, buscando identificar elementos clave que permitan el diseño del marco de trabajo y definir criterios para la selección de las mejores estrategias.

Luego, mediante la comparación de estudios y tomando en consideración los criterios previamente definidos, se seleccionan estas estrategias, recopilando a su vez actividades de ejemplo que facilitan la creación de una pauta de buenas prácticas.

Tras un análisis de cada estrategia y de las actividades, se identificaron las características que componen las mejores actividades, con el objetivo de integrar esto en el marco de trabajo, enfocándose en la educación en edades tempranas.

Finalmente, este trabajo presenta un caso de uso para validar tanto el marco de trabajo propuesto como la pauta de buenas prácticas diseñada, aplicándolos en dos sesiones con profesores de un colegio de la comuna de San Bernardo: una primera sesión teórica y una segunda sesión práctica.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

"En los últimos años, ha habido un creciente interés en la educación del CT en las escuelas K-12 (desde preescolar hasta el 12° grado), y su papel en la adquisición de habilidades de pensamiento y competencias digitales por parte de los niños. En respuesta a esta necesidad, el pensamiento computacional y la programación se han convertido en una parte integral de los planes de estudio escolares de muchos países en años recientes." [Charoula Angeli, 2020]

El pensamiento computacional entendido como un conjunto de habilidades importantes para la resolución de problemas, que cualquier persona puede aprender [Wing, 2006], últimamente ha ganado gran popularidad y repercusión debido a su integración como habilidad clave del siglo 21 [Zhou, 2021], por lo que muchos sistemas educativos se encuentran integrando el pensamiento computacional para su enseñanza en los distintos niveles educativos.

El pensamiento computacional se encuentra directamente relacionado con la programación ya que las habilidades que se busca desarrollar en los alumnos como, por ejemplo: Reconocimiento de patrones, Descomposición de problemas, Pensamiento algorítmico y abstracción son habilidades que se pueden desarrollar a través de la realización de ejercicios de programación o robótica, sin embargo, la programación no siempre es la mejor manera de introducir estas habilidades y conceptos en los alumnos, esto ocurre por diferentes motivos, siendo los principales: La dificultad y tiempo que lleva el aprender a programar y que normalmente no se enseña a programar en niveles educativos bajos, lo que en Chile vendría a ser la educación pre básica y básica. Lo anterior genera una problemática que debe ser abordada por los educadores que se refiere a la dificultad de enseñar las habilidades del pensamiento computacional sin ayuda de la programación en edades tempranas.

Es aquí donde surgen diferentes estrategias que permiten resolver esta problemática, las llamadas estrategias "unplugged"¹ [Brackmann, 2017] que se refieren a las estrategias pedagógicas que no requieren de un dispositivo como intermediario para enseñar los conceptos básicos del pensamiento computacional y programación y que también permiten desarrollar las habilidades asociadas a este. Sin embargo, aún falta claridad sobre cuáles de estas estrategias son las mejores y más eficientes y sobre todo como implementarlas de manera óptima, para lo cual se propone diseñar un framework basado en este tipo de estrategias que permita enseñar y desarrollar las habilidades del pensamiento computacional en edades tempranas, donde se definan las mejores estrategias, y como llevarlas a la práctica.

¹Estrategias unplugged hace referencia a estrategias y actividades que no requieran de un computador o dispositivos para realizarse

1.1. Causas del problema

1.1.1. Dificultad en la enseñanza del CT sin programación y dispositivos electrónicos en edades tempranas

Según Brackman et al. [Brackmann, 2017] actualmente existen dos metodologías de enseñanza del CT y las habilidades asociadas a este que se están estudiando a fondo, estas son a través de actividades "plugged-in" y "unplugged". Aunque la estrategia predominante para enseñar habilidades de pensamiento computacional se basa principalmente en actividades computarizadas que abordan diversas tareas de programación, es decir, "plugged-in", los educadores y académicos también están recurriendo a otro enfoque mediante actividades "unplugged" (es decir, que no requieren el uso de dispositivos digitales). Estas actividades incluyen juegos lógicos, cartas o movimientos físicos que sirven para representar y comprender conceptos informáticos como algoritmos o transmisión de datos.

Las actividades y estrategias "unplugged" no han sido muy investigadas sino hasta años recientes, donde se buscó en primer lugar validar que fueran efectivas [Brackmann, 2017], para luego abordar su efectividad en diferentes niveles educacionales [Javier del Olmo Muñoz, 2020], esto provoca que existan dificultades a la hora de crear actividades efectivas para desarrollar el CT y las habilidades asociadas a este.

1.1.2. Falta de claridad sobre la enseñanza del CT en edades tempranas y su reciente incorporación a diferentes currículos educativos

Existen diversos estudios que han buscado validar la efectividad de estrategias para enseñar el CT en edades tempranas, tanto con actividades "plugged-in" como "unplugged", sin embargo aun no existe claridad sobre como plantear estas estrategias en el aula, esto ya que aun no se presenta un consenso en "Frameworks" o definiciones del CT en edades tempranas [Jiahong Su, 2023] y también debido a que su relevancia ha aumentado recientemente con su incorporación a currículos escolares como el K-12.

En Chile en particular el CT viene a tomar relevancia en la educación recién en 2019 con la creación del plan "Chile Aprende Más" donde se incluye el Plan Nacional de Lenguajes Digitales, el cual busca promover la enseñanza del pensamiento computacional y la programación en el sistema educativo para potenciar las habilidades de resolución de problemas e innovación en ambientes tecnológicos.

1.2. Efectos del problema

1.2.1. Bajo rendimiento académico en asignaturas asociadas a programación y rechazo a estas

Una falta de una base solida en cuanto a CT provoca que asignaturas asociadas a programación y/o matemáticas se dificulten para los alumnos, esto puede producir bajo rendimiento académico y un rechazo a este tipo de asignaturas, algo recurrente en el caso de matemáticas donde alumnos asocian frustración y sentimientos negativos a dichas asignaturas lo cual dificulta aún mas el aprendizaje ya que los procesos emocionales están profundamente involucrados en la dirección de los procesos cognitivos: la atención, todos los actos perceptivos, el aprendizaje y la memoria [Jadue, 2002]

1.2.2. Falta de desarrollo de habilidades asociadas al CT y Falta de habilidades para resolver problemas

El CT esta formado por 4 principales habilidades Descomposición, Reconocimiento de Patrones, Abstracción y Pensamiento Algorítmico. Cada una de estas habilidades resulta de mucha utilidad para resolver problemas de alta complejidad, que no necesariamente están asociados a la programación y/o matemáticas [Wing, 2006] por ende no desarrollar estas habilidades limita al alumno en su capacidad de resolver problemas o de encontrar diferentes maneras para lograrlo. Estas habilidades se buscan desarrollar al incluir el CT en los currículos escolares.

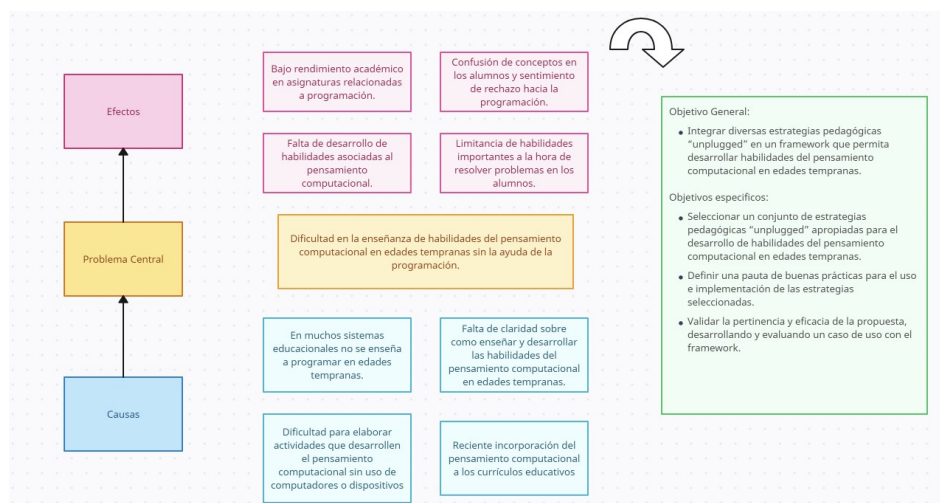


Figura 1: Árbol del Problema.
Fuente: Elaboración propia.

1.3. Objetivos

Objetivo General: Integrar diversas estrategias pedagógicas "unplugged" en un framework que permita desarrollar habilidades del pensamiento computacional en edades tempranas.

Objetivos específicos:

- Seleccionar un conjunto de estrategias pedagógicas "unplugged" apropiadas para el desarrollo de habilidades del pensamiento computacional en edades tempranas.
- Definir una pauta de buenas prácticas para el uso e implementación de las estrategias seleccionadas.
- Validar la pertinencia y eficacia de la propuesta, desarrollando y evaluando un caso de uso con el framework.

1.4. Impacto de la solución

Actualmente la problemática mencionada anteriormente afecta a los sistemas educativos que han integrado la enseñanza del pensamiento computacional en su programa educativo, esto incluye diversos países y cabe destacar que cada vez serán mas a futuro, por lo que entregar una alternativa de solución a esta problemática es algo que beneficiaría a muchos niños para que logren desarrollar las habilidades del CT de una manera adecuada y sin necesidad de invertir en tecnología para la educación.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Pensamiento Computacional

2.1.1. Definición

El pensamiento computacional es un término mencionado desde la década de los 50, sin embargo, diversos autores coinciden en que, como concepto, fue propuesto por primera vez por Papert en 1980 [Papert, 1980]. Luego, años más tarde, Wing en 2006 propone un cambio sustancial en el concepto del pensamiento computacional, definiéndolo como un proceso que envuelve resolver problemas, diseñar sistemas y entender el comportamiento humano, todo esto a través de los conceptos fundamentales de la ciencia de la computación [Wing, 2006]. Esta nueva forma de ver el pensamiento computacional abrió nuevas oportunidades para su estudio, ya que se generaliza el concepto, alejándolo de estar directamente relacionado con la informática y la programación, convirtiéndolo en algo importante a desarrollar por cualquier persona. Más tarde, Wing continúa desarrollando el concepto, ampliándolo aún más, considerando que el pensamiento computacional comparte el proceso de solución de problemas con el pensamiento matemático, ingenieril y científico [Wing, 2008], y finalmente relacionándolo con el pensamiento lógico y sistemas de pensamiento [Wing, 2011].

El pensamiento computacional es también considerado como un conjunto de habilidades, tales como reconocimiento de patrones, descomposición de problemas, pensamiento algorítmico y abstracción [Wing, 2006]. Inicialmente, se consideraba que estas habilidades estaban totalmente relacionadas con las ciencias de la computación y la programación; sin embargo, con el trabajo y cambio de conceptos propuestos por Wing y más tarde por los trabajos de Mannila et al. [Mannila, 2014], se ha dado un nuevo enfoque donde estas habilidades pueden ser desarrolladas para resolver problemas en diferentes áreas, incluyendo o no el uso de computadores.

Las habilidades anteriormente mencionadas también han sido catalogadas como los 4 pilares del pensamiento computacional que se presentan para alcanzar el objetivo principal, que es resolver problemas [Susana., 2016].

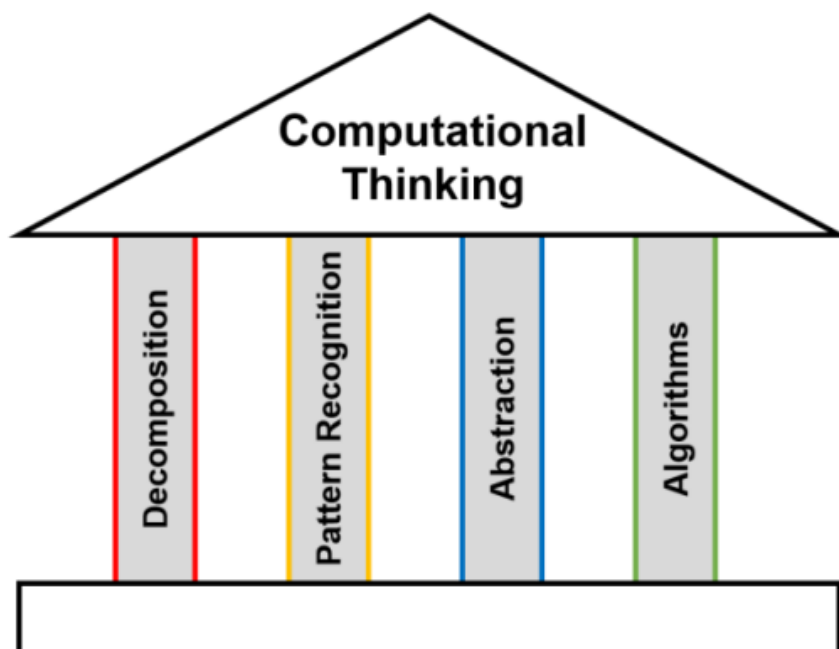


Figura 2: Los 4 pilares del pensamiento computacional.
Fuente: Computational Thinking: Panorama of the Americas.

- **Descomposición:** Comprende la habilidad de identificar un problema complejo y separarlo en pequeñas partes o subproblemas más fáciles de resolver.
- **Reconocimiento de Patrones:** Comprende la habilidad de encontrar regularidades en datos y utilizarlos, o también encontrar similitudes con problemas anteriores para encontrar una solución.
- **Abstracción:** Comprende la habilidad de enfocarse en los puntos importantes dejando de lado la información irrelevante de un problema para así crear un modelo general que permita resolver el problema más fácilmente.
- **Algoritmos:** Comprende la habilidad de identificar pasos secuenciales o instrucciones que dado cierto orden de realización permitan resolver el problema.

2.2. Pensamiento Computacional en la educación

En años recientes, la enseñanza del pensamiento computacional y el desarrollo de las habilidades relacionadas a este ha tomado gran importancia, esto debido al creciente desarrollo de tecnologías y a la generalización y ampliación del concepto a diferentes áreas del conocimiento. De esta manera, investigadores han buscado maneras de enseñar los conceptos y promover el desarrollo de las habilidades del CT en alumnos de todos los niveles educativos,

proponiendo diseños de aprendizaje que incorporan diferentes estrategias y herramientas. Dentro de estos acercamientos, la programación ha sido lo más usado para el desarrollo del pensamiento computacional [Sun, 2023].

En base al contexto anterior, en diversos sistemas educativos se está integrando la enseñanza del pensamiento computacional, en particular en el K-12. También se destaca su integración en la educación STEM [diSessa A.A. et al, 2020], por lo que diversos países ya se encuentran creando iniciativas y planes para impulsar su enseñanza y añadirlo en los currículos educativos. En Europa, por ejemplo, se encuentra el "Digital Education Action Plan", una iniciativa que busca ayudar a las personas, instituciones educativas y sistemas educativos a adaptarse al rápido cambio digital; por lo tanto, el pensamiento computacional ha estado en la agenda europea por varios años [Earp, 2018]. Países como Inglaterra y Finlandia han promovido la enseñanza de la programación y el pensamiento computacional, siendo Inglaterra uno de los primeros en agregarlo al K-12; luego, países como Japón y Corea se encuentran en camino a agregar la programación como asignatura obligatoria en sus sistemas educativos [Wu, 2019].

En Estados Unidos, en el año 2015, se presenta la ley federal "Every Student Succeeds Act", en la cual se coloca a la computación al mismo nivel que otras asignaturas como matemáticas y ciencias. Esto provocó que los estados que adoptaron esta ley integraran el programa "A Model Curriculum for K-12 Computer Science", el cual presenta una estructura para enseñar computación en el sistema K-12 [Susana., 2016].

Se destaca entonces la importancia que ha tomado el pensamiento computacional en la educación; sin embargo, en muchos países aún no se toman iniciativas grandes que promuevan su enseñanza. Por ejemplo, en Chile, este se encuentra en la línea de innovación como el programa de Pensamiento Computacional y Programación propuesto por la Unidad de Currículum y Evaluación, y existen iniciativas bajo la misma línea como:

- El Plan nacional de Lenguajes Digitales planteado por el Centro de Innovación Educativa.
- La Hora del Código, iniciativa donde participan millones de estudiantes en más de 180 países, incluyendo Chile.
- El programa de Modelamiento de Incendios Forestales, de la Unidad de Curriculum y Evaluación y CIAE.

A pesar de estas iniciativas en Chile los estudiantes se encuentran alejados aún de los conceptos y aprendizaje de computación y pensamiento computacional, especialmente aquellos en contextos de vulnerabilidad donde se tiene menos acceso a tecnología.[Susana., 2016]

En 2020 Charoula Angeli realizó un estudio sobre los problemas y desafíos del pensamiento computacional en la educación [Charoula Angeli, 2020], donde presenta diversas ideas que

podrían ser trabajadas a futuro para complementar la enseñanza del pensamiento computacional y mejorarla.

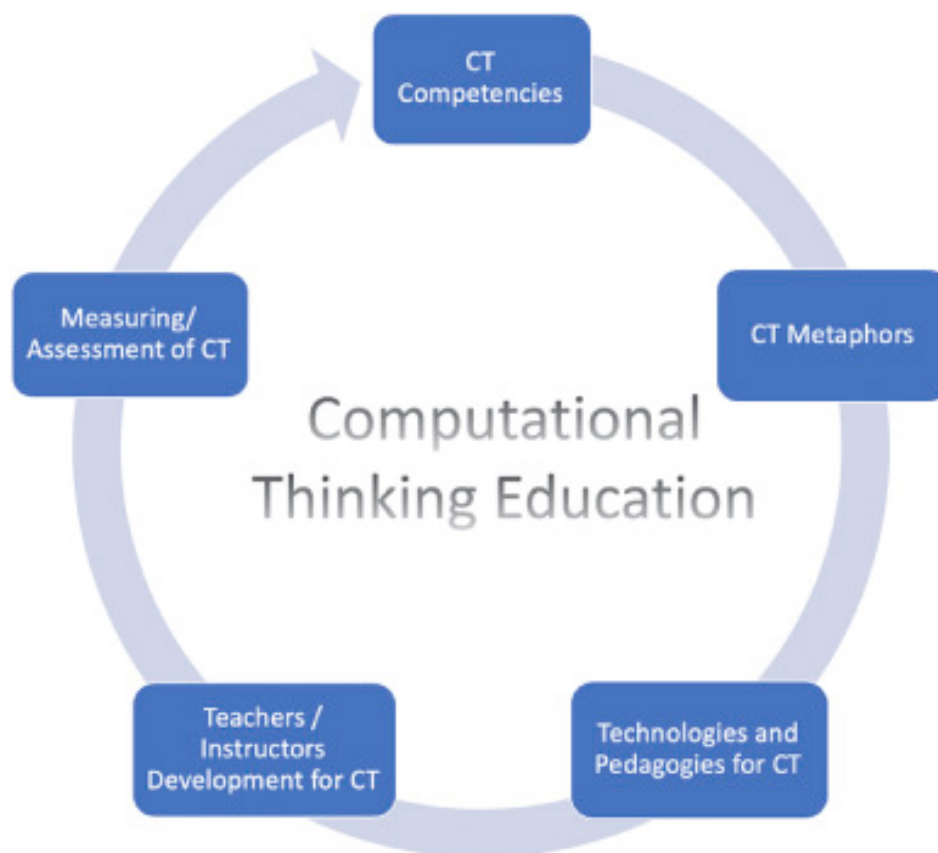


Figura 3: A five-step research plan for CT Education.
Fuente: Computational thinking education: Issues and challenges.

De la figura anterior se destaca el papel de Tecnologías y Pedagogías para CT, donde se presentan los posibles caminos para la enseñanza del CT, y se refuerza la importancia de buscar diferentes estrategias para impulsar el desarrollo de las habilidades del CT en los diferentes niveles educativos. También es importante mencionar el concepto de Metáforas de CT, que se refiere al uso de metáforas y actividades que permitan a alumnos en edades tempranas el aprender conceptos de CT de una manera más amena, este concepto será tratado en profundidad más adelante ya que considera actividades unplugged para la enseñanza de las habilidades del CT.[Charoula Angeli, 2020] Finalmente otro concepto importante es el desarrollo de profesionales para la enseñanza del CT, en este sentido se explica la necesidad de instruir a profesores en la enseñanza del CT, y también buscar herramientas que faciliten aquello sobre todo para la educación en edades tempranas.

2.3. Educación en edades tempranas

2.4. Pensamiento Computacional en edades tempranas

Con la reciente integración del pensamiento computacional a distintos currículos educativos, como lo es, por ejemplo, el K-12 que incluye desde preescolar hasta el 12° grado, se empieza a destacar la importancia de incluir el pensamiento computacional en alumnos en edades tempranas, dado los beneficios que esto conlleva [Jiahong Su, 2023]. El término edad temprana se refiere a los primeros años de educación en las personas, aunque esto puede variar dependiendo del país y sistema educacional. La Asociación Nacional para la Educación de Niños Pequeños (NAEYC) de Estados Unidos define como edad temprana desde el nacimiento hasta los 8 años; sin embargo, por cuestiones prácticas, normalmente los estudios realizados acotan esta edad desde los 3 o 4 años hasta los 7 u 8. Con el objetivo de integrar estrategias existentes en la literatura, se definirá para el trabajo futuro que la edad temprana incluya niños desde los 4 a los 8 años, lo que en Chile incluye enseñanza prebásica y los primeros tres años de enseñanza básica.

En los últimos años se han realizado muchas investigaciones sobre la educación del pensamiento computacional en edades tempranas. Diferentes autores se han dedicado a discutir sobre la importancia del aprendizaje del pensamiento computacional en etapas tempranas de la niñez, desde las guarderías hasta la educación primaria, y los beneficios potenciales que esto podría traer a los alumnos. Estudios revelan que niños en edades entre 3 y 6 años son capaces de construir y programar robots [Sullivan, 2014] y que este tipo de actividades y aprendizaje les pueden traer beneficios relacionados con el desarrollo de las habilidades asociadas al pensamiento computacional y también al trabajo en equipo, creatividad y colaboración [Jiahong Su, 2023].

Dentro de las oportunidades y posibles beneficios investigados encontramos los siguientes [Jiahong Su, 2023]:

- Aprendizaje y desarrollo de las habilidades asociadas al pensamiento computacional: Desarrollar las habilidades antes mencionadas, descomposición, reconocimiento de patrones, abstracción y pensamiento algorítmico, que les proveen a los alumnos otra forma de resolver problemas.
- Estrategias para resolver problemas: Mediante el desarrollo del pensamiento computacional, los alumnos aprenden diferentes formas de resolver problemas, lo cual les ayuda a mejorar su rendimiento académico a futuro y les facilitará el aprendizaje de otras asignaturas.
- Habilidades no cognitivas como pensamiento crítico, colaboración y comunicación: Mediante las diferentes formas de aprender conceptos de computación, como lo son el trabajo con robots simples, actividades colaborativas y juegos didácticos, los alumnos pueden desarrollar otro tipo de habilidades blandas fundamentales.

- Creatividad y curiosidad: A través de la interacción con objetos tecnológicos o mediante las actividades propuestas para desarrollar el pensamiento computacional, los alumnos impulsan su creatividad y curiosidad.

En este caso, la dificultad radica en cómo enseñar estos conceptos e incentivar el desarrollo de las habilidades del pensamiento computacional en niños, sin ayuda de la programación, buscando alternativas con otro tipo de actividades y estrategias. Otras dificultades encontradas son la falta de especialización de profesores, escasas formas válidas de evaluar el desarrollo del pensamiento computacional en niños pequeños y la motivación de los alumnos para el aprendizaje.

2.5. Enseñanza del Pensamiento Computacional

Dentro de las investigaciones realizadas se encuentran diferentes formas de enseñar el pensamiento computacional y desarrollar las habilidades asociadas a este. Como fue mencionado anteriormente, la manera más popular es a través de la programación; es por esta razón que diversos países integran cursos de programación en sus currículos. Sin embargo, existen alternativas más fácilmente accesibles. A continuación se presenta un listado de las formas más populares para la enseñanza del pensamiento computacional:

- Programación: Los conceptos del pensamiento computacional y en particular las habilidades asociadas a este están directamente relacionadas con la programación por lo que aprender un lenguaje de programación y desarrollar actividades y soluciones con este impulsan el desarrollo de estas habilidades.[Wing, 2008]
- Robótica: A través de la construcción y programación de robots los alumnos pueden desarrollar las habilidades asociadas al pensamiento computacional y colaborar con trabajo en equipo.[Sullivan, 2014]
- Lenguajes de programación por bloques: Como una alternativa más accesible a la programación directa se tiene la programación por bloques, lenguajes como Scratch Jr se han popularizado debido a su facilidad de aprendizaje y a que permite introducir los conceptos de programación y pensamiento computacional. Se destaca su uso en edades tempranas debido a diversas investigaciones al respecto que demuestran empíricamente que estos lenguajes permiten a los niños aprender más fácilmente.[Tsavara, 2017]
- Actividades "unplugged": Son las actividades que se realizan sin la necesidad de un dispositivo, es decir actividades desenchufadas, estas buscan incentivar a los alumnos mediante actividades lúdicas que presenten los conceptos de computación y desarrollen las habilidades del pensamiento computacional en los alumnos.[Zhou, 2021]

Además del listado anterior se han realizado diversos estudios sobre alternativas para que los alumnos puedan aprender y desarrollar las habilidades relacionadas al CT, por ejemplo, a través de juegos [Tsavara, 2017], evaluando el uso de metáforas para la enseñanza de conceptos [Pizarro, 2020], entre otros.

2.6. Estrategias Unplugged

Dentro de los estudios sobre las diversas formas de enseñar y desarrollar las habilidades asociadas al pensamiento computacional, se encuentra un área que propone estrategias y actividades que no requieran del uso de computadores o dispositivos; a esto se le llama estrategias “unplugged” [Javier del Olmo Muñoz, 2020]. Diversos autores se han encargado de validar y estudiar la efectividad de este tipo de actividades a la hora de introducir el pensamiento computacional en alumnos de diferentes niveles educativos. En este sentido, se destaca el trabajo de Brackman et al. [Brackmann, 2017] y Tsavara et al. [Tsavara, 2017], quienes estudiaron el efecto de este tipo de estrategias y actividades en alumnos de educación primaria, logrando resultados positivos, sobre todo a la hora de motivar a los alumnos a aprender a través de actividades dinámicas y diferentes. Diversas estrategias pedagógicas se han puesto a prueba en estos estudios, desde actividades donde los alumnos proponen instrucciones a sus pares para hacer un dibujo [Javier del Olmo Muñoz, 2020], hasta introducir conceptos clave del pensamiento computacional mediante “storytelling” o contar historias donde partes de la historia incluyen conceptos como “loops”, algoritmos, etc. [Jiahong Su, 2023].

A pesar de los esfuerzos y validaciones propuestas, aún no se llega a un consenso general sobre cómo aplicar estas estrategias y cuáles son las mejores [Javier del Olmo Muñoz, 2020]; sin embargo, los estudios demuestran que su efectividad existe, destacando su papel en la enseñanza del pensamiento computacional en edades tempranas.

2.6.1. Beneficios de las estrategias unplugged en la educación del CT

Como fue mencionado anteriormente, diversos estudios se han realizado para verificar los beneficios asociados a introducir el CT mediante actividades unplugged. En este ámbito, el trabajo de Javier del Olmo Muñoz et al. [Javier del Olmo Muñoz, 2020] describe tres grandes beneficios validados empíricamente. En primer lugar, se tiene el aprendizaje de los conceptos de CT y el desarrollo de las habilidades relacionadas a este. En este caso, se demostró mediante una comparativa que los alumnos que recibieron educación con actividades unplugged mejoraron más sus habilidades asociadas al CT, algo que Brackmann et al. en 2017 también había sugerido [Brackmann, 2017]. Luego se tiene la motivación de los alumnos para aprender. En este sentido, los descubrimientos hechos fueron que los alumnos se sienten más motivados a aprender con actividades que no involucren programar directamente, sino a través de juegos o actividades que lo hagan de manera indirecta, obteniendo resultados similares en cuanto a aprendizaje y desarrollo de las habilidades del CT. Finalmente, se en-

cuentra la participación y motivación con respecto al género del alumno; en este caso, el estudio verifica que actividades unplugged o mixtas, unplugged seguidas de plugged-in, logran motivar más tanto a niños como a niñas, generando un interés mayor por parte de las niñas. Todos estos beneficios se presentan comparando grupos de niños donde se aplicaron actividades unplugged y plugged-in a forma de introducción al CT en edades tempranas.

2.6.2. Metáforas

Dentro de las alternativas más mencionadas en la literatura de estrategias unplugged, se tiene el uso de metáforas; esto se refiere a actividades que, a través de juegos, logran introducir conceptos como loops, algoritmos, etc. en los alumnos, sin describirlos de forma directa.

Un ejemplo del uso de metáforas es la actividad "Monica's Map" presentada en el trabajo de Brackmann et al. [Brackmann, 2017], donde los alumnos deben encontrar el camino más corto para llevar ciertos personajes a un punto en un mapa, todo esto a través de instrucciones como: arriba, abajo, izquierda y derecha. De esta forma, los alumnos trabajan el reconocimiento de patrones y el pensamiento algorítmico.

Recientemente, también se ha planteado el uso de metáforas usando el cuerpo, con actividades que impliquen que los niños se muevan y colaboren. Estudios como el de Andrew Manches et al. [Manches, 2020] buscan encontrar las mejores metáforas para la enseñanza del CT.

2.6.3. Graph Paper Programming

En estudios recientes se ha encontrado presente un tipo de actividad que contempla el uso de dibujos para desarrollar habilidades del CT. Existen diversos tipos; por ejemplo, en el estudio de Javier del Olmo et al. [Javier del Olmo Muñoz, 2020], se utiliza una actividad donde un alumno da instrucciones a otro para realizar un dibujo. El otro alumno debe seguir estas instrucciones intentando dibujar exactamente lo que pidió el primero. Otro ejemplo es la actividad Repetition Drawing [Brackmann, 2017], un ejercicio individual donde se presenta un dibujo con una flecha y los alumnos deben escribir un comando que permita completar el dibujo usando instrucciones de arriba, abajo, izquierda y derecha. Este tipo de actividades potencia la descomposición, la abstracción, el reconocimiento de patrones y el pensamiento algorítmico.

2.6.4. StoryTelling

El contar historias (storytelling) ha sido un método de enseñanza estudiado para diferentes áreas de la educación. En el caso del CT, estudios como el de Weipeng Yang et

al.[Weipeng Yang, 2023] investigan el impacto que puede llegar a tener el aplicar conceptos de programación y CT a través del storytelling, demostrando la motivación y participación de los alumnos, así como también el desarrollo de habilidades como la abstracción y el reconocimiento de patrones. En este tipo de actividades se cuentan historias que presentan patrones o repetitividad y se busca que el alumno identifique estos patrones. Otro enfoque es la creación de historias a partir de instrucciones o la introducción de conceptos del CT en las historias contadas.

2.6.5. Juegos de Clasificación y Ordenación

Dentro de las posibilidades a la hora de crear una actividad que permita desarrollar las habilidades asociadas al pensamiento computacional, se encuentran los juegos. Existen diversos tipos de juegos posibles a la hora de enseñar reconocimiento de patrones, algoritmos y descomposición. Uno de estos son los juegos de clasificación y ordenación; en estos, el objetivo es enseñar al alumno diferentes tipos de algoritmos de ordenamiento, desde los más básicos hasta algunos más complejos, de esta forma potenciando el desarrollo de las habilidades mencionadas anteriormente. Ejemplos de este tipo de estrategia se encuentran en el libro CS Unplugged.

2.6.6. Actividades de Role-Playing

Un tipo de estrategia que se utiliza para enseñar conceptos de programación es mediante actividades donde se simulen procesos computacionales, asignándole roles a cada alumno. De esta forma, es posible recrear escenarios donde los alumnos formen parte del proceso y puedan interiorizar los conceptos y desarrollar habilidades asociadas al pensamiento computacional. Con este tipo de actividades se pueden desarrollar las habilidades de algoritmos y reconocimiento de patrones, así como también la abstracción.

2.6.7. Juegos de Estrategia y Resolución de Problemas

Los juegos de estrategia permiten desarrollar el pensamiento crítico de los alumnos, fomentando habilidades importantes en la resolución de problemas, en este caso, en especial la abstracción y el reconocimiento de patrones, además de potenciar la motivación de los alumnos mediante la gamificación.

2.6.8. Simulación de Programación

Simular la programación haciendo uso de tarjetas u objetos que representen comandos comunes, esto para introducir, por ejemplo, algoritmos de forma didáctica, sin la necesidad de una computadora. Los alumnos, entonces, resuelven problemas mediante el uso de estos objetos.

2.6.9. Actividades Kinestésicas

Actividades o juegos realizables tanto en aula como al aire libre, donde los alumnos aprendan conceptos y desarrollen habilidades del pensamiento computacional realizando actividades que involucren movimiento y trabajo en equipo. De esta manera, es posible promover la motivación de los alumnos y desarrollar pensamiento algorítmico, descomposición y reconocimiento de patrones.

2.6.10. Estrategias Mixtas

Este tipo de estrategias comprenden el uso de actividades unplugged para introducir conceptos y, luego, para desarrollarlos a fondo, se utilizan ejercicios de programación con herramientas como ScratchJr o lenguajes de programación por bloques similares[Javier del Olmo Muñoz, 2020]. Los beneficios de este tipo de estrategias provienen de que al introducir la programación por bloques se puede dar mayor profundidad al aprendizaje y se pueden poner en práctica los conceptos y habilidades aprendidas, así como también se puede evaluar de mejor manera el aprendizaje de los alumnos.

2.6.11. CS Unplugged

CS Unplugged es un proyecto educativo creado por Tim Bell, Mike Fellows e Ian Witten, diseñado para enseñar conceptos fundamentales de ciencias de la computación a través de actividades prácticas que no requieren el uso de computadoras. El objetivo principal es hacer que la informática sea accesible y comprensible para todos, especialmente para niños y estudiantes jóvenes, mediante la utilización de juegos, actividades y ejercicios interactivos.

CS Unplugged cuenta con un libro completo con actividades unplugged que permiten enseñar conceptos y desarrollar habilidades asociadas al pensamiento computacional. Entre los tipos de actividades se encuentran:

- Algoritmos
- Codificación y Decodificación
- Compresión de Datos
- Teoría de Grafos
- Representación de Datos
- Computación Paralela

Cabe destacar que las actividades en este libro están categorizadas por la habilidad a trabajar y por la edad del alumno. A continuación, se detalla una actividad incluida en el libro asociada a los números binarios:

Cuenta los Puntos—Números Binarios

Resumen: Los datos en las computadoras se almacenan y transmiten como una serie de ceros y unos. ¿Cómo podemos representar palabras y números usando solo estos dos símbolos?

Matemáticas: Números: Exploración de números en otras bases. Representación de números en base dos. Matemáticas: Álgebra: Continuación de un patrón secuencial y descripción de una regla para este patrón. Patrones y relaciones en potencias de dos.

Habilidades:

- Contar
- Emparejar
- Secuenciar

Edad: 6 años en adelante

Materiales:

Un conjunto de cinco tarjetas binarias para la demostración. Tarjetas A4 con pegatinas de caritas funcionan bien. Cada estudiante necesitará un conjunto de cinco tarjetas. Fotocopia del maestro: Números binarios en cartulina, recortadas. Objetivo: Se pide a los alumnos utilizar las cartas para lograr representar un número de forma binaria.

La actividad Cuenta los Puntos trabaja desde un punto de vista matemático el reconocimiento de patrones y está catalogada para niños de 6 años en adelante. Estas actividades no están categorizadas por tipo; en este caso particular, se trata de una actividad que podría encontrarse dentro del tipo Juegos de Tarjetas y Tableros.

2.7. Framework Educativo

En el ámbito educativo, un framework o marco de trabajo se refiere a una estructura conceptual que sirve como guía para la organización y desarrollo de procesos educativos. Estos frameworks proporcionan una base sólida para la planificación, implementación y evaluación de programas educativos, asegurando que se alineen con objetivos pedagógicos claros y eficaces.

Un framework para educación se define como un conjunto estructurado de principios, teorías, metodologías y prácticas que orientan el diseño y la ejecución de actividades educativas. Estos marcos sirven como herramientas para los educadores, permitiéndoles crear experiencias de aprendizaje coherentes y alineadas con metas educativas específicas.

Según el trabajo de [McTighe, 2005], un framework educativo es esencial para la comprensión profunda de los conceptos y habilidades que los estudiantes deben adquirir. Estos frameworks ayudan a los educadores a planificar sus lecciones de manera que fomenten la comprensión y el pensamiento crítico, en lugar de la simple memorización de información.

A continuación se describen algunos frameworks educativos presentes en la literatura:

2.7.1. Constructivismo

El constructivismo es un enfoque teórico que sostiene que el conocimiento se construye activamente por el aprendiz, a través de la interacción con su entorno y la reflexión sobre esas experiencias.

En un marco constructivista, los estudiantes son considerados agentes activos en su proceso de aprendizaje. Se enfatiza la importancia de las experiencias previas y el contexto social en la construcción del conocimiento.

El trabajo de Lev Vygotsky, que introduce conceptos como la "Zona de Desarrollo Próximo" (ZDP), ha sido fundamental para los enfoques constructivistas en educación. Según Vygotsky, los estudiantes aprenden mejor cuando trabajan dentro de su ZDP, con la ayuda de andamiajes proporcionados por un mentor o compañero más competente [Vygotsky, 1978].

2.7.2. El Enfoque Basado en Competencias

Este enfoque se centra en el desarrollo de competencias específicas que los estudiantes deben adquirir para desempeñarse eficazmente en diversos contextos. Las competencias incluyen conocimientos, habilidades y actitudes que se consideran esenciales para el éxito académico y profesional. El enfoque basado en competencias es común en la educación técnica y profesional, pero también se está adoptando en otros niveles educativos.

Las competencias se definen como un conjunto integrado de conocimientos, habilidades y actitudes que permiten a una persona realizar una actividad de manera eficaz y eficiente. En el contexto educativo, estas competencias pueden abarcar desde habilidades técnicas específicas hasta capacidades transversales como el pensamiento crítico, la resolución de problemas, la comunicación y el trabajo en equipo.

La implementación del enfoque basado en competencias requiere una planificación cuidadosa y la participación de diversos actores educativos. Involucra la identificación de las competencias clave que se desean desarrollar, el diseño de actividades de aprendizaje que fomenten estas competencias, y la creación de sistemas de evaluación que puedan medir de manera efectiva el desarrollo de las mismas.[Sergio, 2006]

2.7.3. El Aprendizaje Cooperativo

El aprendizaje cooperativo es un enfoque pedagógico en el que los estudiantes trabajan en pequeños grupos para alcanzar objetivos comunes. Este enfoque promueve la interdependencia positiva, la responsabilidad individual y grupal, y el desarrollo de habilidades interpersonales y de grupo. Investigadores como David y Roger Johnson han sido pioneros en el desarrollo y la promoción del aprendizaje cooperativo.

Para implementar efectivamente el aprendizaje cooperativo, los educadores deben diseñar actividades y tareas que requieran colaboración y que no puedan ser completadas individualmente. Además, es importante estructurar los grupos de manera que haya una diversidad de habilidades y que cada miembro tenga un rol claro y significativo.

Los frameworks mencionados son aplicables a lo largo de todos los niveles educativos, y pueden ser tomados como punto inicial para frameworks más específicos dentro de la educación.

2.7.4. Taxonomía de Bloom

La taxonomía de Bloom es un marco teórico desarrollado en 1956 por el psicólogo educativo Benjamin Bloom [D.R., 1956] y un grupo de colaboradores, con el propósito de proporcionar una clasificación sistemática de los objetivos educativos y facilitar el diseño curricular, la enseñanza y la evaluación del aprendizaje. Este marco se estructura en una jerarquía de niveles de complejidad cognitiva, organizados en una secuencia ascendente de habilidades.

2.7.5. Taxonomía de Bloom Original

La taxonomía original de Bloom consta de seis categorías principales, cada una representando un tipo de habilidad cognitiva, desde las más simples hasta las más complejas:

1. **Conocimiento:** Se refiere a la capacidad de recordar datos, hechos, conceptos y principios. Es el nivel más básico y fundamental del aprendizaje.
2. **Comprensión:** Implica la capacidad de interpretar, traducir y extrapolar información. En este nivel, los estudiantes demuestran que entienden el material al poder explicarlo con sus propias palabras.
3. **Aplicación:** Consiste en usar la información aprendida en nuevas situaciones concretas. Los estudiantes aplican conocimientos, habilidades y reglas en contextos prácticos.
4. **Análisis:** Involucra descomponer la información en sus partes componentes y entender su estructura. Los estudiantes examinan relaciones, identifican motivos o causas, y diferencian entre hechos y opiniones.
5. **Síntesis:** Implica combinar elementos para formar una estructura coherente y original. Los estudiantes crean nuevos productos o puntos de vista integrando diferentes ideas.
6. **Evaluación:** Se refiere a la capacidad de emitir juicios basados en criterios y estándares. Los estudiantes valoran la validez de ideas, métodos o materiales.

2.7.6. Revisión de la Taxonomía

En 2001, Lorin Anderson, un antiguo estudiante de Bloom, y David Krathwohl publicaron una revisión de la taxonomía, que introdujo cambios significativos en la estructura original. Las categorías fueron renombradas y reorganizadas en una nueva jerarquía:

1. **Recordar:** (anteriormente Conocimiento) Implica recuperar información relevante de la memoria a largo plazo.
2. **Entender:** (anteriormente Comprensión) En este nivel, los estudiantes construyen significado a partir de mensajes educativos, incluyendo la interpretación, ejemplificación, clasificación, resumen, inferencia, comparación y explicación.
3. **Aplicar:** (sin cambios) Consiste en usar procedimientos y conceptos en situaciones nuevas.
4. **Analizar:** (sin cambios) Involucra descomponer información en partes componentes y entender cómo esas partes están relacionadas entre sí y con un todo estructural.

5. **Evaluar:** (anteriormente Síntesis) Involucra emitir juicios basados en criterios y estándares, lo que incluye la crítica y valoración de ideas y métodos.
6. **Crear:** (anteriormente Evaluación) Este es el nivel más alto de la taxonomía revisada y se refiere a la capacidad de generar nuevos productos, ideas o formas de entender combinando elementos de maneras novedosas.

2.7.7. Zona de desarrollo próximo (ZPD)

La zona de desarrollo próximo es la brecha entre lo que un aprendiz ya ha dominado (nivel actual de desarrollo) y lo que puede lograr cuando se le proporciona apoyo educativo (desarrollo potencial). Es el nivel de desarrollo de un niño que se manifiesta en la actividad colaborativa con un adulto o una pareja que tiene mayor conocimiento, pero no en la actividad individual del niño.

El concepto fue introducido por Lev Vygotsky para abordar dos problemas de la psicología del desarrollo y la educación: (1) cómo evaluar correctamente las habilidades intelectuales de los niños y (2) cómo evaluar la eficacia de las prácticas educativas.[Vygotsky, 1978] Este concepto sugiere que el aprendizaje es más efectivo cuando las tareas están en el nivel de dificultad adecuado, es decir, justo fuera del alcance de las capacidades actuales del individuo pero accesibles con la asistencia adecuada.

Dentro de la teoría formulada por Vygotsky se mencionan diversos conceptos clave para entender la ZPD y su aplicación; a continuación se describen algunos de estos:

- **Capacidad Actual y Potencial:** La ZPD define la brecha entre lo que una persona puede hacer sin ayuda (su nivel de desarrollo actual) y lo que puede lograr con la orientación adecuada (su nivel de desarrollo potencial).
- **Andamiaje:** El apoyo proporcionado por el mentor o guía durante el aprendizaje en la ZPD se conoce como andamiaje. Este apoyo se retira gradualmente a medida que el aprendiz se vuelve más competente y capaz de realizar la tarea de manera independiente.
- **Carga Cognitiva:** La carga cognitiva se refiere a la cantidad de esfuerzo mental requerido para procesar y comprender la información durante el aprendizaje.

En conclusión, lo que busca la ZPD es adaptar la educación de modo que el estudiante pueda aprender a través de trabajo colaborativo, mediante actividades que vayan más allá de su ZPD, con el objetivo de que, al terminirlas y habiendo adquirido el conocimiento necesario, más tarde pueda realizarlas por su cuenta.

Este concepto resulta clave para la creación de actividades que busquen potenciar el aprendizaje de los alumnos y desarrollar ciertas habilidades para resolver problemas. En este concepto también se destaca la presencia de una visión constructivista y un enfoque asociado al trabajo cooperativo.

2.8. Frameworks en la educación del pensamiento computacional

En la literatura también se encuentran propuestas de frameworks para la enseñanza del CT; algunos toman como base frameworks más generales, como el constructivismo, por ejemplo, y aplican conceptos del CT abarcando diferentes aspectos de este, como las diferentes habilidades asociadas, conceptos de programación y el nivel educacional al que apuntan.

A continuación, se describen algunos de estos frameworks:

2.8.1. The Three Dimensional Framework

En 2012, Brennan y Resnick realizaron una propuesta de framework debido a la falta de consenso en cómo acercar el pensamiento computacional en edades tempranas. Luego de diversos estudios, desarrollaron el denominado "Three Dimensional Framework", bajo el contexto de Scratch, un entorno de programación que permite a los jóvenes crear sus propias historias interactivas, juegos y simulaciones, y luego compartir esas creaciones en una comunidad en línea con otros jóvenes programadores de todo el mundo.

El marco 3D de Brennan y Resnick está diseñado para evaluar y fomentar el desarrollo del pensamiento computacional en los estudiantes. Este marco se compone de tres dimensiones principales: Conceptos Computacionales, Prácticas Computacionales y Perspectivas Computacionales. A continuación, se detalla cada una de estas dimensiones y su uso en la educación.

- **Conceptos Computacionales:** Esta dimensión se centra en los conceptos fundamentales que los estudiantes deben comprender para poder programar y resolver problemas computacionales. Los conceptos clave incluyen: Secuencias, Bucles (Iteración), Condicionales, Operadores y Datos.
- **Prácticas Computacionales:** Esta dimensión describe las actividades y procesos que los estudiantes desarrollan y aplican al trabajar en proyectos de programación. Las prácticas clave incluyen: Experimentación y Depuración, Reutilización y Modificación, y Abstracción y Modularización.
- **Perspectivas Computacionales:** Esta dimensión se refiere a cómo el pensamiento computacional cambia la forma en que los estudiantes ven y entienden el mundo. Incluye: Expresión Personal, Conexión con Otros y Reflexión.

Para la evaluación de este framework, Brennan y Resnick proponen tres acercamientos: Análisis del portafolio de proyectos en Scratch, Entrevistas basadas en artefactos y Escenarios de diseño. En la siguiente tabla se presenta cómo cada acercamiento ayuda a la evaluación de cada dimensión del framework 1.

Tabla 1: Fuerzas y limitaciones de los acercamientos para evaluación.

Fuente: Brennan y Resnick, AERA 2012.

	Conceptos	Prácticas	Perspectivas
Acercamiento 1: Análisis de Proyectos	Presencia de bloques indica encuentros conceptuales	N/A	N/A
Acercamiento 2: Entrevistas basadas en artefactos	Orlando Ayala, EE.UU.	Sí, basado en diseños propios y auténticos de experiencias, pero sujeto a limitaciones de memoria	Quizás, pero difícil de preguntar directamente
Acercamiento 3: Escenarios de diseño	Matices y variedad de comprensión conceptual, pero seleccionando proyectos externos	Sí, en tiempo real y en una situación novedosa, pero seleccionando proyectos externos	Quizás, pero difícil de preguntar directamente

2.8.2. The Three Dimensional Framework Redeveloped for ECE

En el año 2023, Weipeng Yang et al. [Weipeng Yang, 2023] realizaron un estudio para mejorar el 3D framework y adaptarlo para su uso en edades tempranas. Para ello, hicieron una investigación sobre la literatura del CT en edades tempranas, encontrando así los conceptos clave y las modificaciones necesarias al framework para su implementación.

Las adaptaciones son descritas a continuación:

- **Conceptos Computacionales:** En lugar de enseñar conceptos avanzados como "buclesz condicionales"de manera abstracta, se presentan en forma de juegos y actividades físicas. Por ejemplo, se puede utilizar un juego de "seguir las instrucciones"donde los niños deben seguir una secuencia de pasos (como levantarse, girar, sentarse) para aprender sobre secuencias y flujo de control.

Los niños aprenden las diferencias básicas entre hardware y software a través de actividades concretas, como identificar partes físicas de una computadora (hardware) y aplicaciones que utilizan (software). Una actividad podría involucrar un juego de emparejar tarjetas con imágenes de hardware (teclado, ratón) y software (programas, aplicaciones).

- **Prácticas Computacionales:** Los niños participan en la creación de algoritmos simples mediante actividades prácticas, como la construcción de torres de bloques siguiendo una secuencia de colores o tamaños predefinidos. Esto les ayuda a entender cómo los pasos específicos conducen a un resultado final.

Los niños se involucran en actividades como clasificar objetos por forma o color, lo que les ayuda a identificar patrones. Por ejemplo, se les puede dar una serie de bloques y pedirles que los organicen siguiendo un patrón específico (rojo, azul, rojo, azul).

Se introducen conceptos de depuración a través de juegos donde los niños deben encontrar errores en una secuencia de instrucciones. Por ejemplo, un maestro podría intencionalmente cometer un error al seguir una receta simple de cocina, y los niños deben identificar y corregir el error.

- **Perspectivas Computacionales:** Se alienta a los niños a expresar sus ideas y crear proyectos utilizando herramientas simples de programación. Por ejemplo, pueden usar aplicaciones de programación visual como ScratchJr para crear historias animadas, permitiéndoles ver cómo sus ideas se pueden convertir en proyectos tangibles.

Los niños conectan el CT con su vida diaria mediante actividades que integran CT con otras áreas del aprendizaje. Por ejemplo, se pueden utilizar juegos de construcción donde los niños deben planificar y construir una estructura, conectando conceptos de diseño y construcción con CT.

Los niños desarrollan la perseverancia al resolver problemas a través de actividades que requieren ensayo y error. Por ejemplo, en un juego de puzzles, se les anima a intentar diferentes soluciones hasta encontrar la correcta, reforzando la idea de que el fracaso es una parte natural del proceso de aprendizaje.

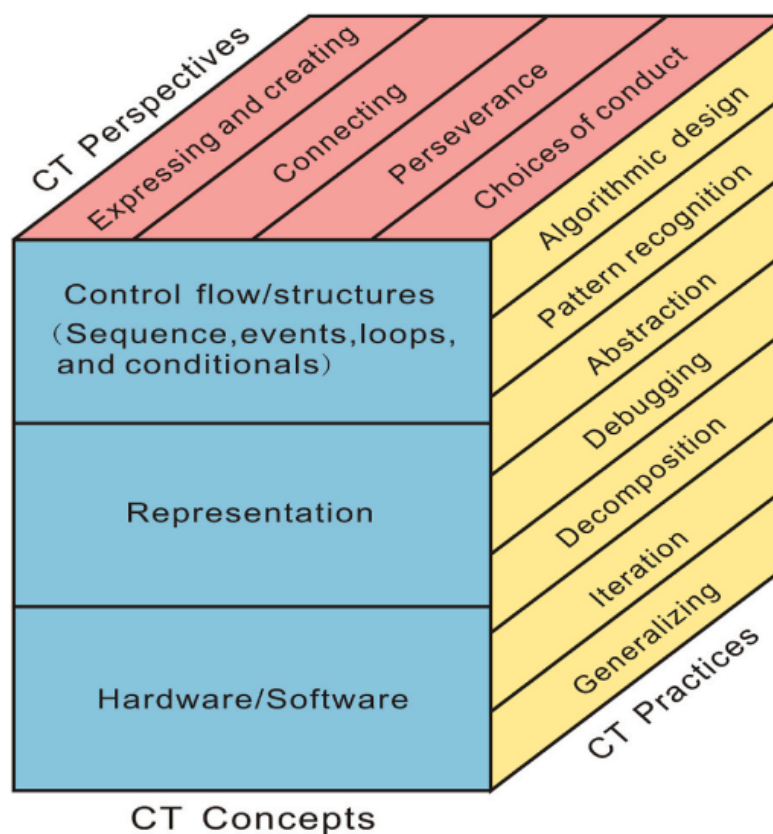


Figura 4: CT curriculum framework for early childhood education: A three-dimensional model.

Fuente: Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework .

El rediseño del 3D framework para su implementación en edades tempranas utiliza estrategias desenchufadas y enchufadas por igual, sin embargo, al estar únicamente basado en el 3D framework propuesto por Brennan y Resnick puede tener ciertos sesgos y omitir estrategias o puntos clave que otros framework puedan incluir.

2.8.3. The CT-cube

En el año 2022, Piatti et al. [Piatti, 2022] desarrollaron un framework que ofrece un enfoque integral para el diseño y la evaluación de actividades de pensamiento computacional. El marco está diseñado para abordar las necesidades de desarrollo de los alumnos en varias etapas y garantizar una enseñanza y aprendizaje efectivos de las habilidades de CT. Se propone que la enseñanza del CT debe contemplar desde la edad del alumno hasta el contexto de aprendizaje, para así determinar la mejor forma de enseñanza.

El CT-cube es una matriz tridimensional de $3 \times 3 \times 3$ que puede utilizarse, en combinación con cualquier modelo de pensamiento computacional, para el diseño, la realización, el análisis y la evaluación del pensamiento computacional. A continuación, se describen las tres dimensiones del framework:

- **Actividad:** El tipo de actividad que se está realizando o que se requiere (planteamiento del problema, algoritmo, evaluación).
- **Entorno:** El mundo (computacional) en el que se llevan a cabo las actividades, caracterizado por el entorno de artefactos que se está utilizando (personificado, simbólico, formal).
- **Autonomía:** La autonomía del individuo con respecto a los demás individuos en el sistema cognitivo situado (rol inactivo o pasivo, rol de apoyo activo o actividad individual con apoyo, rol de liderazgo activo o actividad individual autónoma).

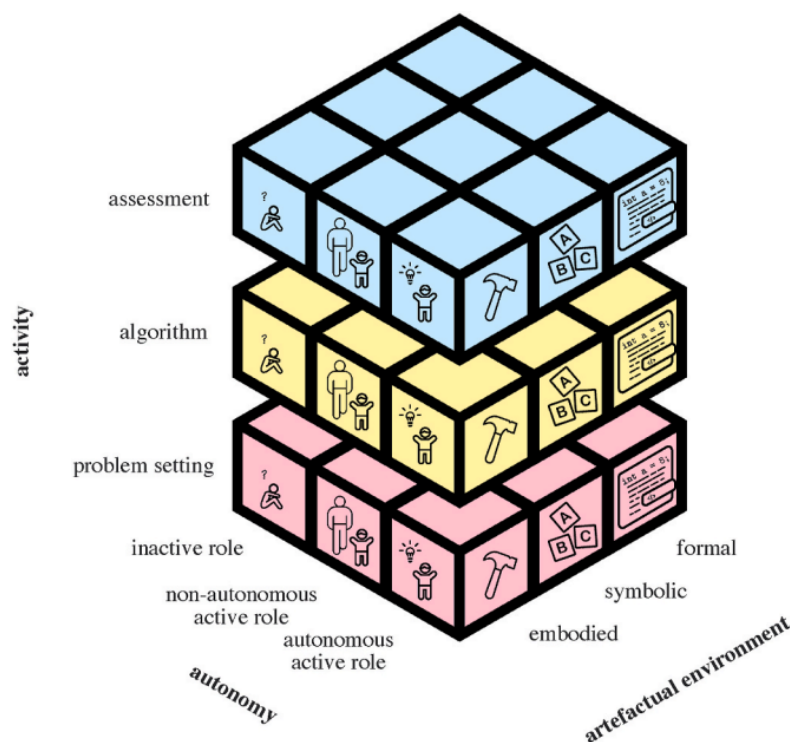


Figura 5: The CT-cube

Fuente: The CT-cube: A framework for the design and the assessment of computational thinking activities.

El framework definido permite diseñar, realizar y analizar actividades para el desarrollo del CT, así como también permite evaluar las competencias de un alumno en la dimensión seleccionada y el desarrollo a futuro de una dimensión.

Es importante destacar que el CT-cube no es un modelo de pensamiento computacional independiente, sino una extensión que puede aplicarse a cualquier modelo de pensamiento computacional, con el fin de tener en cuenta explícitamente el carácter evolutivo y la naturaleza situada de las actividades y habilidades de pensamiento computacional.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En base a la investigación realizada en los capítulos anteriores, es posible comenzar a plantear una propuesta de un framework basado en estrategias unplugged con un enfoque en edades tempranas, para promover el desarrollo de las habilidades asociadas al pensamiento computacional definidas con anterioridad. Los pasos a seguir para el diseño y validación del framework son los siguientes:

- **Selección de estrategias unplugged apropiadas:** En base a la investigación realizada y un análisis de diferentes fuentes, se seleccionan las estrategias unplugged con mejores resultados y con mayor aplicación dentro de la enseñanza del pensamiento computacional; luego se define su potencial para aplicación en aula.
- **Análisis de frameworks existentes:** En capítulos anteriores se presentaron diversos frameworks ya existentes para la enseñanza del pensamiento computacional; se analizan estos para encontrar las características más importantes y puntos que sirvan de ayuda para el diseño del framework.
- **Diseño de Framework en base a las estrategias seleccionadas:** Tomando las estrategias seleccionadas y los puntos más importantes de frameworks existentes, así como también añadiendo características destacadas encontradas durante la investigación, se presenta el diseño de un framework para la enseñanza de habilidades del PC en edades tempranas en base a estrategias unplugged.
- **Diseño de pauta de buenas prácticas para cada estrategia:** Para cada estrategia seleccionada, se diseñará una pauta de buenas prácticas donde se incluyan: actividades de ejemplo, qué habilidades se pueden trabajar de mejor manera, cómo implementarlas en el aula y cómo aumentar o disminuir la dificultad de las actividades.
- **Caso de uso:** Para la validación del framework se realizará un caso de uso, donde se ponga en práctica el framework y se evalúe el aprendizaje con su uso.
- **Validación del framework en base a resultados:** En base a los resultados obtenidos, se evaluará el funcionamiento del framework, se analizarán posibles mejoras y se realizarán conclusiones en base a la investigación realizada.

3.1. Selección de estrategias unplugged

Las estrategias elegidas fueron brevemente descritas en la sección anterior; dentro de estas se encuentran algunas que están claramente definidas en la literatura y otras que fueron definidas a partir de la agrupación de actividades que compartían características similares y que fueron mencionadas en diversos estudios e investigaciones.

3.1.1. Criterios de selección:

1. **Presencia en la literatura:** El primer criterio es efectivamente que la estrategia esté presente en la literatura asociada al CT en edades tempranas, ya sea mediante su referencia directa o agrupando actividades que estén presentes en la literatura que compartan características similares y que se hayan usado para validar la efectividad de las actividades unplugged en la enseñanza del CT.
2. **Diversidad de actividades:** Este criterio se refiere a la cantidad de actividades posibles de diseñar con la estrategia elegida. Se eligieron las estrategias que tienen un gran potencial en cuanto a diseño de actividades diferentes; esto también está asociado con la cantidad de habilidades que se pueden trabajar con la estrategia en cuestión, por ende se seleccionan las que permitan trabajar más habilidades.
3. **Variabilidad de dificultad:** Se seleccionan las estrategias donde se puedan diseñar actividades con distintos tipos de dificultades de forma simple. Para ello, se analizan las actividades existentes y se verifica que existan formas de aumentar la dificultad o disminuirla. Se valora, entonces, la variabilidad en cuanto a dificultad que puedan tener las actividades diseñadas.

A continuación, tomando en cuenta la investigación realizada y diversos estudios analizados, se presentan las estrategias elegidas, junto con los motivos de su elección:

1. **Metáforas:** Dentro de las estrategias elegidas, esta se presenta como la más general. En el estudio de [Manches, 2020] se presenta este tipo de estrategia usando metáforas personificadas para la enseñanza de conceptos fundamentales del CT, y se analiza también su efectividad. La razón de la elección de este tipo de estrategias es su presencia en diversas fuentes y su amplia aplicación debido a la generalidad de su definición. Usar metáforas para la enseñanza de conceptos y para el desarrollo de actividades resulta en una mejor comprensión inicial de conceptos, así como también en una mayor motivación por parte de los alumnos para participar en actividades. Esto es especialmente relevante en edades tempranas, donde mantener el foco del alumno en la clase resulta un desafío para los educadores. Usando metáforas divertidas, como por ejemplo: algoritmos representados como un camino para llegar a cierto lugar, o identificar cosas en común entre objetos conocidos para identificar patrones, se puede lograr

trabajar las habilidades asociadas al CT de una forma que el alumno se mantenga motivado y le sea más fácil entender el objetivo de las actividades.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción.

Otras aplicaciones: Introducción de conceptos.

2. **Graph Paper Programming:** Esta estrategia también se presenta en diversos artículos y destaca su presencia en trabajos recientes como el de [Javier del Olmo Muñoz, 2020] o también en el libro CS Unplugged. Esta estrategia se asocia a actividades que requieren lápiz y papel, donde normalmente el recurso que se elige para el diseño de actividades es el dibujo, ya sea dibujando en base a instrucciones, dibujando rutas en base a algoritmos, introduciendo comandos para completar dibujos de cierta forma previamente definida, etc. El motivo de elección de esta estrategia es que desarrolla de muy buena manera las habilidades asociadas al CT, sobre todo el pensamiento algorítmico, que es al que se apunta en la mayoría de actividades encontradas en la literatura. Además, también resulta en una mayor motivación por parte de los alumnos y permite el diseño de actividades en grupos o parejas, aportando al desarrollo del trabajo en equipo.

Habilidades aplicables: Reconocimiento de patrones y pensamiento algorítmico.

3. **Storytelling:** Esta estrategia, si bien se encuentra presente en algunos artículos asociados al CT, es un tipo de estrategia mayormente usada en otras áreas, por ejemplo, estudios humanísticos. Esto es evidente si se analiza la naturaleza del tipo de actividad resultante del storytelling; normalmente se utiliza para el análisis de historias en literatura, aportando a la comprensión lectora y análisis crítico. Sin embargo, su aplicación en el desarrollo de las habilidades presentes en el CT presenta un gran potencial. Esto se logra variando el tipo de historia que se cuenta y la actividad que se le presenta al alumno. De esta forma, esta estrategia se elige debido a que, usando estos factores, es posible promover el desarrollo de estas habilidades contando historias que presenten patrones donde el alumno deba encontrarlos, o presentando una historia por partes y donde el alumno deba unir de forma correcta estas partes.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Desarrollo conjunto con la lectura y/o escritura, Introducción de conceptos.

4. **Juegos de clasificación y ordenación:** En la literatura estudiada, muchas veces se encuentran actividades asociadas al pensamiento algorítmico que, mediante juegos, buscan enseñar ciertos algoritmos de clasificación y/o ordenación. Un ejemplo de esto es a través de juegos colaborativos para ordenar por tamaño ciertos objetos. En este caso, se podría presentar al alumno, de forma indirecta, instrucciones para que realicen bubblesort y comparar realizando el mismo proceso con instrucciones que lleven a otro tipo de algoritmo. De esta forma, el alumno aprende distintos tipos de algoritmos sin directamente aprender la teoría de estos, notando cuáles son más efectivos que otros. La definición y elección de esta estrategia radica en su poder para la enseñanza

de algoritmos, donde la dificultad de la actividad depende del algoritmo a enseñar, y a medida que se presentan algoritmos más complejos, el alumno desarrolla aún más la habilidad del pensamiento algorítmico.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Aprendizaje de algoritmos específicos.

5. **Actividades de role-playing:** Esta estrategia hace uso de la personificación para actividades donde el trabajo colaborativo forma parte importante. Los alumnos son capaces de trabajar la abstracción tomando roles donde, por ejemplo, un alumno sea un robot y otro alumno haga el rol de programador dándole instrucciones. También es posible simular procesos donde cada alumno represente un componente de un computador que forme parte de este proceso. De esta forma, dependiendo de la actividad que se diseñe, es posible trabajar todas las habilidades de forma integral, motivando a los alumnos a trabajar en equipo; por estos motivos, es que se elige esta estrategia.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Introducción de conceptos, trabajo en equipo.

6. **Juegos de estrategia y Resolución de Problemas:** Esta estrategia hace alusión al uso de juegos de estrategia para el desarrollo de las habilidades del CT. No se define en particular algún juego mejor que otro; sin embargo, algunos juegos son más difíciles que otros. Un ejemplo de esto sería el tic-tac-toe, o el gato como se le llama en Chile, que, si se compara al ajedrez, la diferencia de dificultad es muy grande. De todas maneras, con ambos se puede trabajar el pensamiento estratégico de los alumnos. Es aquí donde se encuentra el motivo de la elección de esta estrategia, ya que, dependiendo del juego elegido, es posible trabajar el reconocimiento de patrones, pensamiento algorítmico y la abstracción. El ejemplo más claro sería el ajedrez, donde los patrones se identifican en el movimiento de las piezas, así como también en posiciones del tablero que se pueden repetir entre juegos. De esta forma, eligiendo diferentes juegos, es posible trabajar las habilidades de los alumnos, ya sea dejándolos jugar libremente o diseñando actividades que integren estos juegos, enseñando estrategias para ganar en forma de algoritmos, etc.

Habilidades aplicables: Reconocimiento de patrones, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Pensamiento crítico.

7. **Simulación de programación:** La simulación de programación comprende la agrupación de actividades donde se presenten problemas que puedan ser resueltos usando comandos predefinidos entregados al estudiante. Estos pueden ser a través de tarjetas o cualquier recurso que represente un comando. De esta forma, es posible simular un proceso de programación para llegar a una solución al problema propuesto. Debido a esto, se elige este tipo de estrategia, ya que, dependiendo del problema a resolver,

los comandos propuestos y cómo se representen estos, es posible variar la dificultad de los ejercicios y trabajar integralmente todas las habilidades asociadas al CT.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Introducción de conceptos, Introducción a la programación.

8. **Actividades Kinestésicas:** Este tipo de estrategia se basa en la implementación de actividades asociadas a la actividad física. La elección de esta estrategia nace a partir de la versatilidad del tipo de actividades que se pueden diseñar. Es posible, por ejemplo, enseñar conceptos y desarrollar habilidades asociadas al CT usando actividades como el aprendizaje de coreografías, donde se trabaja la descomposición, reconocimiento de patrones y algoritmos. Así como también es posible desarrollar la abstracción. De esta forma, entonces, además de potenciar el desarrollo de las habilidades que se busca trabajar, se promueve la actividad física y la vida sana en los estudiantes, además de generar trabajo en grupo dependiendo de la actividad realizada.

Habilidades aplicables: Reconocimiento de patrones, Descomposición y Pensamiento algorítmico.

Otras aplicaciones: Desarrollo de actividad física, trabajo en equipo.

9. **Estrategias Mixtas:** Si bien es cierto que la investigación realizada y este trabajo en particular están enfocados en el uso de estrategias unplugged, cabe destacar que es posible integrar estrategias mixtas en el aula. Esto puede permitir una mayor variedad de actividades, así como también potenciar el desarrollo de todas las habilidades asociadas al CT. En el estudio de Javier del Olmo [Javier del Olmo Muñoz, 2020] se presentan ventajas de usar estrategias mixtas, donde se presentan actividades que comienzan como unplugged y, a medida que se avanza en dificultad, se añaden herramientas plugged-in para que los alumnos desarrollen las actividades. De esta forma, se pueden introducir conceptos con actividades unplugged y profundizarlos usando herramientas como Scratch JR. Debido a este potencial de profundización, es que se elige este tipo de estrategia.

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Otras aplicaciones: Introducción de conceptos, Aplicación de herramientas de programación.

Como es posible apreciar se hace una distinción entre lo que es una estrategia y una actividad, siendo en este caso una estrategia una forma para planificar la enseñanza o el desarrollo de las habilidades del pensamiento computacional, y una actividad vendría a ser el resultado de aplicar una estrategia en particular para formular un ejercicio que permita al alumno practicar y desarrollar estas habilidades.

Dentro de la sección del diseño de la pauta de buenas practicas se presentarán ejemplos de actividades para cada una de las estrategias, además de profundizar en que habilidades aplican de mejor manera.

3.2. Análisis de frameworks existentes

En la sección anterior se presentaron 3 frameworks ya existentes asociados directamente a la enseñanza del CT: el 3D framework, el 3D framework redevelop for ECE y el CT-cube, siendo estos frameworks relativamente nuevos, el primero de 2012, el segundo de 2023 y el tercero de 2022, por ende, su diseño toma en cuenta gran parte de la literatura presente hasta la actualidad.

Dentro de estas propuestas se encuentran diferentes características en común. En primer lugar, las tres propuestas abarcan tres dimensiones sobre las cuales se trabaja la enseñanza del CT. Esto permite profundizar en el aprendizaje de los alumnos, considerando no solo los conceptos enseñados y su dificultad asociada, sino que se aborda la enseñanza desde diferentes perspectivas que se deben tomar en cuenta para proveer una mejor educación.

En el caso del 3D framework y el 3D framework redevelop for ECE, las dimensiones son las siguientes: Conceptos Computacionales, Prácticas Computacionales y Perspectivas Computacionales. Es importante destacar para esta investigación que en Prácticas Computacionales estos frameworks abordan indirectamente el desarrollo de las habilidades del CT a través de actividades prácticas, en este caso particular haciendo uso de Scratch Jr como base. Otro punto a destacar es que en la dimensión de Perspectivas Computacionales se aborda cómo los estudiantes cambian su forma de ver y entender el mundo a medida que integran el CT en su aprendizaje. Este punto es importante ya que permite al framework ser adaptable también al alumnado en cuestión, no centrándose en su totalidad en los conceptos y lo teórico.

En el caso particular del 3D framework for ECE, se explica la importancia de añadir las perspectivas computacionales en niños, alentándolos a expresar ideas y crear sus propios proyectos, desarrollando así su creatividad e integrando aún más las habilidades del pensamiento computacional a un mayor nivel. Para tal dimensión se mencionan las siguientes perspectivas:

1. Expressing and creating
2. Connecting
3. Perseverance
4. Choices of conduct

El punto dos habla de conectar, en un sentido de trabajo en equipo, donde los niños sean capaces de expresar sus ideas a sus compañeros y trabajar en equipo para llegar a un resultado. En este sentido, toma también en cuenta un enfoque pedagógico de aprendizaje cooperativo, donde, por ejemplo, a través de proyectos o actividades en grupo, los alumnos aprenden y potencian su desarrollo mediante el trabajo en conjunto con sus pares.

Es de destacar también de ambos frameworks que, además del aprendizaje cooperativo presente, estos se construyen a partir de un enfoque basado en competencias, en este caso apuntando a la dimensión de Prácticas Computacionales. También se encuentra presente, en cierta medida, el constructivismo al tomar en cuenta las perspectivas computacionales, donde el alumno aprende activamente interactuando con su entorno.

En el caso del CT-cube, este framework apunta al diseño de actividades, sin diferenciar entre plugged-in o unplugged; por tanto, aplica a ambas. En este framework también se abordan diferentes enfoques. En el caso del constructivismo, este se presenta en la dimensión de Autonomía, donde se valora el nivel de acción que toma el alumno en la actividad desarrollada, donde la mayor dificultad radica en actividades donde el alumno participe activamente de manera autónoma.

El Aprendizaje Basado en Competencias se ve reflejado en la dimensión de Actividad, donde se definen distintos objetivos que se deben lograr al realizar una actividad y las competencias que el alumno debe desarrollar mediante estas. En este punto no se hace alusión a las habilidades asociadas al pensamiento computacional de forma directa, sino que se enfoca en obtener competencias asociadas a la resolución de problemas mediante pensamiento algorítmico. Sin embargo, se abarcan actividades de todo tipo que aportan al desarrollo del resto de habilidades.

Finalmente, para el CT-cube también se asocia un enfoque de aprendizaje cooperativo de forma indirecta, ya que no se menciona explícitamente en las dimensiones, pero sí se hace referencia a que el desarrollo de actividades en grupos ayuda a potenciar el aprendizaje y debe tomarse en cuenta a la hora del diseño de estas.

A partir del análisis realizado, se concluye que los frameworks presentados muestran diferentes enfoques educativos a la vez y también presentan una alta valoración de la percepción del alumnado y de cómo se presenta su participación en las actividades y su aprendizaje. Esto complejiza las dimensiones; sin embargo, permite una mejor aplicación en el aula. También se destaca que se valoran los objetivos que se quieren lograr como una dimensión, ya sea como actividad o como habilidad a desarrollar.

3.3. Diseño de Framework en base a las estrategias seleccionadas

En base al análisis de los frameworks estudiados y a la investigación realizada, se identifican algunos puntos clave para el diseño del framework:

1. Objetivo
2. Enfoque
3. Dimensiones
4. Metodología
5. Evaluación

El objetivo del framework propuesto sería el de desarrollar las habilidades del CT en alumnos de edades tempranas, a través del uso de las estrategias previamente seleccionadas. Para esto, además se debe considerar la introducción de conceptos previos, lo cual puede lograrse también aplicando alguna de las estrategias mencionadas.

Dentro de los enfoques, se tomaría como base el constructivismo, ya que dentro de las estrategias seleccionadas durante las actividades, el alumno presentaría un rol activo. En este sentido, el framework debe tener en cuenta las capacidades del alumno para potenciar su desarrollo y, mediante la aplicación de las estrategias para el diseño de actividades, hacer que el alumno reflexione sobre lo realizado y los conceptos aplicados. Al tener en cuenta las habilidades que se quieren desarrollar, el framework debe presentar también un enfoque basado en competencias, donde las competencias específicas que se requieren obtener son las 4 habilidades asociadas al CT que se definieron en secciones anteriores. Finalmente, también cabe destacar que un criterio de selección de estrategias fue el potencial de diseño de actividades cooperativas, esto con el objetivo de fomentar el trabajo en equipo y poder aplicar la ZPD, donde con conceptos como el andamiaje, el alumno sea capaz de resolver actividades cada vez más complejas.

Tomando en cuenta el objetivo del framework y los enfoques a integrar, se definen 3 dimensiones clave y una dimensión extra que vendría a ser el eje por el que lograría el objetivo:

1. **Habilidad a desarrollar:** Esta dimensión comprende las 4 habilidades principales que se busca desarrollar: Reconocimiento de patrones, descomposición, abstracción y pensamiento algorítmico. El framework busca que el alumno sea capaz de desarrollar todas estas habilidades por igual y, de esta forma, resolver problemas cada vez más complejos haciendo uso de estas habilidades aprendidas. Se abordan las habilidades como una dimensión, ya que al momento de planificar una clase, es posible que el educador requiera trabajar una habilidad en particular por sobre las otras, tomando en cuenta a sus alumnos y la evaluación de sus capacidades.

2. **Nivel de escolaridad:** Comprende los diferentes niveles escolares asociados al rango de edad temprana definido de 4 a 8 años. Por simplicidad, se refiere a los niveles escolares del sistema chileno, desde pre-kinder a tercero básico, y se definen rangos. Estos rangos están basados en los presentados para ejercicios en el libro y la documentación de CS Unplugged, los cuales se traducen en lo siguiente: pre-kinder a kinder (4 a 5 años), primero a segundo básico (6 a 7 años) y tercero básico (8 años).
3. **Objetivo de aprendizaje:** Dimensión basada en la taxonomía de Bloom. Se simplifica esta taxonomía en 3 objetivos generales, con el objetivo de darles otro propósito a las actividades creadas, añadiendo dificultad a estas y promoviendo un mayor desarrollo de las habilidades asociadas al CT al llegar a actividades con objetivos como la creación.
4. **Estrategia unplugged a implementar:** Comprende las diferentes estrategias seleccionadas. Cada estrategia posee distintas aplicaciones para distintas habilidades, y su aplicación también varía dependiendo de las dimensiones mencionadas anteriormente. La finalidad es que la estrategia elegida sea un marco para crear actividades que promuevan el desarrollo de una habilidad en particular bajo cierto contexto.

Con las dimensiones definidas anteriormente, es necesario explicar la metodología requerida para el uso del framework. En este caso, este está dirigido a educadores, con la finalidad de que estos puedan diseñar y adaptar actividades directamente en el aula usando el framework como base.

Con las 3 dimensiones base que son Habilidad a desarrollar, Nivel de escolaridad y Objetivo de aprendizaje, el educador tiene el contexto necesario para elegir una estrategia dentro de las Estrategias Unplugged seleccionadas que le permita crear actividades para desarrollar una o más habilidades asociadas al CT. Una vez que el educador haya elegido la estrategia, deberá crear o adaptar una actividad para aplicarla en el aula. Para facilitar esta tarea, se crea una pauta de buenas prácticas para cada estrategia, donde se definen casos de uso y ejemplos, así como también cómo variar la dificultad de las actividades. Esta pauta de buenas prácticas será presentada en las siguientes secciones.

Otro punto importante a la hora de presentar un framework para educación es cómo evaluar si se está cumpliendo el objetivo en los alumnos, que en este caso es desarrollar las habilidades asociadas al CT y aprender conceptos básicos del CT y la programación. Anteriormente se definió el concepto de ZPD (Zona de Desarrollo Próximo). Dentro de las posibles opciones para realizar evaluación de aprendizajes, se propone que el educador evalúe en base a la ZPD de sus alumnos. Es decir, en primer lugar, cuando se dé comienzo al curso donde se aplique el framework, se hagan actividades que permitan al educador establecer un nivel inicial de los alumnos, esto con el objetivo de evaluar la ZPD actual de estos en cuanto a las habilidades a desarrollar. Luego, cada cierta cantidad de clases o actividades, se deberá volver a realizar esto para evaluar la ZPD y ver qué tanta mejora se ha logrado y cómo manejar la dificultad de las siguientes actividades.

A continuación, se presenta una imagen representativa del framework propuesto:

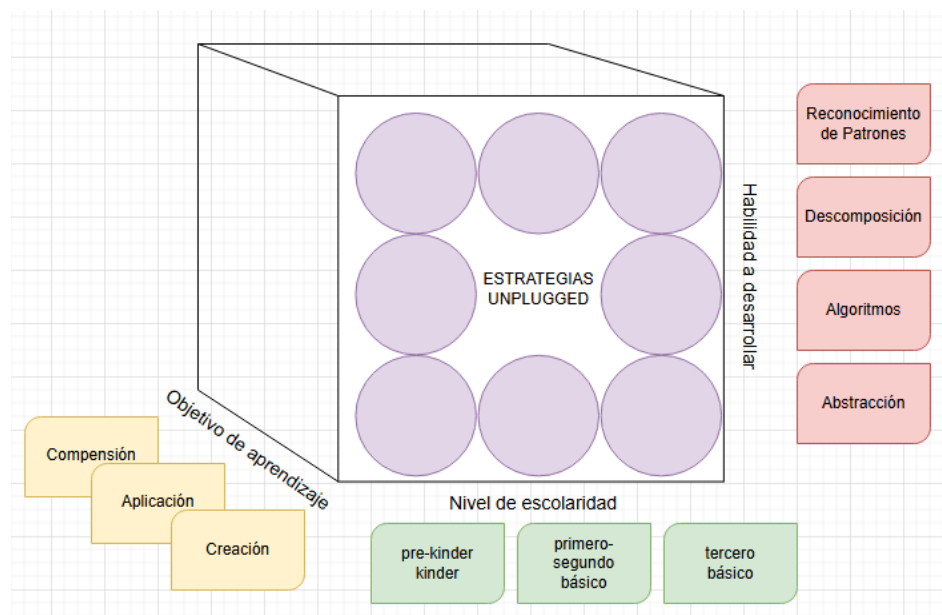


Figura 6: Framework para la enseñanza de las habilidades del pensamiento computacional en base a estrategias unplugged.

Fuente: Elaboración propia.

Como es posible apreciar en la imagen se ven las tres principales dimensiones que forman un cubo y sus respectivas secciones, donde las estrategias unplugged vienen a ser también parte del cubo pero derivadas a partir de la contextualización de las otras tres.

3.4. Diseño de pauta de buenas practicas para cada estrategia

Con el objetivo de facilitar la aplicación del framework en el aula se presentan a continuación pautas de buenas practicas para cada estrategia seleccionada, con ejemplos de actividades, así como también una guía general sobre la aplicación del framework.

Para el diseño de las pautas se tomo en consideración el análisis de los frameworks vistos en secciones anteriores, así como también se analizaron diferentes actividades encontradas en la literatura, sobretodo en el libro CS Unplugged, dentro de este grupo de actividades se eligieron los ejemplos presentados.

3.4.1. Introducción de conceptos y evaluación inicial

Inicialmente se deben introducir los conceptos clave que el alumno debe conocer para el desarrollo de las actividades. Se debe introducir a los alumnos al pensamiento computacional de una forma general y lúdica, y luego hacer una breve introducción de las diferentes habilidades. Para esto se pueden elegir dos maneras: la primera es entregar la información directamente, lo cual es más recomendado para cursos más avanzados. Para alumnos en edades más tempranas se recomienda que la introducción de conceptos sea a través de actividades; en este caso, actividades unplugged simples que permitan al alumno entender de forma básica qué habilidades se están trabajando. Esto, además, permite realizar una evaluación inicial de las capacidades de los alumnos. Es aquí donde se puede realizar un primer análisis de la ZPD de los alumnos a través de actividades que realicen en forma individual y actividades que realicen con ayuda.

Luego de introducir los conceptos y hacer una primera evaluación de la ZPD de los alumnos, es momento de realizar el análisis para saber qué estrategia es más conveniente para el contexto en donde se busca aplicar el framework.

3.4.2. Pauta de buenas prácticas para cada habilidad

1. Metáforas

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción.

Niveles escolares recomendados: Todos los niveles.

Ajuste de dificultad de las actividades: Para aumentar la dificultad de actividades donde se usen metáforas, existen distintas maneras: en primer lugar, se puede recrear la metáfora e incrementar la dificultad del problema o del objetivo de la actividad; otra manera es combinar metáforas donde se introduzcan más conceptos dentro de la misma actividad; otra forma es integrando trabajo en equipo dentro de las actividades; y, finalmente, es posible crear actividades que usen conceptos más complejos, por ende, la actividad sea más compleja.

Ejemplos:

Monica's Map

Un mapa con muchos personajes se muestra a los estudiantes, y ellos tienen que encontrar la ruta más corta entre estos personajes usando solo flechas hacia arriba, abajo, izquierda y derecha ($\rightarrow$, $\leftarrow$, $\uparrow$, y $\downarrow$). En un segundo momento, deben usar multiplicadores (es decir, $\rightarrow \rightarrow \rightarrow \rightarrow \rightarrow = 5x\rightarrow$) para escribir las soluciones.

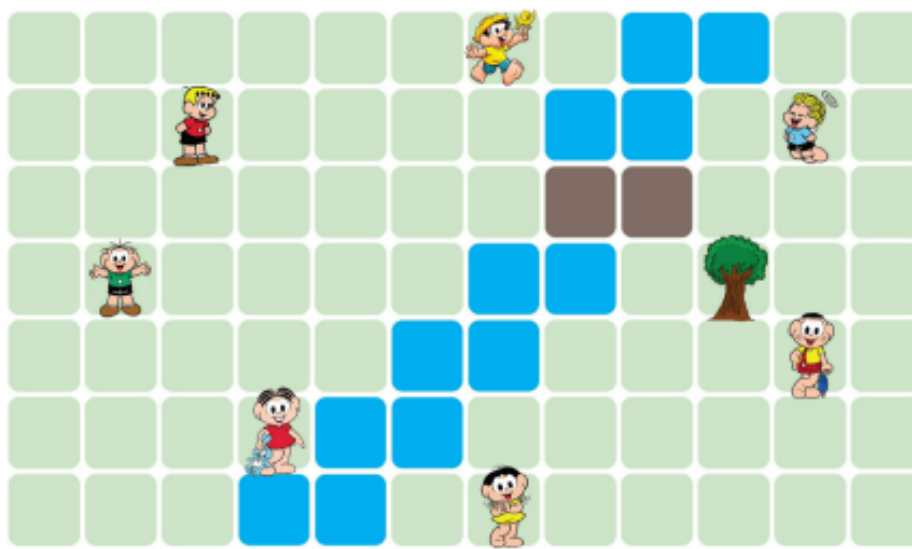


Figura 7: Ejemplo de mapa.

Fuente: Development of Computational Thinking Skills through Unplugged Activities in Primary School.

Metáfora: La metáfora utilizada es el camino que representa el algoritmo para llegar a un destino junto con el contexto de la actividad.

Habilidad principal: Algoritmos.

Objetivo de aprendizaje: Comprensión o aplicación, dependiendo del enfoque de la actividad. Por ejemplo, se puede llevar a la comprensión si se introducen conceptos de algoritmos y patrones mediante la actividad, o aplicación si los conceptos ya fueron enseñados y se presenta un mapa más complejo.

Decomposition activity

Los estudiantes tienen que desglosar muchos problemas (por ejemplo, plantar un árbol), identificando todos los pasos necesarios para resolverlos. Otros ejemplos son: lavarse las manos, preparar el desayuno, tomar un ascensor, atarse los zapatos, etc.

Metáfora: El problema elegido representa un proceso y desglosar los pasos para resolverlo representa la descomposición de ese problema.

Plantar un árbol



1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____

Figura 8: Ejemplo de actividad de descomposición.

Fuente: Development of Computational Thinking Skills through Unplugged Activities in Primary School.

Habilidad principal: Descomposición.

Objetivo de aprendizaje: Comprensión; idealmente, esta actividad se usa sin presentar directamente al estudiante que está realizando descomposición.

2. Storytelling

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Niveles escolares recomendados: Primeros dos niveles (Pre-kinder a segundo básico).

Ajuste de dificultad de las actividades: La variabilidad de la dificultad depende del enfoque de la actividad y de la historia construida. Se pueden utilizar recursos narrativos que hagan más compleja la historia, así como también introducir conceptos más complicados, por ejemplo, añadiendo condicionales que cambien el curso de una historia.

Ejemplos:

El Misterio del Bosque Encantado

Los estudiantes reciben un cuento detallado sobre un bosque encantado donde los personajes encuentran diversos desafíos como ríos, cuevas y criaturas mágicas.

Los alumnos deben descomponer la historia en partes, identificando los eventos principales.

Introducir condicionales en el cuento, por ejemplo, Si los personajes encuentran un río, entonces deben construir un puente para cruzarlo, o Si se encuentran con una criatura mágica, deben resolver un acertijo para continuar.

Habilidad principal: Descomposición.

Objetivo de aprendizaje: Aplicación; se busca que el estudiante entienda que debe descomponer la historia en partes con un sentido en particular, por lo que se busca que apliquen la habilidad de descomposición.

La Ciudad de los Patrones

Los estudiantes ayudan a un detective a resolver misterios en una ciudad donde los eventos ocurren siguiendo patrones específicos.

Describir eventos que ocurren en la ciudad en intervalos regulares, como:

“Cada lunes, hay un desfile en la plaza central.”

“Cada tercer día, aparece un gato en la ventana de la panadería.”

Los estudiantes deben identificar los patrones en los eventos descritos y predecir cuándo y dónde ocurrirá el próximo evento.

Habilidad principal: Reconocimiento de patrones.

Objetivo de aprendizaje: Comprensión o aplicación, dependiendo del enfoque de la actividad; se puede usar como actividad introductoria a conceptos o como actividad donde los alumnos apliquen sus conocimientos ya aprendidos.

3. Juegos de clasificación y ordenación

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Niveles escolares recomendados: segundo y tercer nivel (primero a tercero básico).

Ajuste de dificultad de las actividades: En el caso particular de esta estrategia, es posible variar la dificultad presentando algoritmos más o menos complejos, de ordenación y clasificación.

Ejemplos:

Ordenando Objetos

Ordenar un grupo de 8 envases por su peso, sin conocer inicialmente el peso de cada uno.

Los alumnos deben llenar vasos del mismo tamaño con agua y arena en cantidades diferentes, luego ordenarlos aleatoriamente hasta no recordar cuál es más pesado que el otro.

En primer lugar, se pide ordenarlos del más liviano al más pesado, contando la cantidad de comparaciones que se usan para lograrlo.

Luego, se repite el proceso, pero enseñando un algoritmo de ordenamiento que permita lograr el mismo resultado, pero con menos comparaciones.

Por ejemplo, con un ordenamiento de selección: Uno de los métodos que una computadora puede utilizar se le conoce como ordenamiento por selección. Así es como trabaja: Primero encuentra el envase más liviano en la lista y ponlo en una nueva lista. Luego encuentra el más liviano entre los envases que quedaron y ponlo en la nueva lista. Repite este proceso hasta que hayas quitado todos los envases. Contar las comparaciones realizadas.

Habilidad principal: Reconocimiento de patrones.

Objetivo de aprendizaje: Comprensión o aplicación, dependiendo del enfoque de la actividad; se puede usar como actividad introductoria a conceptos o como actividad donde los alumnos apliquen sus conocimientos ya aprendidos.

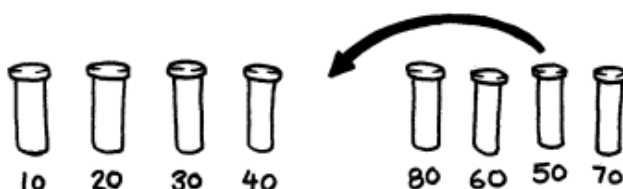


Figura 9: Ilustración visual del algoritmo por selección.

Fuente: CS Unplugged.

4. Actividades de role-playing

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Niveles escolares recomendados: segundo y tercer nivel (primero a tercero básico).

Ajuste de dificultad de las actividades: La complejidad de este tipo de actividades depende de diferentes factores, como, por ejemplo, la cantidad de alumnos que desempeñan un cierto rol, o también el proceso que se quiera representar. El objetivo que se presente a los alumnos también puede variar la dificultad de una actividad, añadiendo restricciones o reglas, o haciendo que los alumnos vayan cambiando de rol.

Ejemplos:

Concurso de Robótica

Los estudiantes se dividen en equipos y se les asigna la tarea de programar a un compañero de equipo para que actúe como un robot.

Cada equipo debe desarrollar un algoritmo que permita al robot realizar una tarea específica (e.g., llevar una bandeja con objetos desde un punto A hasta un punto B, sin caer en trampas).

El robot "solo puede seguir instrucciones simples, como "mueve un paso adelante", "gira a la derecha", "levanta el objeto", lo que obliga a los estudiantes a pensar en secuencias claras y precisas.

Después, los equipos compiten para ver quién creó el algoritmo más eficiente, o pueden intercambiar sus algoritmos para que otro equipo los ejecute.

Habilidad principal: Algoritmos.

Objetivo de aprendizaje: Creación, los alumnos deben ingeniárselas para encontrar el mejor conjunto de instrucciones que lleven al robot al destino final.

5. Juegos de estrategia y Resolución de Problemas

Habilidades aplicables: Reconocimiento de patrones, Abstracción y Pensamiento algorítmico.

Niveles escolares recomendados: Segundo y Tercer nivel.

Ajuste de dificultad de las actividades: La complejidad de este tipo de actividades depende de diferentes factores, como, por ejemplo, la cantidad de alumnos que desempeñan un cierto rol, o también el proceso que se quiera representar. El objetivo que se presente a los alumnos también puede variar la dificultad de una actividad, añadiendo restricciones o reglas, o haciendo que los alumnos vayan cambiando de rol.

Ejemplos:

El Gato

En esta actividad, los estudiantes juegan varias rondas del Gato y observan qué movimientos conducen a una victoria, derrota o empate. La idea es que, a medida que jueguen, los estudiantes identifiquen patrones en las jugadas óptimas para ganar o evitar que el oponente gane.

El objetivo es que los estudiantes comprendan los patrones recurrentes en el juego y sean capaces de aplicarlos en nuevas partidas.

Luego de varias rondas, el profesor puede hacer una pausa y pedir a los estudiantes que expliquen qué patrones han reconocido y cómo los usaron para planificar sus movimientos.

Habilidad principal: Reconocimiento de patrones.

Objetivo de aprendizaje: Comprensión; los alumnos deben identificar y entender qué patrones conducen a una victoria, derrota o empate.

Torre de Hanoi

En este juego, los estudiantes tienen que mover una torre de discos desde un poste a otro, siguiendo ciertas reglas (solo se puede mover un disco a la vez, y no se puede colocar un disco más grande sobre uno más pequeño). Esta actividad invita a los estudiantes a abstraer el problema de mover los discos de manera eficiente y a identificar una solución generalizable.

El objetivo es enseñar a los estudiantes a abstraer el problema, identificando la estrategia clave para resolver la Torre de Hanoi con cualquier número de discos.

Los estudiantes jugarán varias rondas con diferentes cantidades de discos. Posteriormente, tratarán de generalizar un algoritmo que funcione para cualquier cantidad de discos.

Habilidad principal: Abstracción.

Objetivo de aprendizaje: Aplicación; los estudiantes ponen en práctica las reglas del juego y además usan conocimientos previos para encontrar una solución.

6. Simulación de programación

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico.

Niveles escolares recomendados: Todos los niveles.

Ajuste de dificultad de las actividades: La complejidad de este tipo de actividades depende principalmente del proceso que se quiera simular; mientras más pasos y más caminos posibles se tengan, más complicado es encontrar una solución.

Ejemplos:

Simulación de cajero automático

Los estudiantes simularán el funcionamiento de un cajero automático utilizando comandos como ingresar PIN, retirar dinero, consultar saldo y salir. Cada comando se representa con tarjetas o fichas, y los estudiantes deben secuenciar estos comandos para cumplir con las tareas solicitadas.

El docente introduce el problema del cajero automático y los comandos disponibles.

Los estudiantes eligen una tarea (por ejemplo, retirar dinero) y deben ordenar las tarjetas con los comandos necesarios para completar la tarea.

Simulan el proceso colocando las tarjetas en el orden correcto para cumplir con el objetivo, siguiendo un diagrama de flujo.

Se puede agregar dinero falso para simular cantidades específicas de billetes.

Habilidad principal: Abstracción y Algoritmos.

Objetivo de aprendizaje: Aplicación; el estudiante aplica sus conocimientos de algoritmos para lograr una secuencia correcta de comandos que logre el objetivo de la tarea.

7. Actividades Kinestésicas

Habilidades aplicables: Reconocimiento de patrones, Descomposición y Pensamiento algorítmico.

Niveles escolares recomendados: Todos los niveles.

Ajuste de dificultad de las actividades: La complejidad de este tipo de actividades se puede ajustar modificando, por ejemplo, la cantidad de instrucciones en una actividad, añadiendo o quitando condiciones u obstáculos y evaluando según tiempo y velocidad de ejecución o finalización.

Ejemplos:

La Cueca

Los estudiantes aprenderán a bailar la cueca mientras desarrollan habilidades de pensamiento computacional como descomposición, reconocimiento de patrones y algoritmos. A través de una actividad práctica, seguirán y crearán secuencias de pasos que simulan un programa, utilizando su propio cuerpo como el "sistema" que ejecuta el algoritmo.

El docente introduce la cueca como un algoritmo de baile, explicando que cada paso del baile (paseo, floreo, zapateo, escobillado) es como un comando que el cuerpo debe seguir en un orden específico. Los estudiantes entenderán que, al igual que una computadora sigue un programa, ellos seguirán una secuencia de movimientos para ejecutar correctamente el baile.

Se descompone la cueca en sus partes más simples: paseo, floreo, zapateo, escobillado. Cada paso se enseña de manera aislada para que los estudiantes lo practiquen. Luego, el docente explica que, igual que un programa de computadora, estos pasos deben seguir un orden preciso para lograr una cueca completa.

Habilidad principal: Descomposición, Algoritmos y Reconocimiento de patrones.

Objetivo de aprendizaje: Creación y Aplicación; los estudiantes aplican sus conocimientos de pensamiento computacional creando y ejecutando una secuencia de pasos para realizar la cueca.

Se puede proponer un desafío en el que los estudiantes creen su propia versión del baile, modificando pasos y creando su propio algoritmo de danza, presentándolo al resto de sus compañeros.



Figura 10: Ilustración visual de los pasos de la cueca.
Fuente: Diario La Tercera.

8. Estrategias Mixtas

Habilidades aplicables: Reconocimiento de patrones, Descomposición, Abstracción y Pensamiento algorítmico..

Niveles escolares recomendados: Todos los niveles.

Ajuste de dificultad de las actividades: La complejidad de este tipo de actividades depende de diversos factores como el nivel de instrucción, mientras más específicas las instrucciones, más fáciles de realizar será para los alumnos, también del diseño del desafío, al contar con herramientas Plugged-In es posible crear actividades más complejas que luego llevar a código, otro factor es el numero de variables y cantidad de condicionales en la actividad, es decir, mientras mas condiciones se le indiquen al alumno que debe cumplir a la hora de crear su solución más complejo se vuelve el ejercicio.

Ejemplos:

Robots de rescate

Los estudiantes simulan un escenario donde deben programar robots de rescate”para salvar personas en una ciudad afectada por desastres. La actividad comienza con una fase unplugged en la que los estudiantes planifican su estrategia de rescate utilizando

materiales físicos, y luego pasan a una fase plugged-in en la que implementan esa estrategia utilizando una plataforma de programación visual.

Sección Unplugged: Los estudiantes desarrollan una estrategia de rescate utilizando un mapa físico y comandos secuenciados para mover un “robot” a través de un laberinto. Los estudiantes deben planificar los movimientos del robot usando las tarjetas de comandos. Colocan las tarjetas en la secuencia correcta para guiar al robot desde un punto de inicio hasta las víctimas y luego llevarlas a la zona segura.

Sección Plugged-In: Los estudiantes programan el robot de rescate en una plataforma digital para simular el rescate de las víctimas, basándose en la planificación que hicieron en la fase unplugged. Los estudiantes trasladan la secuencia de comandos que planificaron con tarjetas en la fase unplugged a un entorno digital. Utilizan bloques de código (en el caso de Scratch o Blockly) para programar el movimiento del robot virtual, siguiendo la misma lógica que usaron en el tablero físico.

Habilidades Principales: Descomposición, Algoritmos y Abstracción.

Objetivo de aprendizaje: Creación y Aplicación, los estudiantes aplican conceptos de programación y pensamiento computacional tanto en entornos unplugged como plugged-in, desarrollando soluciones creativas y efectivas para rescatar a las víctimas en diferentes escenarios.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Uno de los objetivos planteados inicialmente fue realizar un caso de uso. En este sentido, para validar la propuesta de solución, se presenta un caso de uso en el que a un grupo de educadoras se les capacita en los fundamentos del pensamiento computacional y el marco propuesto. Finalmente, se busca validar la efectividad del marco y la pauta de buenas prácticas a través de una actividad, obteniendo información relevante para el trabajo futuro y posibles mejoras.

Esta capacitación y actividad fueron planteadas a través de un taller voluntario en el Colegio Denham School de San Bernardo, dividido en dos sesiones con un grupo de 6 educadoras en total, de pre-básica y básica.

Cabe destacar que, previamente a las dos sesiones realizadas, se envió a los participantes del taller un video explicativo sobre los fundamentos del pensamiento computacional, definiciones de las habilidades y explicaciones generales, con el fin de ganar tiempo en las sesiones y responder dudas.

4.1. Capacitación

La primera sesión del taller consistió en una capacitación presencial sobre los fundamentos del pensamiento computacional. En esta sesión, hubo asistencia total y se abordaron los siguientes puntos mediante una presentación:

1. **CONTEXTUALIZACIÓN:** Cuál es el escenario actual y qué motiva realizar este taller y propuesta.
2. **OBJETIVOS:** Objetivos finales del taller.
3. **PENSAMIENTO COMPUTACIONAL:** Repaso general de los contenidos.
4. **ACTIVIDADES UNPLUGGED:** Revisión del tipo de actividad propuesto.
5. **MARCO DE TRABAJO:** Revisión de un marco de trabajo propuesto.
6. **PAUTA DE BUENAS PRÁCTICAS:** Presentación de una guía para el educador.

La idea de esta primera sesión era dar contexto al grupo sobre la importancia del pensamiento computacional y sus habilidades asociadas, y explicar conceptos fundamentales para crear actividades que permitan desarrollar esas habilidades. Una vez repasados los fundamentos

se comenzaría con la introducción de las estrategias unplugged y como pueden usarse para diseñar actividades. Finalmente se presentaría el marco de trabajo propuesto junto con una explicación y ejemplos de uso.

Con respecto al último punto, es decir, la pauta de buenas practicas en esta primera sesión solo se mostraría un ejemplo de esta, presentando el formato y explicando como podría usarse.

En esta sesión el grupo se mostró participativo y se realizaron preguntas acerca de conceptos tales como: Algoritmos, Patrones, Programación, Simulación de Programación, Scratch Jr y Objetivos de aprendizaje.

4.2. Actividad Práctica

Para la segunda sesión se presentó una versión resumida de la pauta de buenas prácticas descrita en la sección anterior, donde por cada estrategia se presentaba un solo ejemplo. Esta pauta fue entregada en forma de guía impresa con una copia para cada asistente.

Cabe destacar que debido a problemas de topes de horario, en esta sesión hubo 5 participantes de los 6 iniciales.

Antes de realizar la actividad se realizó una breve lectura de la guía entregada y se resolvieron las dudas acerca de esta.

La actividad propuesta fue la siguiente:

Creación de actividad

Crear o adaptar una actividad en base a las estrategias presentadas dado cierto contexto.

- Elegir un nivel escolar (Pre-Kinder a tercero basico)
- Elegir una habilidad a desarrollar (Descomposición, Reconocimiento de Patrones, Algoritmos o Abstracción)
- Elegir un Objetivo para la actividad (Comprensión, Aplicación o Creación)
- Elegir una Estrategia Unplugged (Alguna de las presentadas anteriormente)
- Describir brevemente una actividad acorde a los criterios seleccionados.

El objetivo de esta actividad era que cada participante fuera capaz de describir una actividad usando los conocimientos entregados en la sesión anterior, aplicando también el marco de trabajo propuesto donde para elegir una estrategia unplugged se debe contextualizar con el

nivel de escolaridad, habilidad a desarrollar y objetivo de aprendizaje. Además se permitía el uso de la guía entregada, donde el objetivo de esto era validar si esta facilitaba el desarrollo de la actividad propuesta.

Finalmente el tiempo entregado para realizar la actividad fue de 25 minutos, aunque el grupo completo terminó antes del tiempo estipulado.

Luego de completar la actividad se solicitó responder a las siguientes preguntas:

- ¿Te fue de utilidad la Guía Práctica?
- ¿Qué cosas quitarías/agregarías a la guía?
- Considerando las sesiones y la guía entregada, ¿En una escala del uno al diez que tan complejo te parece el diseño de este tipo de actividades?
- Dentro de los conceptos estudiados, ¿Hay alguno que te pareció más complicado de entender que el resto?

El objetivo de estas preguntas es realizar una apreciación sobre la calidad del trabajo realizado y la utilidad de la pauta de buenas prácticas, además de obtener información de valor para el trabajo a futuro y mejoras al framework.

Para finalizar la sesión se leyeron en voz alta las respuestas a la actividad y en conjunto se discutieron posibles mejoras a cada respuesta, así como también, cada detalle bien abordado de estas y dudas finales.

4.3. Resultados

A continuación se presentan las respuestas a la actividad y los comentarios obtenidos de cada una.

Respuesta 1

Nivel escolar: Kinder

Habilidad: Reconocimiento de Patrones

Objetivo: Creación

Estrategia: Storytelling

Descripción de la actividad: Luego de escuchar un cuento por la profesora y hablar sobre los patrones que se repitieron en la historia. Se invita a los estudiantes a crear un cuento a través del dibujo, recordando que deben existir patrones en la historia.

Comentarios: En este caso la actividad descrita cumple con las expectativas esperadas, donde se aplicaron los conceptos aprendidos y el framework. Cabe destacar que quizás este tipo de actividad resulte en mejores resultados en niveles escolares mayores, sin embargo, es totalmente aplicable en el nivel escolar elegido tomando en cuenta los matices necesarios.

Respuesta 2

Nivel escolar: Primero Básico

Habilidad: Reconocimiento de Patrones

Objetivo: Aplicación

Estrategia: Juegos de clasificación y ordenación

Descripción de la actividad:

- Cada niño escoge un animal (cintillo) y deberá ponerlo en su cabeza.
- Después se ira solicitando que se junten los animales según se indique
- Salvajes y Domésticos.
- Vivíparos y ovíparos.
- Carnívoros, Omnívoros y herbívoros.
- Según pelaje, etc.

Comentarios: En este caso la actividad descrita cumple con las expectativas esperadas, donde se aplicaron los conceptos aprendidos y el framework. Cabe destacar que quizás este tipo de actividad resulte en mejores resultados en niveles escolares mayores, sin embargo, es totalmente aplicable en el nivel escolar elegido tomando en cuenta los matices necesarios. Se destaca que la aplicación esperada sea para reconocer patrones o similitudes entre animales de la misma especie.

Respuesta 3

Nivel escolar: Pre-Kinder

Habilidad: Reconocimiento de Patrones

Objetivo: Comprensión

Estrategia: Storytelling

Descripción de la actividad: Cuento "La oruga muy hambrienta"

Docente lee en voz alta cuento sobre oruga, en relato se detalla lo que come cada día de la semana ej. El lunes comió 2 manzanas, el martes comió 3 ciruelas... así con todos los días de la semana. Se pueden realizar preguntas, ¿Cuál es el próximo día? ¿Qué comerá?

Luego de comer, oruga duerme y al despertar... ¿Qué pasará? Oruga se convierte en mariposa.

Explorar el patrón al finalizar (Se puede hacer manualidad: Ordenar imágenes).

Comentarios: Para esta actividad la educadora indica que contaba con conocimiento previo del cuento, que lo había oído con anterioridad, en este aspecto se destaca la capacidad de relacionar esto con los conocimientos adquiridos sobre las habilidades del pensamiento computacional. La elección de habilidad, estrategia y objetivo en este caso son correctas en base a lo esperado y la actividad presentada.

Respuesta 4

Nivel escolar: Kinder

Habilidad: Descomposición

Objetivo: Comprensión

Estrategia: Metáforas

Descripción de la actividad: Receta de jugo de naranja (Decomposition activity)

Los niños siguen los pasos de una receta de jugo de naranjas.

Se presentan imágenes en forma de ayuda para identificar los pasos que componen el hacer un jugo de naranja.

Comentarios: Para esta actividad se hizo uso de una referencia presentada en la pauta de buenas prácticas, en particular de la actividad de ejemplo "Decomposition Activity", se realizó una adaptación de esta a una complejidad menor y con un "problema" más cotidiano como lo es hacer un jugo de naranjas. La elección tanto de habilidad como de estrategia y objetivo son correctos con respecto a la actividad planteada, se deja también como posible mejora, el uso de diversas recetas y comparar recetas para explorar la habilidad del reconocimiento de patrones.

Respuesta 5

Nivel escolar: Kinder

Habilidad: Abstracción

Objetivo: Aplicación

Estrategia: Metáforas

Descripción de la actividad: Los niños observan un laberinto y escogen el camino que los lleva a la salida, primero en papel (indicando con flechas) y luego en la sala de clases.

Comentarios: El caso de esta actividad es una variación o adaptación de una actividad de ejemplo presentada en la pauta de buenas practicas, la elección de estrategia y objetivo están correctas para la actividad descrita, sin embargo, la habilidad en este caso a trabajar estaría mas relacionada al trabajo con algoritmos haciendo que los alumnos mediante flechas representen un algoritmo para salir del laberinto y luego puedan aplicarlos y ver sus resultados.

Respuesta 6

Nivel escolar: Pre-Kinder

Habilidad: Descomposición

Objetivo: Aplicación

Estrategia: Juegos de clasificación y ordenación

Descripción de la actividad: Realizar una búsqueda de hojas de árbol en el patio del colegio y clasificarlas según sus tipo/colores/tamaño-rugosidad y luego crear un patrón utilizando dos grupos que seleccionen.

Pueden realizarlo de forma grupal y luego individual.

Comentarios: Esta actividad esta pensada para abordar la descomposición desde el punto de vista de la clasificación de hojas de distinto tipo, donde se tiene un problema que es la agrupación y deben encontrar como dividirlos en distintos grupos, esto además estaría trabajando el reconocimiento de patrones. Como posible mejora se propone introducir tipos de algoritmos sencillos para facilitar la clasificación, donde se podría trabajar el pensamiento algorítmico.

A continuación se presenta una tabla que contiene las respuestas a las preguntas realizadas al final de la sesión:

¿Te fue de utilidad la Guía Práctica?	¿Qué cosas quitarías/agregarías a la guía?	¿En una escala del uno al diez qué tan complejo te parece el diseño de este tipo de actividades?	Dentro de los conceptos estudiados, ¿Hay alguno que te pareció más complicado de entender que el resto?
Sí	Más ejemplos de actividad	1	Metáforas
Si	Quizás más ejemplos	3	Pensamiento algorítmico
Si	Descripción de la habilidad	1	Pensamiento algorítmico
Si	Más ejemplos de actividades	1	Simulación de programación
Si	Descripción de las habilidades previo a los ejemplos y estrategias	8	Simulación de programación / Pensamiento algorítmico

Tabla 2: Tabla resultados preguntas

Con respecto a la Pregunta 1 se observa que un 100 % de los participantes respondió que les fue de utilidad la guía práctica (Pauta de Buenas Practicas).

En cuanto a la Pregunta 2 la respuesta que mas se repite es agregar ejemplos de actividades, esto se puede explicar con que la guía presentada era un resumen de la pauta completa con un ejemplo por cada estrategia.

El promedio de las respuestas obtenidas para la pregunta 3 es de 2.8 lo que indicaría un nivel de complejidad bajo a la hora de diseñar actividades.

Dentro de los conceptos que más generaron dudas representados en las respuestas de la pregunta 5, se destaca el Pensamiento Algorítmico o Algoritmos como habilidad, este concepto fue explicado reiteradas veces durante las sesiones. Resulta interesante este punto ya que estos conceptos pueden estar un poco alejados con respecto a lo que se enseña en este colegio, razón por la cual resultan ser los que más dudas generan.

CAPÍTULO 5

CONCLUSIONES

Para finalizar este trabajo es importante comentar los alcances y aprendizajes obtenidos a partir del diseño del framework y el caso de uso, donde se puso en practica la solución propuesta, así como también aspectos mejorables a futuro con respecto al framework y la pauta de buenas prácticas.

5.1. Alcances y Aprendizajes

Con respecto a los aprendizajes obtenidos a la hora de realizar este trabajo, se destacan los diferentes conceptos asociados en primer lugar al pensamiento computacional y en segundo lugar a la pedagogía en edades tempranas. En cuanto al pensamiento computacional, se realizó una investigación profunda para dar con conceptos clave que facilitarían la puesta en práctica del framework diseñado. En este sentido se destaca la importancia que tienen las habilidades asociadas al pensamiento computacional en el desarrollo de la carrera de Ingeniería Civil en Informática y como estas no solo aplican a la hora de aprender a programar, sino que se pueden llevar a distintas áreas del conocimiento y pueden ser aplicables para resolver distintos tipos de problemas.

Resulta interesante también introducirse en como es posible enseñar estos conceptos y promover el desarrollo de estas habilidades, encontrándose con un abanico inmenso de opciones a la hora de diseñar actividades que tengan como objetivo trabajar estas habilidades en los alumnos, esto es lo que buscaba representar la selección de estrategias unplugged, dar visualización a esta gran cantidad de alternativas, esto sin contar lo que se podría realizar incluyendo actividades plugged-in donde este abanico se amplía aún más.

Estrategias como Storytelling y actividades Kinestésicas que nacen a partir de la investigación de la literatura asociada al pensamiento computacional en edades tempranas, tienen gran potencial para diseñar distintos tipos de actividades y tomando en consideración los artículos realizados a la fecha, es posible profundizar en su estudio y como pueden aplicarse no solo en el desarrollo de estas habilidades del pensamiento computacional sino que también en otras áreas.

Con respecto a las actividades como tal, es imperante destacar el trabajo realizado en el libro CS Unplugged y como este fue de gran ayuda a la hora de seleccionar las estrategias y ejemplos de actividades, así como también para el diseño del framework, tomando en cuenta la cantidad de actividades que provee y su enfoque en edades tempranas. A la hora de diseñar actividades y estudiar la enseñanza del pensamiento computacional en edades tempranas, el libro CS Unplugged resulta ser un elemento clave.

Dados los resultados obtenidos en el caso de uso, en particular con las respuestas de la Pregunta 1 y 3 podemos decir que el framework y la pauta de buenas prácticas resultaron de utilidad para los participantes facilitándoles el trabajo del diseño de actividades y la comprensión de conceptos, se vió también que luego de la capacitación y la entrega de la pauta, el grupo paso de tener conocimientos prácticamente nulos sobre el tema a diseñar actividades simples que permiten el desarrollo de las habilidades del pensamiento computacional, donde las actividades presentadas por los participantes cumplen en su mayoría con lo esperado y son totalmente aplicables en el contexto elegido por cada uno.

Se destaca también que el trabajo realizado cumple con los objetivos planteados en un primer momento donde se realizó la selección de estrategias, se definió una pauta de buenas practicas y se desarrolló un caso de uso, donde además se validó que el framework propuesto fue bien recibido por los participantes, aunque se podría generar un caso de uso a mayor escala con más participantes y ver como es recibido y los resultados de esto.

El pensamiento computacional y el desarrollo de sus habilidades en edades tempranas es un punto relevante en la educación actual, y se espera que este framework y este trabajo permita que en Chile los educadores puedan integrar estos conceptos en la educación y crear actividades, talleres o cursos que no necesiten de dispositivos electrónicos, ni de abundantes recursos para su implementación y que resulten tanto llamativos como provechosos para los alumnos de manera que estos puedan desarrollar estas habilidades de la mejor forma posible.

5.2. Trabajo a futuro y posibles mejoras

Con el framework desarrollado y tomando en consideración los resultados obtenidos en el caso de uso, es posible mencionar algunos aspectos mejorables y en los que se podría trabajar a futuro.

En primer lugar es posible ahondar más en la investigación y ampliar la selección de estrategias, lo que llevaría a una mayor variedad de opciones a la hora de diseñar actividades, esto claramente, tomando en cuenta criterios que permitan que la selección sea adecuada y que las estrategias sean realmente de utilidad.

En segundo lugar están las mejoras a la pauta de buenas practicas, ya que en los resultados obtenidos del caso de uso, los participantes mencionan ciertas cosas que se podrían agregar, por ejemplo, más ejemplos para cada estrategia, ya que de esta forma los educadores mas que crear actividades desde 0 pueden optar por adaptar alguno de los ejemplos.

Luego está el agregar definiciones de conceptos a la pauta, lo cual en un primer momento no se hizo ya que se consideraba una capacitación previa, pero podría ser de utilidad para que los educadores sin necesidad de una capacitación completa sean capaces de entender los conceptos y aplicarlos con actividades, o simplemente para refrescar la memoria, ya que

hay que tomar en cuenta que la mayoría no serian expertos en el tema.

Finalmente a futuro se podrían realizar más casos de uso, en distintos colegios y con mayor cantidad de participantes, con el objetivo de buscar puntos de mejora del framework y también de promover el desarrollo del pensamiento computacional en edades tempranas dentro de la educación en Chile.

ANEXOS

5.1. Imágenes de documentación sobre el caso de uso: Resultados y Guía entregada.

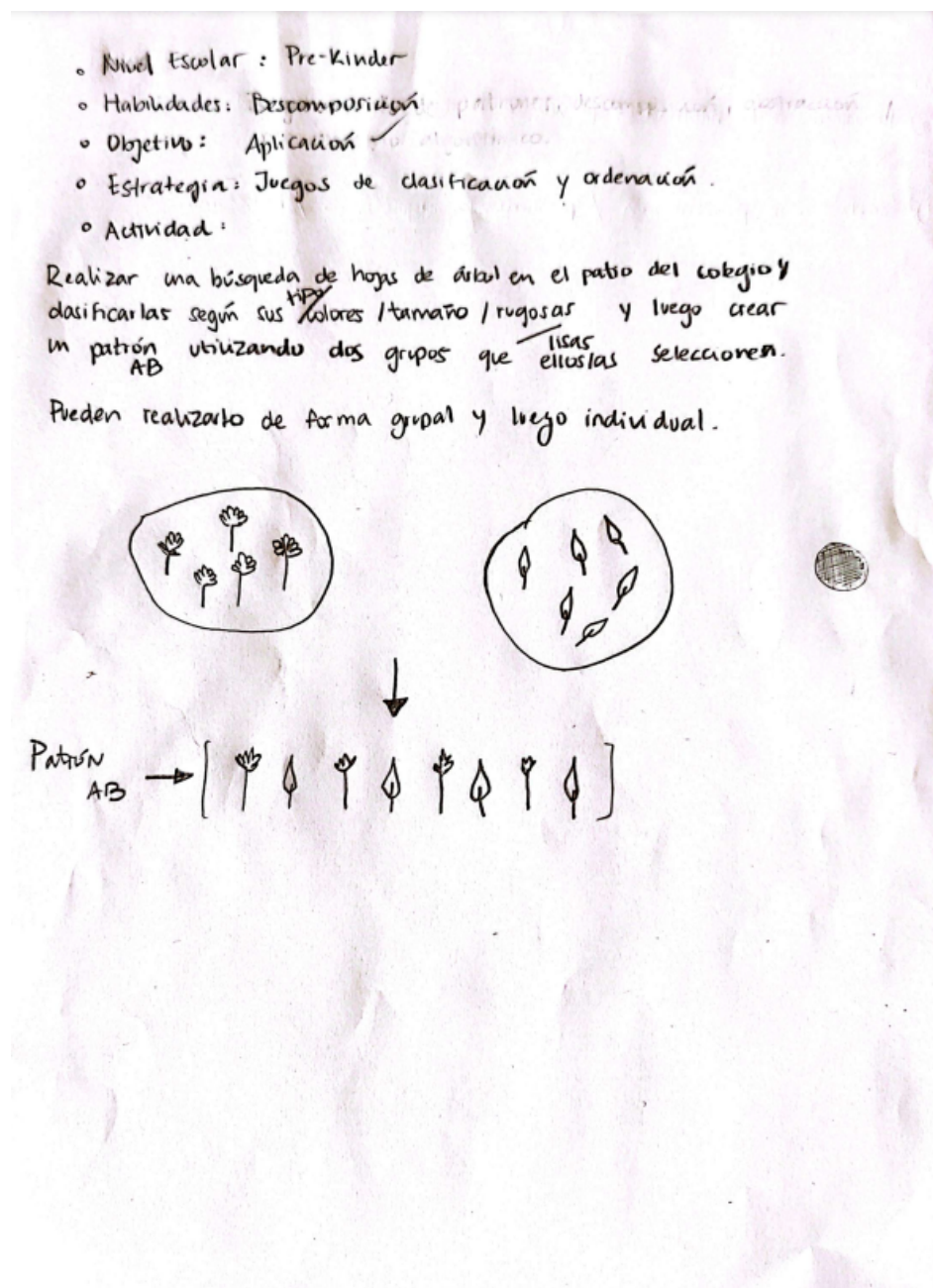


Figura 11: Hoja de Respuestas 1.

Fuente: Taller realizado en validación de solución.



Figura 12: Hoja de Respuestas 2.
Fuente: Taller realizado en validación de solución.

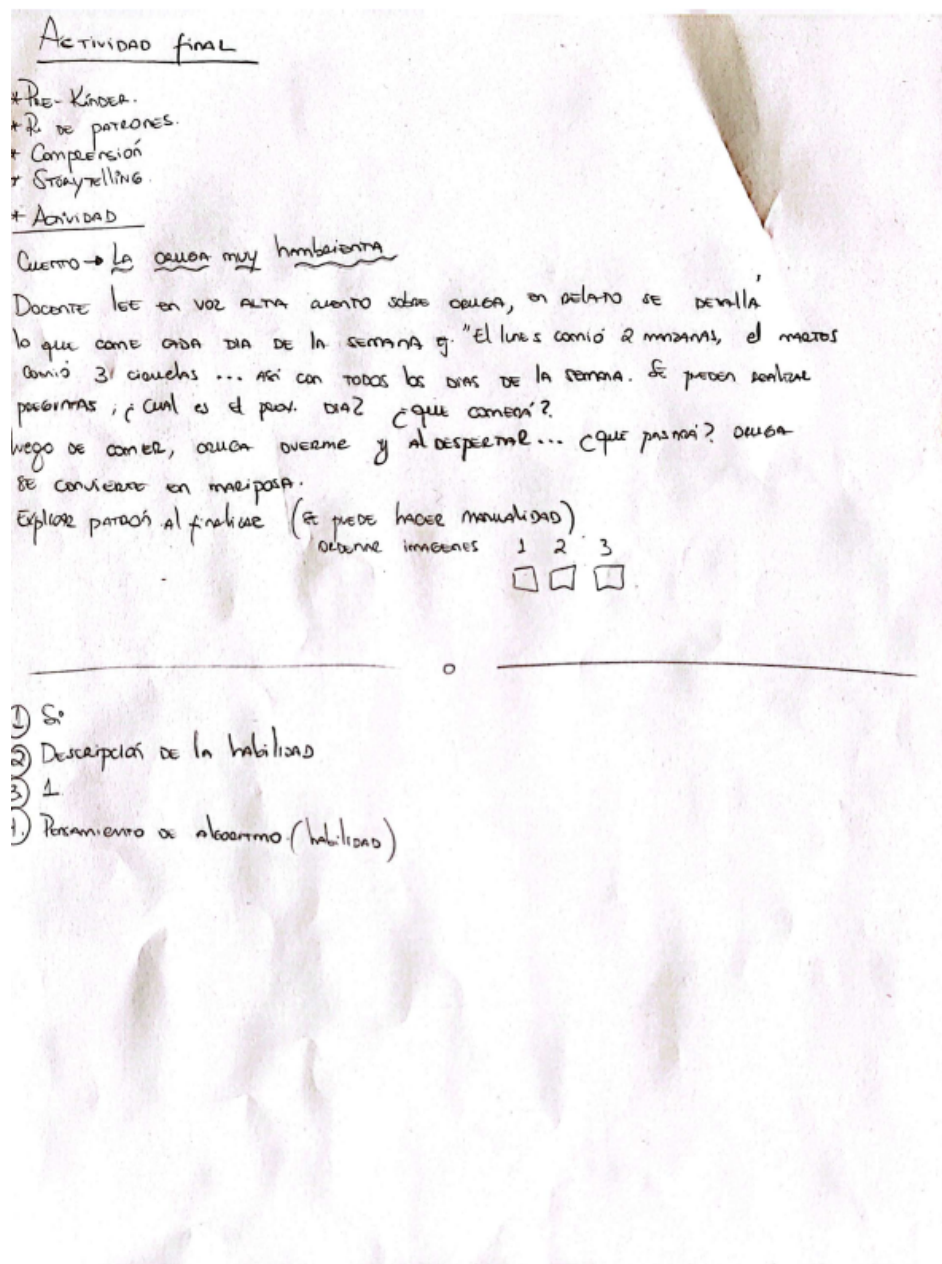


Figura 13: Hoja de Respuestas 3.
Fuente: Taller realizado en validación de solución.

nivel escolar = Primero basico
Habilidad = Reconocimiento de patrones
objetivo = Aplicar conceptos ya aprendidos
Estrategia = juego de clasificación y ordenación
Kinestésica.
★ Cada niño escogera un animal (cintillo) y deberá ponerlo
en su cabeza.
★ Después se ira solicitando que se junten los animales
según las siguientes indicaciones:
★ salvajes y domesticos
★ viviparos y oviparos
★ carnívoros, onívoros y herbívoros
★ según pelaje entre otros.
★

Figura 14: Hoja de Respuestas 4.
Fuente: Taller realizado en validación de solución.

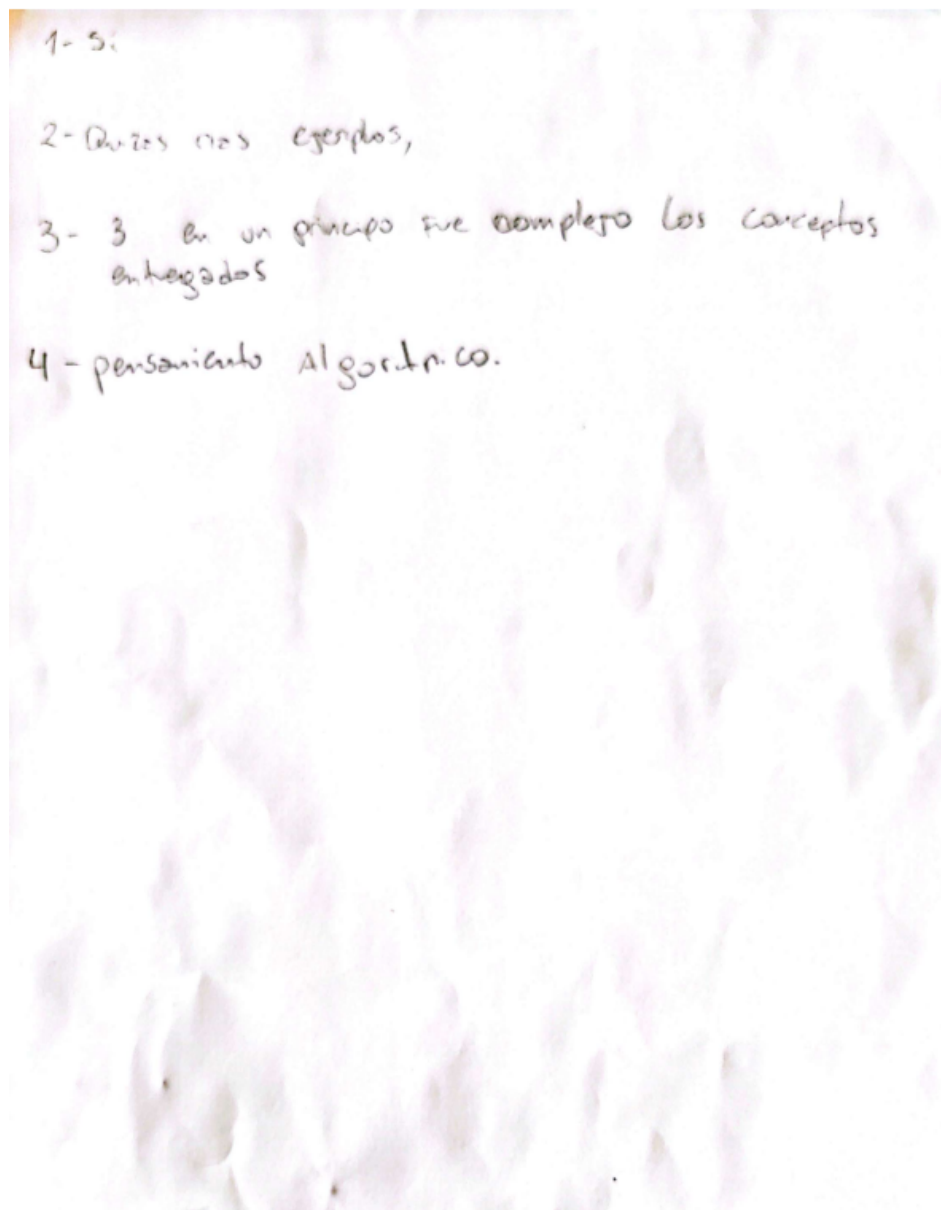


Figura 15: Hoja de Respuestas 5.
Fuente: Taller realizado en validación de solución.

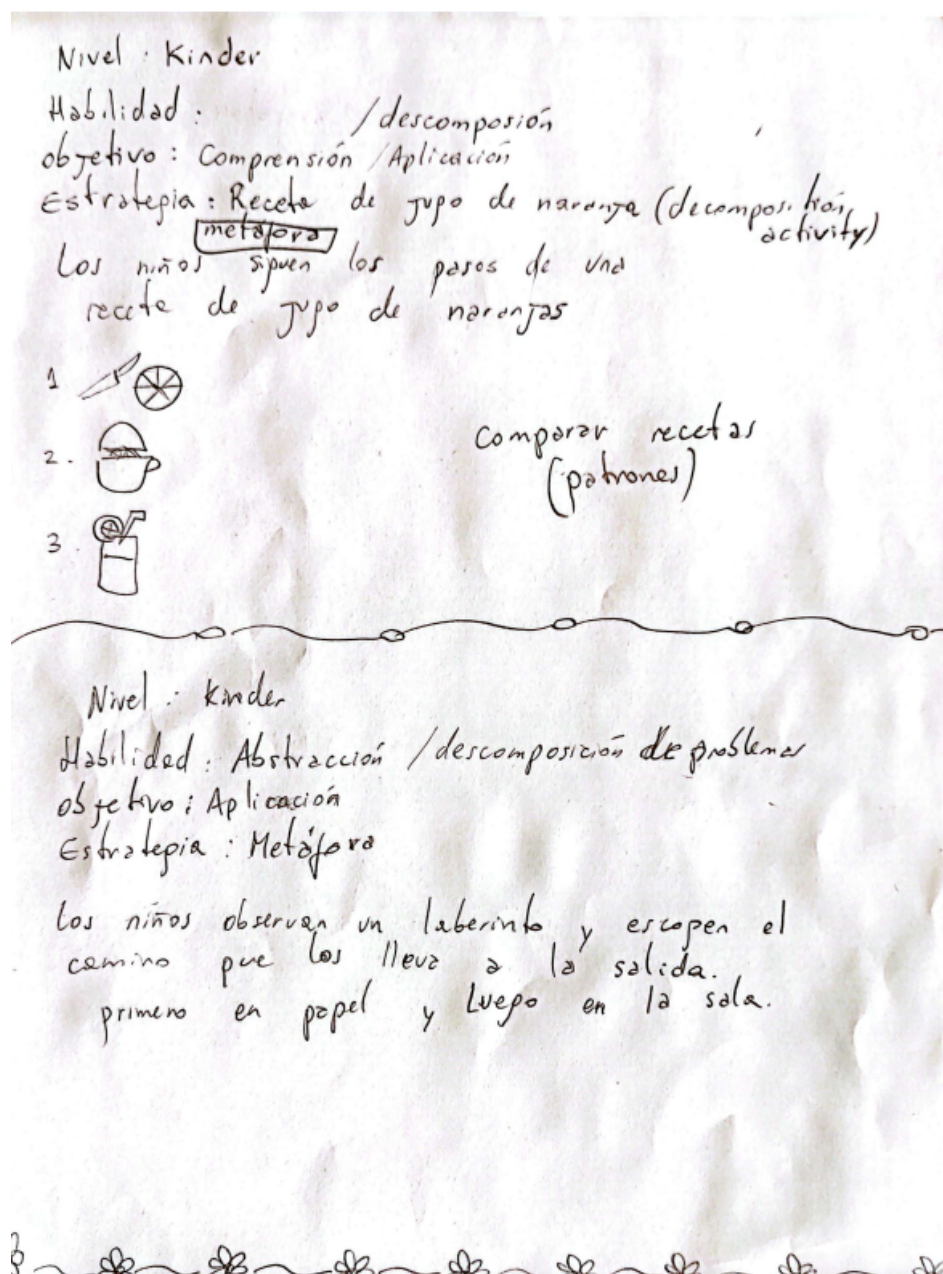


Figura 16: Hoja de Respuestas 6.
Fuente: Taller realizado en validación de solución.

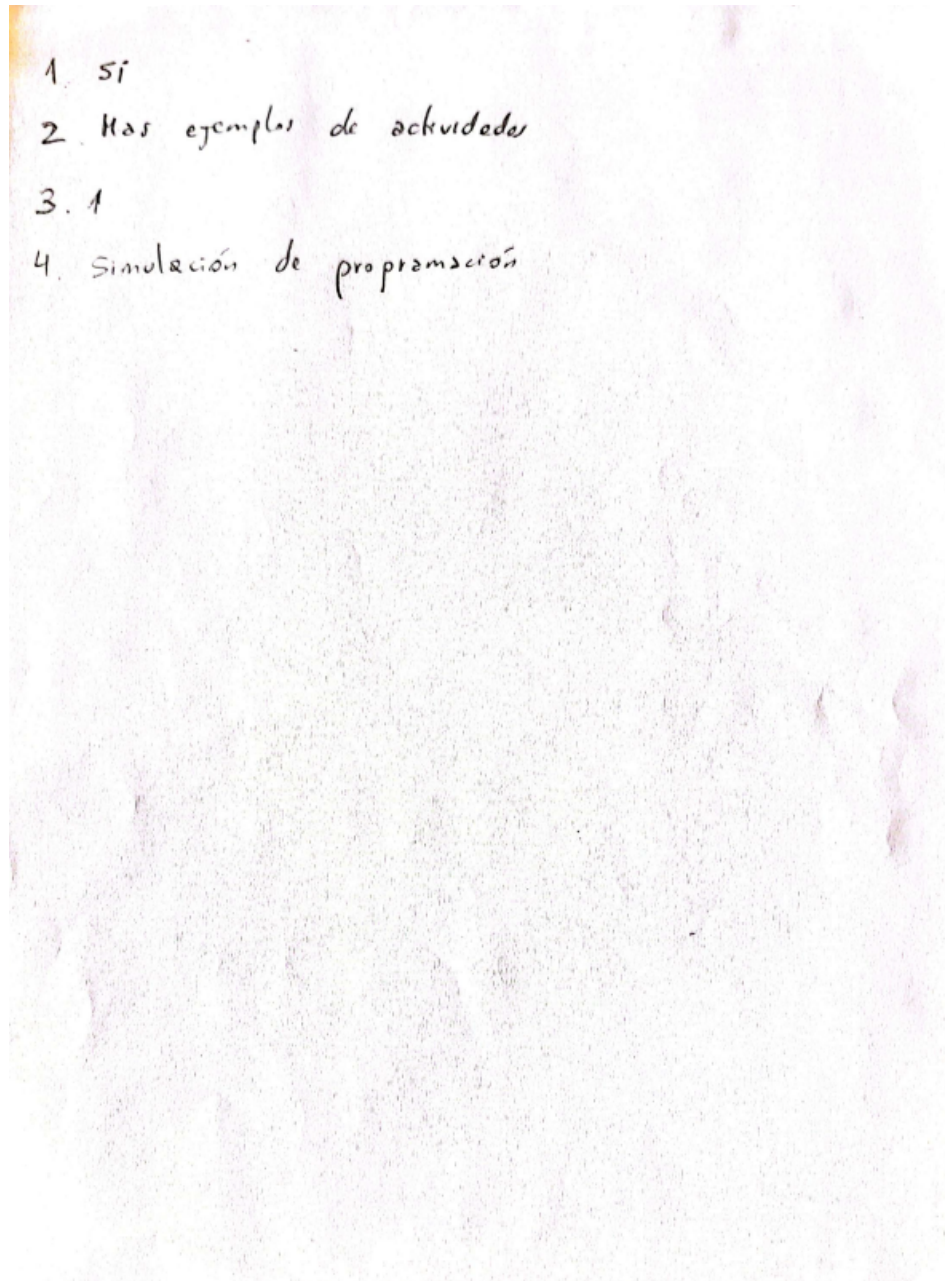


Figura 17: Hoja de Respuestas 7.
Fuente: Taller realizado en validación de solución.

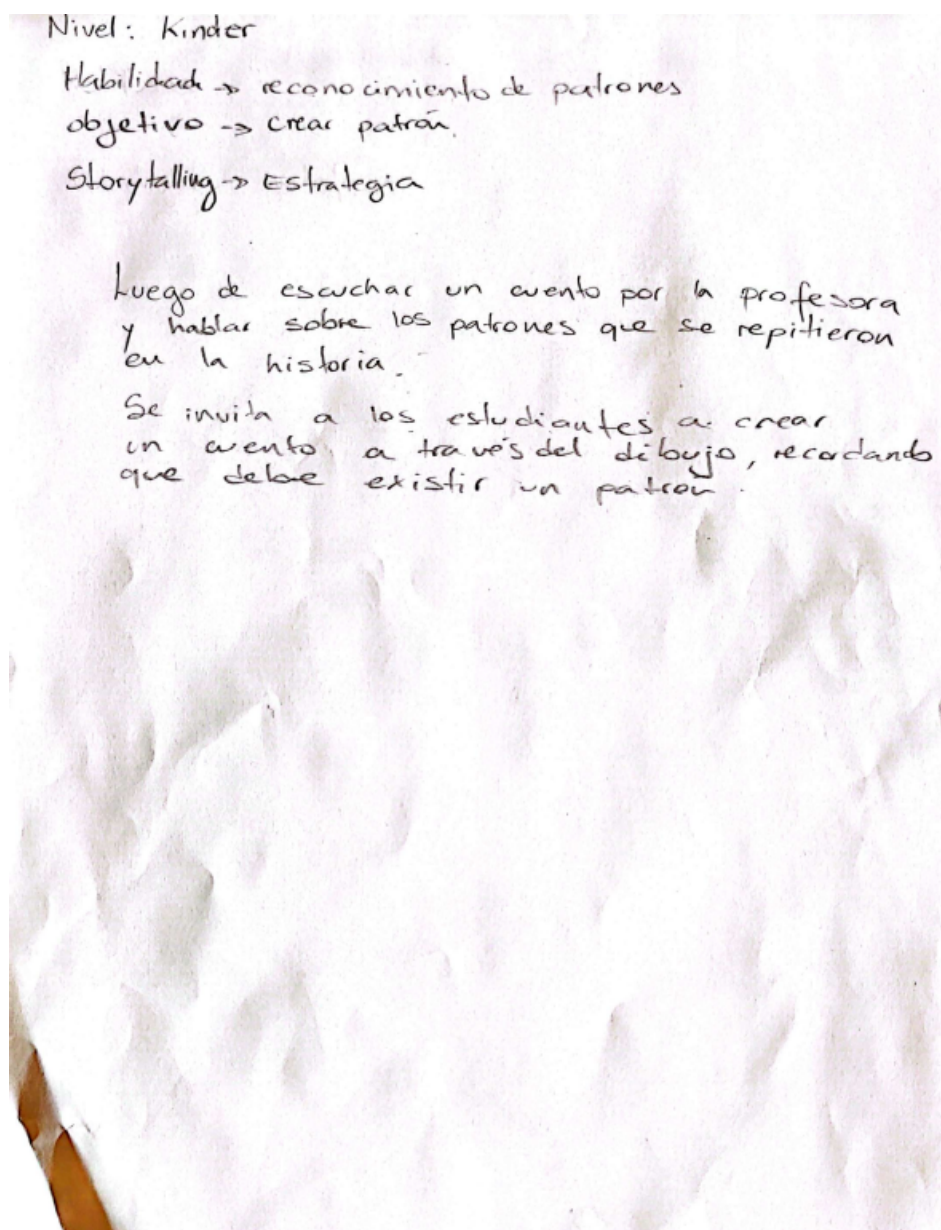


Figura 18: Hoja de Respuestas 8.
Fuente: Taller realizado en validación de solución.

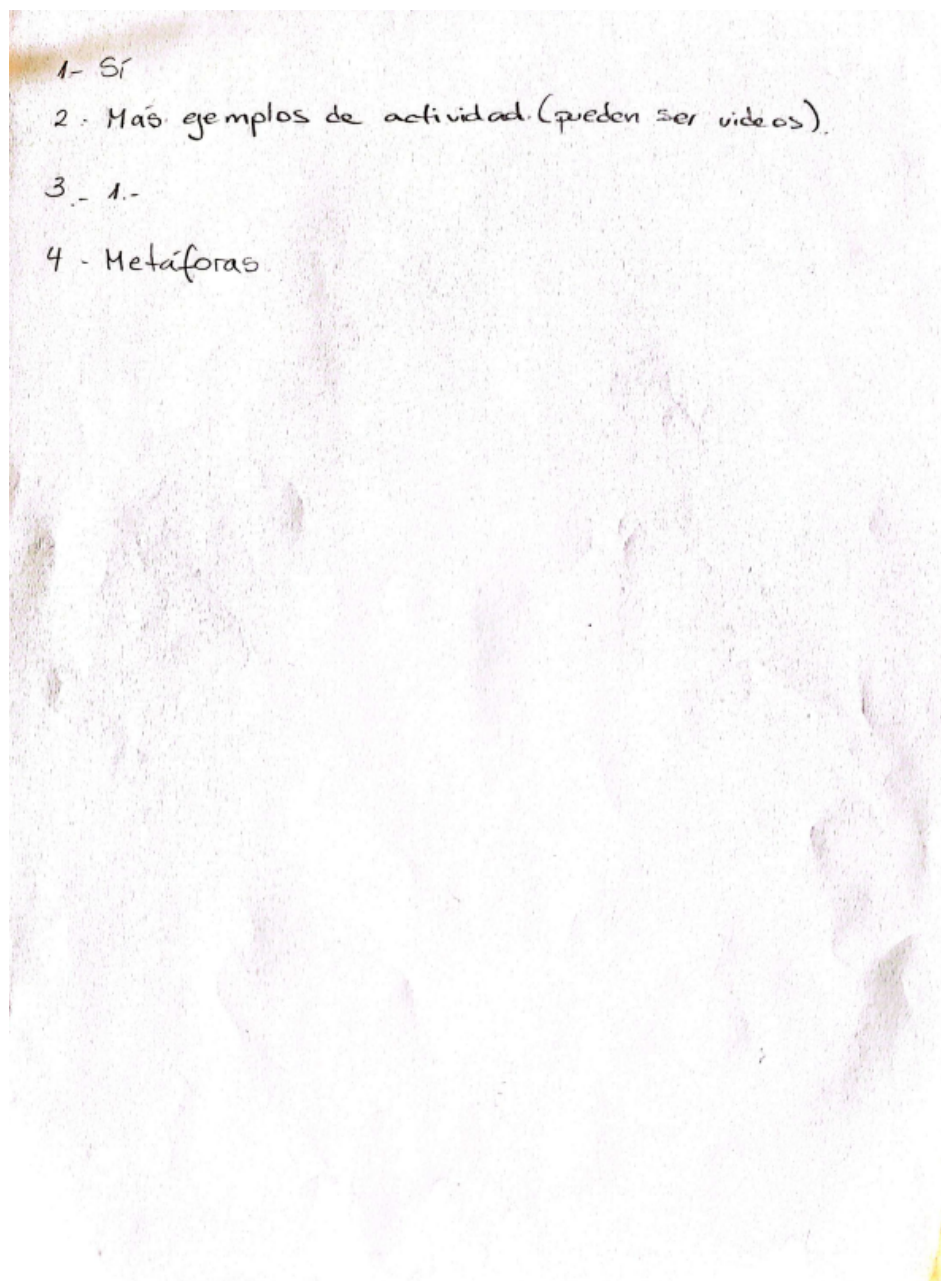


Figura 19: Hoja de Respuestas 9.
Fuente: Taller realizado en validación de solución.

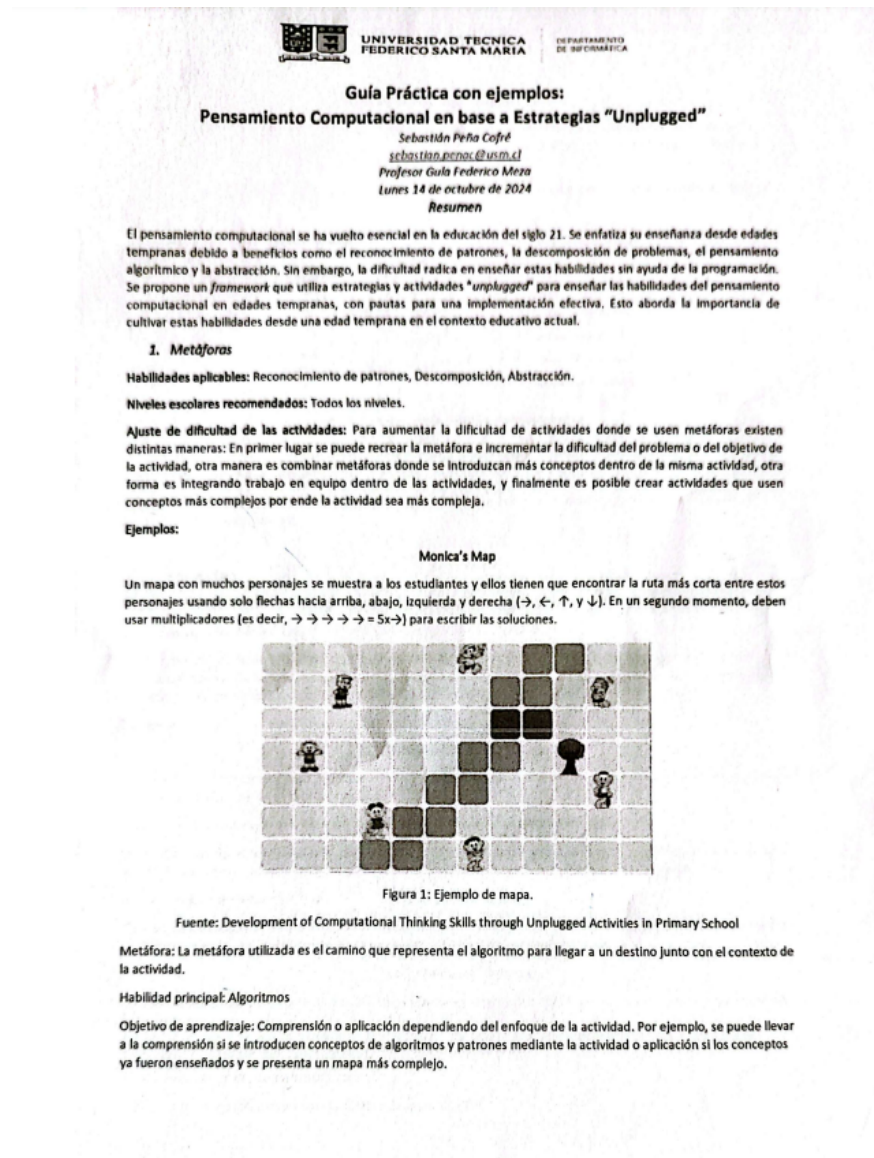


Figura 20: Guía entregada a participantes Hoja 1.
Fuente: Elaboración propia.

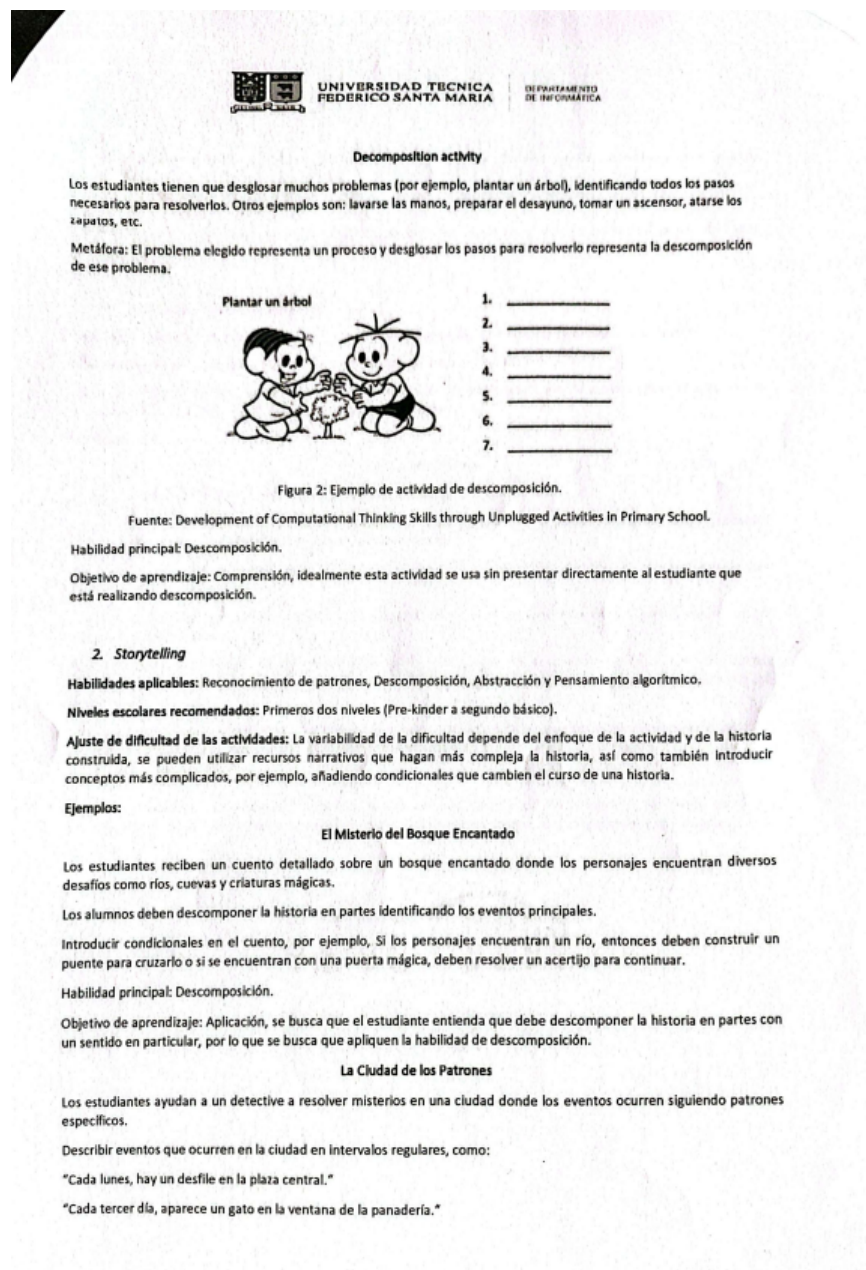


Figura 21: Guía entregada a participantes Hoja 2.
Fuente: Elaboración propia.

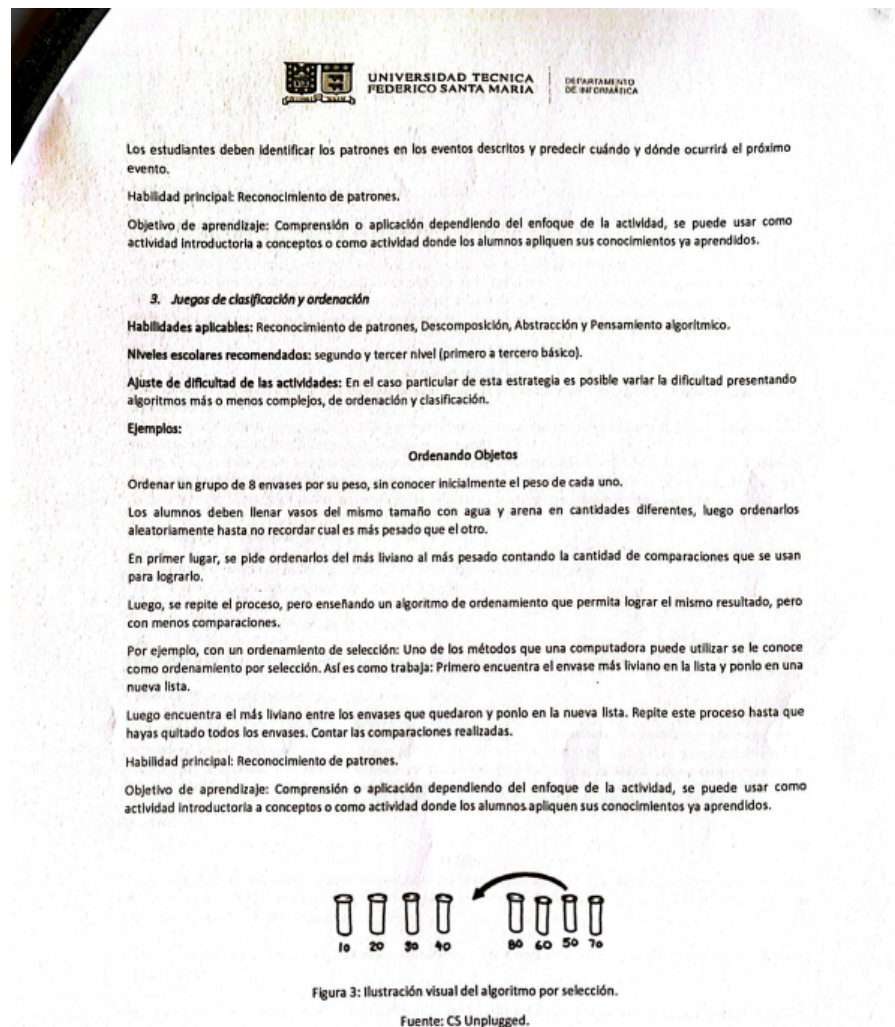


Figura 22: Guía entregada a participantes Hoja 3.
Fuente: Elaboración propia.

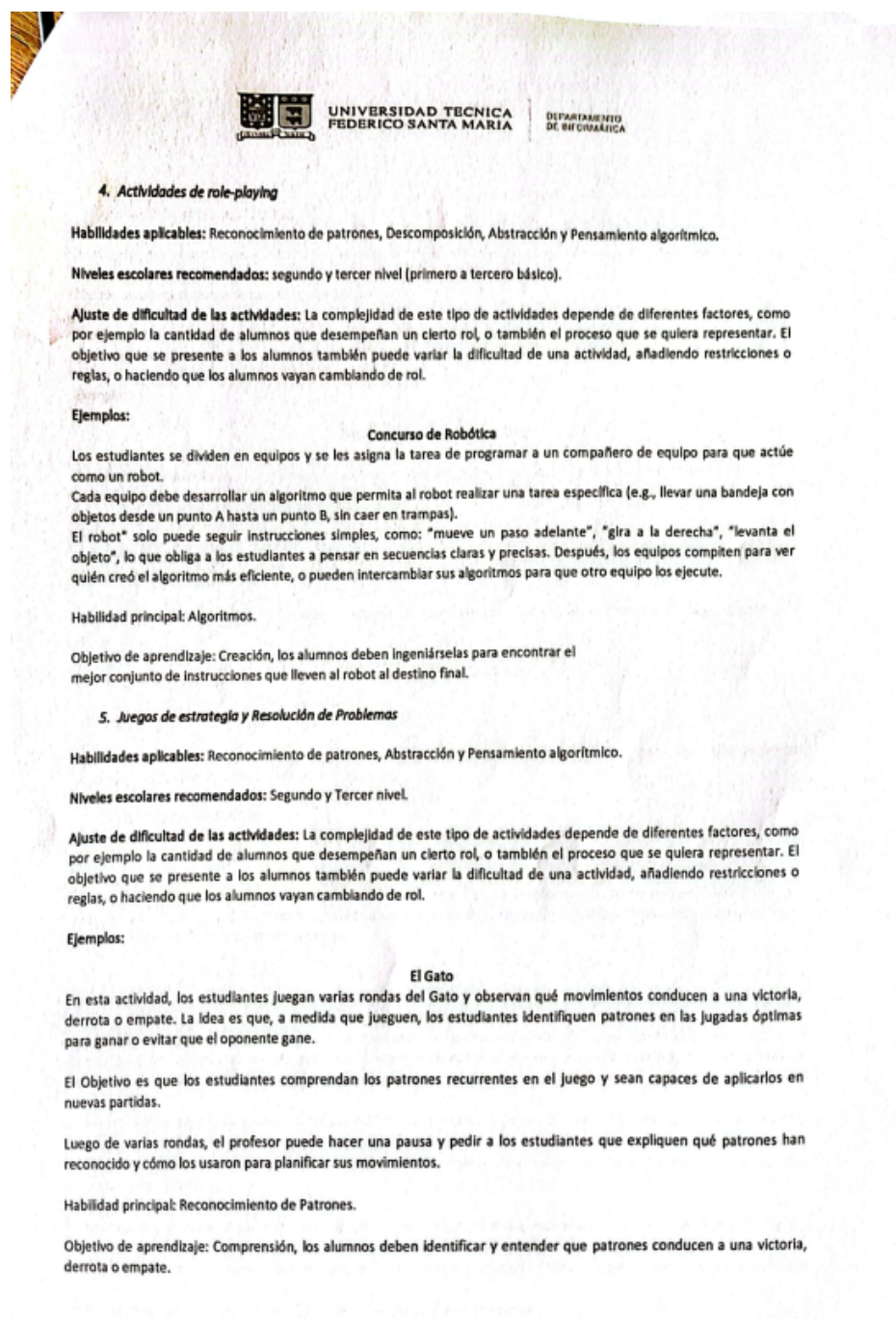


Figura 23: Guía entregada a participantes Hoja 4.
Fuente: Elaboración propia.

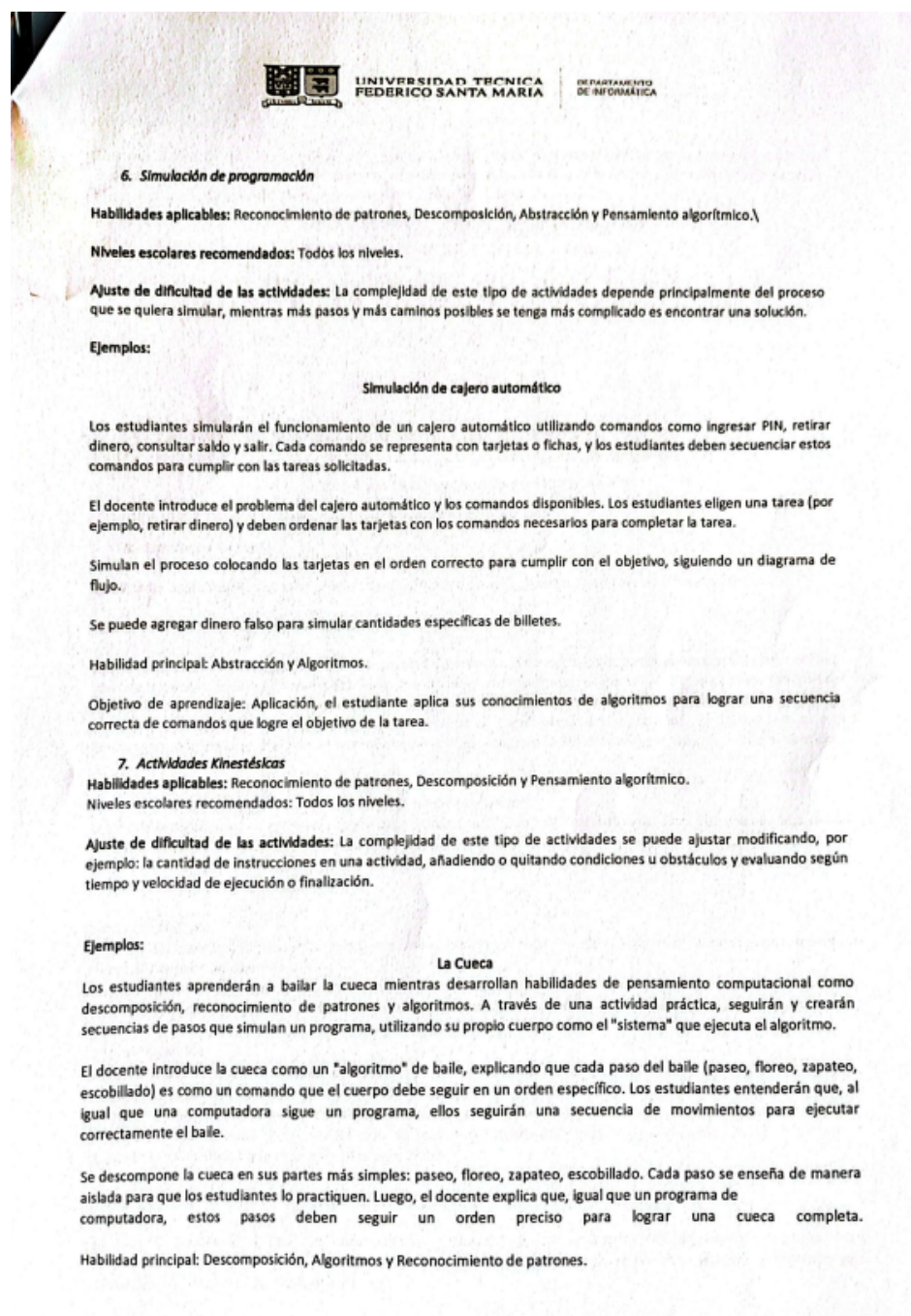


Figura 24: Guía entregada a participantes Hoja 5.
Fuente: Elaboración propia.

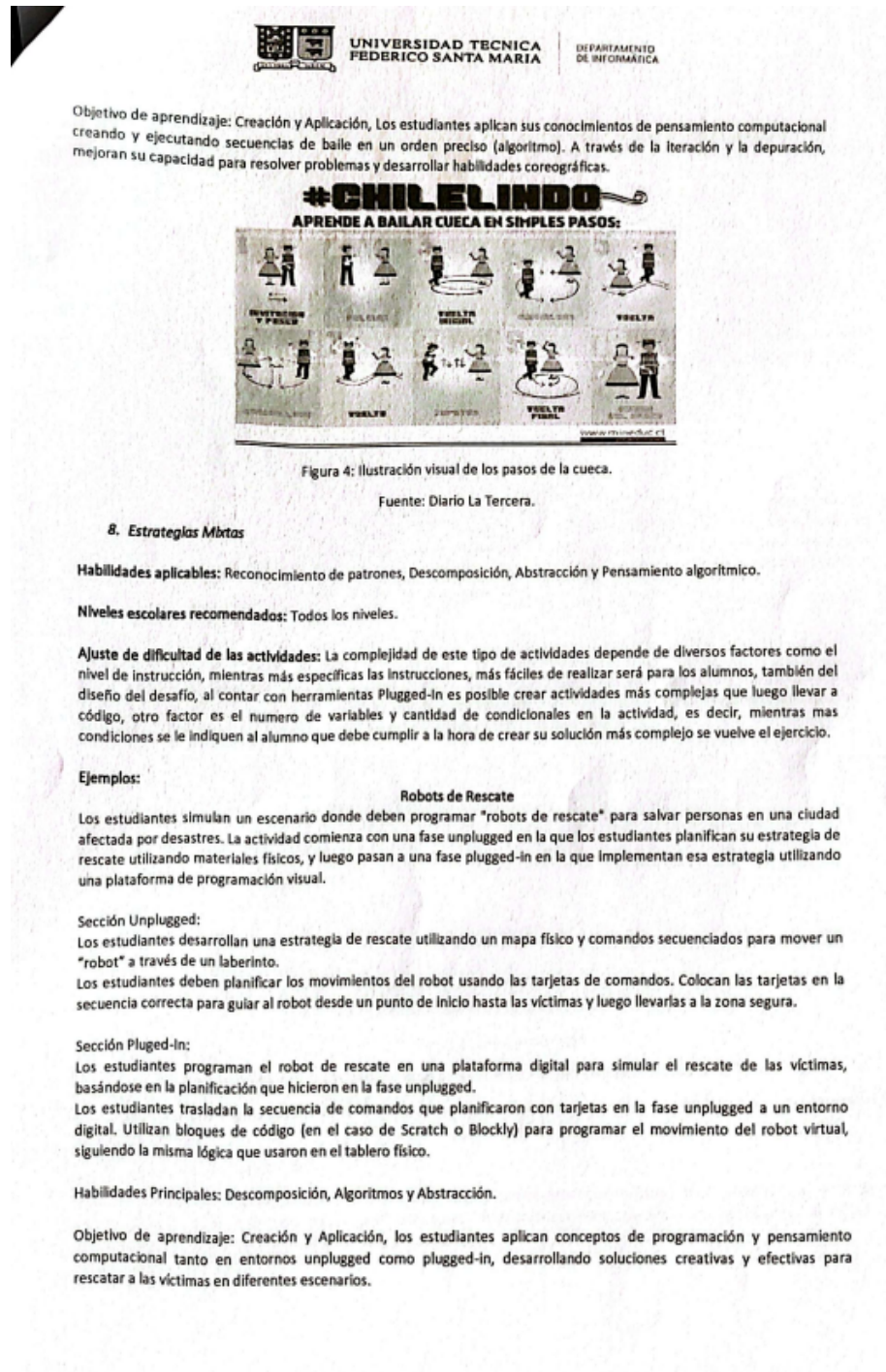


Figura 25: Guía entregada a participantes Hoja 6.
Fuente: Elaboración propia.

REFERENCIAS BIBLIOGRÁFICAS

- [Brackmann, 2017] Brackmann, D. B. . M. R.-G. . G. R. . J. M.-L. . A. C. . C. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE '17)*, pp. 65–72.
- [Charoula Angeli, 2020] Charoula Angeli, M. G. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105(106185):0747–5632.
- [diSessa A.A. et al, 2020] diSessa A.A. et al, L. Y. S. A. (2020). On computational thinking and stem education. *Journal for STEM Educ Res*, 3:147–166.
- [D.R., 1956] D.R., B. B. . K. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longmans, Green.
- [Earp, 2018] Earp, B. S. . C. A. . J. (2018). The nordic approach to introducing computational thinking and programming in compulsory education. *Report prepared for the Nordic@BETT2018 Steering Group*.
- [Jadue, 2002] Jadue, G. (2002). Factores psicologicos que predisponen al bajo rendimiento, al fracaso y a la desercion escolar. *Estud. pedagóg*, (28):193–204.
- [Javier del Olmo Muñoz, 2020] Javier del Olmo Muñoz, Ramón Cózar Gutiérrez, J. A. G. C. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers and Education*, 150(103832).
- [Jiahong Su, 2023] Jiahong Su, W. Y. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, 4(100122):2666–5573.
- [Manches, 2020] Manches, J. R. . P. E. M. . G. R. . A. (2020). Identifying embodied metaphors for computing education. *Computers in Human Behavior*, 105(105859).
- [Mannila, 2014] Mannila, A. S. . V. D. . B. D. . N. G. . C. M. . L. R. . L. (2014). Computational thinking in k-9 education. In *Proceedings of the Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference (ITiCSE-WGR '14)*, pp. 1–29.
- [McTighe, 2005] McTighe, G. W. . J. (2005). Understanding by design. *Colombian Applied Linguistics Journal*, 19(1):140–142.
- [Papert, 1980] Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.

- [Piatti, 2022] Piatti, F. M. . G. A. . L. E.-H. . L. N. . D. A. . L. G. . A. (2022). The ct-cube: A framework for the design and the assessment of computational thinking activities. *Computers in Human Behavior Reports*, 5(100166).
- [Pizarro, 2020] Pizarro, D. P. M. . R. H. N. . A. B. . C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105(105849).
- [Sergio, 2006] Sergio, T. (2006). Formación basada en competencias: pensamiento complejo, diseño curricular y didáctica. *Ecoe*, 2.
- [Sullivan, 2014] Sullivan, M. U. B. . L. F. . E. R. K. . A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72:145–157.
- [Sun, 2023] Sun, H. E. C. . D. S. . T.-C. H. . Y. Y. . J. (2023). Visualising trends in computational thinking research from 2012 to 2021: A bibliometric analysis. *Thinking Skills and Creativity*, 47(101224).
- [Susana., 2016] Susana., B. . C. . B. . D. . C. . A. . B. . R. . M.-H. . (2016). Computational thinking: Panorama of the americas. *Computational thinking: Panorama of the Americas*, pp. 1–6.
- [Tsavara, 2017] Tsavara, N. M. . M. K. . B. M. . P. N. . T. U. . K. (2017). Training computational thinking: Game-based unplugged and plugged-in. *In 11th European conference on game-based learning*, pp. 687–695.
- [Vygotsky, 1978] Vygotsky, L. S. (1978). Mind in society: Development of higher psychological processes. *Harvard University Press*.
- [Weipeng Yang, 2023] Weipeng Yang, Davy Tsz Kit Ng, J. S. (2023). The impact of story-inspired programming on preschool children's computational thinking: A multi-group experiment. *Thinking Skills and Creativity*, 47(101218):1871–1871.
- [Wing, 2006] Wing, J. (2006). Computational thinking. *Commun ACM*, p. 49(3):33–5.
- [Wing, 2008] Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3717–3725.
- [Wing, 2011] Wing, J. M. (2011). Research notebook: Computational thinking—what and why. *The Link Magazine*, 6:20–23.
- [Wu, 2019] Wu, S. P. . L. C. . H. M. . W. B. . L. (2019). Educational policy and implementation of computational thinking and programming: Case study of singapore. *Computational Thinking Education*.
- [Zhou, 2021] Zhou, L. S. . L. H. . D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity*, 42(100926):1871–1871.