

HABILIDADES DE PENSAMIENTO COMPUTACIONAL QUE MUESTRAN ESCOLARES DE PRIMARIA AL ABORDAR DOS TAREAS CON SCRATCH

COMPUTATIONAL THINKING SKILLS DISPLAYED BY ELEMENTARY SCHOOL STUDENTS WHEN TACKLING TWO TASKS USING SCRATCH

Rodríguez, M.¹; Vergara, A.²; Vera-Medinelli, M.¹; Esparza, P.¹

Universidad de Playa Ancha de Ciencias de la Educación¹, Universidad Católica del Maule²

RESUMEN

En este artículo se presentan resultados de una investigación que convocó a estudiantes chilenos de entre 10 y 12 años. El propósito del estudio fue documentar habilidades de *pensamiento computacional* cuando dichos estudiantes desarrollaron dos tareas que fueron diseñadas con el software SCRATCH, tanto en modalidad conectada como desconectada. El paradigma de la investigación fue mixto, con un enfoque exploratorio y descriptivo mediante un estudio de caso. Para el tratamiento de los datos se utilizó la *estadística implicativa* y, como estrategia metodológica, las tareas se implementaron siguiendo las tres primeras fases de un modelo socio-cognitivo. El análisis de los resultados se llevó adelante en función de las características del pensamiento computacional. Como principal hallazgo podemos indicar que los estudiantes son capaces de seguir rutinas, proponer ejemplos y dar soluciones específicas a las dos tareas propuestas. Sin perjuicio de aquello, les dificulta formular conjeturas y establecer relaciones aritméticas mediante lenguaje natural cuando se le solicitó que clasificaran figuras que se forman mediante un proceso recursivo.

Palabras clave: Tareas matemáticas, pensamiento computacional, SCRATCH, estadística implicativa.

ABSTRACT

This article presents the results of a research study that involved Chilean students aged between 10 and 12 years. The purpose of the study was to document computational thinking skills when these students completed two tasks designed using the SCRATCH software, both in connected and disconnected modes. The research paradigm was mixed, employing an exploratory and descriptive approach through a case study. Implicative statistics were used for data treatment, and as a methodological strategy, the tasks were implemented following the first three phases of a socio-cognitive model. The analysis of the results was conducted based on the characteristics of computational thinking. The main finding indicates that students are capable of following routines, proposing examples, and providing specific solutions to the two proposed tasks. However, they find it challenging to formulate conjectures and establish arithmetic relationships using natural language when asked to classify figures formed through a recursive process.

Key Words: Mathematical Tasks, Computational Thinking, SCRATCH software, Implicative statistics

INTRODUCCIÓN

Hoy en día, existe consenso por parte de investigadores y educadores sobre el valor intrínseco que tiene el enfoque de *habilidades para el siglo XXI*, a nivel educativo (Bassi, *et al.*, 2012; Hargreaves, 2003; Pellegrino & Hilton, 2012). Pues, a partir de este conjunto de habilidades se cristalizan aquellas capacidades que dan sustento a una educación integral y de calidad, la cual es indispensable para abordar los desafíos que nos impone este nuevo siglo, en el sentido de lo que plantea Delors (1996). De esta forma, las habilidades para el siglo XXI se están transformando en un marco de referencia para que diferentes países puedan sustentar y direccionar sus políticas educativas (Bassi *et al.*, 2012).

De acuerdo con el informe *Partnership for 21st Century Skills* (2019), cuatro son las principales habilidades que conformar el marco de referencia para el aprendizaje del siglo XXI: *pensamiento crítico y resolución de problemas, comunicación efectiva, colaboración y creatividad*. El *pensamiento crítico* permite distinguir entre información válida y no válida y la *resolución de problemas* refiere a la capacidad para tomar decisiones frente a problemas abiertos, nuevos o complejos. La *comunicación efectiva* permite transmitir ideas, colaborar con otros y liderar equipos en un mundo cada vez más interconectado; esta habilidad implica escuchar activamente y comprender diferentes puntos de vista. La *colaboración* involucra tanto el desarrollo de aprendizajes individuales, como el trabajo en equipo, favoreciendo la interacción en torno a metas comunes. La *creatividad*, por otro lado, nos ayuda a pensar de manera flexible para abordar problemas complejos de la vida cotidiana —aquellos que requieren de flexibilidad y de un conocimiento interdisciplinar—. Estas habilidades pueden ser problematizadas en función de los territorios y las realidades locales como insumo para el proceso de enseñanza aprendizaje de las matemáticas, tanto a nivel escolar como universitario. Cabe precisar que dichas habilidades se agrupan en cuatro dimensiones: *formas de pensar, formas de vivir en el mundo, formas de relacionarse con otros y herramientas para integrarse en el mundo* (Delors, 1996).

Es importante destacar que las habilidades para el siglo XXI apuntan a promover el *aprendizaje e innovación* y el *alfabetismo digital*, con miras a mejorar la *vida personal y profesional* de las personas. Esto último, está en estrecha relación con el desarrollo educativo, profesional y personal de un sujeto que aprende (Pellegrino y Hilton, 2012). Por lo tanto, nos parece plausible indagar el rol que tienen algunas de esas habilidades sobre otras más específicas, en la escuela, como son el *pensamiento matemático* y el *pensamiento computacional*. Respecto de eso último, son temas de interés creciente los roles que juegan el *pensamiento crítico*, la *resolución de problemas*, la *comunicación* y el *trabajo en equipo* en el proceso de aprendizaje y enseñanza de las matemáticas a nivel escolar.

En atención a la interrogante anterior, pensando en una perspectiva más local, creemos necesario poner de relieve las *bases curriculares* para la asignatura de matemática en Chile. Desde ese instrumento es posible apreciar los cambios de perspectiva de cómo el Ministerio de Educación entiende la manera de encauzar los aprendizajes de los estudiantes chilenos (Ministerio de Educación [MINEDUC], 2018). Por ejemplo, el foco ya no está sólo en cubrir los contenidos de la asignatura de matemática, sino que, además, se debe atender el desarrollo de habilidades de pensamiento matemático —*Comunicar y argumentar, representar, modelar y resolver problemas*— las que idealmente deben alcanzar todas y todos los estudiantes (MINEDUC, 2018). Estas habilidades son transversales a los contenidos matemáticos del currículo escolar chileno, sin dejar de mencionar el rol protagónico que ha ido adquiriendo la tecnología en el proceso de enseñanza aprendizaje. En relación a eso último, se destaca la iniciativa que lleva adelante el Centro de Innovación del MINEDUC (CIM), quienes promueven el desarrollo de habilidades comunicacionales y digitales desde dos ejes fundamentales —*Innovación educativa y transformación digital*—, en el marco del plan nacional de lenguas digitales que impulsa nuestro país (CIM, 2018).

Actualmente, los profesores y profesoras que se desempeñan en el sistema escolar, deben enseñar contenidos matemáticos —revirtiendo el enfoque mecanicista y empirista en el sentido de Treffers (1987)— poniendo el foco en la resolución de problemas. Además, tienen el desafío de promover en todos sus estudiantes diferentes tipos de habilidades. Principalmente, aquellas que dicen relación con el pensamiento matemático y el uso de Tecnologías de la Información y Comunicación (TIC) (MINEDUC, 2018). Dado lo anterior, explorar aquellas habilidades del siglo XXI, que puedan conectarse con el *pensamiento matemático* y el *pensamiento computacional*, en el marco del currículo escolar, sigue siendo un desafío, especialmente cuando se abordan tareas matemáticas con un soporte tecnológico. En particular, para nuestro caso, se espera indagar en aquellas habilidades de *pensamiento computacional* que surgen al abordar tareas matemáticas diseñadas con programación basada en bloques, a través del uso del software SCRATCH y que se relacionan *a priori* con habilidades del pensamiento matemático.

1.-ANTECEDENTES

Desde la publicación de los trabajos de G. Polya sobre la Resolución de Problemas (RP) en matemática con un enfoque educativo (Polya, 1945), se ha ido incrementado el número de investigadores que se han enfocado en el desarrollo de perspectivas teóricas, estrategias didácticas y enfoques para estudiar el impacto de la RP en el aula (Törner *et al.*, 2007). Lo anterior, con el propósito de transformar la RP en una habilidad y, de paso, desarrollar los contenidos matemáticos (Felmer *et al.*, 2019). Pese a esos esfuerzos, los resultados en mediciones internacionales dan cuenta que los estudiantes, en general, no pueden resolver situaciones problemáticas de forma adecuada. Por ejemplo —según los resultados de la prueba PISA 2018 en Matemática—, sólo el 2,4% de los estudiantes pueden modelar situaciones problemáticas complejas, y el 10,9% pueden seleccionar, comparar y evaluar estrategias apropiadas de RP. En Chile, a modo de ejemplo, solo un 1% de los estudiantes logró alcanzar el nivel superior en matemáticas (OECD, 2019). En términos globales, la dificultad de la RP en matemática se observa en los estudiantes de distintos niveles educativos como se indica en la revisión de la literatura que realizó Ukobizaba *et al.* (2021).

1.1.-La resolución de problemas no rutinarios

Actualmente, la RP es vista como una estrategia para la enseñanza y aprendizaje, lo que ha servido de referencia para adaptar los currículos de la asignatura de matemáticas en diferentes países. Poniendo de relieve tanto la enseñanza directa de RP como la enseñanza de distintos temas matemáticos a través de la RP (Olivares *et al.*, 2020; Liljedahl *et al.*, 2016; Stanic y Kilpatrick, 1989). Al respecto, cabe indicar que en educación primaria el desarrollo de habilidades para la RP se ha puesto a prueba desde distintas perspectivas (Lubis *et al.*, 2019; Ng *et al.*, 2021; Liang *et al.*, 2018). En el caso particular de Chile, algunas investigaciones a nivel escolar se enfocan en el uso de problemas no rutinarios de corte aritmético para describir las estrategias y procedimientos matemáticos que se despliegan en el proceso de resolución (Rodríguez *et al.*, 2017; Rodríguez y Parraguez, 2014). Entre los principales resultados se puede indicar que entre un estudiante talentoso y uno no talentoso al resolver un problema no rutinario, la diferencia no está en el tipo de estrategias que utilizan, sino más bien en la manera en cómo las articulan para su uso. En general, la estrategia ensayo y error, sumado a la búsqueda de regularidades, son las que más prevalecen en este tipo de problemas, donde los procedimientos matemáticos no son sofisticados, es decir, se ajustan a los requerimientos de la tarea.

1.2.- Uso de SCRATCH y el Pensamiento Computacional

SCRATCH fue desarrollado para programar historias, juegos y animaciones, que se pueden compartir con una comunidad en línea para ayudar a profesores y estudiantes a formar habilidades del siglo XXI (Plaza *et al.*, 2018). Varias son las investigaciones que han incorporado el uso de SCRATCH para promover el pensamiento computacional en jóvenes (e.g., Yoshihara & Watanabe, 2016; Rodríguez-Martínez *et al.*, 2019). Fagerlund *et al.* (2020), por ejemplo, realizan un estudio cuyo foco está en observar aspectos del Pensamiento Computacional (PC) incorporando la RP mediante el uso del software SCRATCH. El propósito de la investigación fue establecer qué elementos del PC se ponen en juego cuando estudiantes resuelven un problema matemático. Lo anterior, les permitió encontrar evidencia empírica para aclarar algunas confusiones y perspectivas que existen en torno al PC.

Rodríguez-Martínez *et al.* (2019) dan cuenta de las bondades que se observaron al incorporar la programación —desde el entorno de programación SCRATCH— para promover aprendizajes de conceptos matemáticos a nivel escolar. El estudio se basó en un diseño cuasi-experimental e incluyó a estudiantes de entre 11 y 12 años para mostrar la influencia del lenguaje de programación en la adquisición de conceptos matemáticos y el desarrollo del PC. El estudio se llevó a cabo desde dos fases, con uso de pre y post test: una centrada en conceptos matemáticos y otra con foco en habilidades de programación. La fase matemática abordó problemas verbales, cuya resolución implicó el uso de los conceptos mínimo común múltiplo y máximo común divisor. La fase asociada a la programación fue diseñada para activar conceptos aritméticos elementales desde problemas rutinarios. Los

resultados mostraron que al utilizar SCRATCH se activan conocimientos asociados al PC. Por otra parte, en la investigación que desarrollaron Cabra Páez y Ramírez Gamboa (2022) se evidenció una mejora significativa en las habilidades matemáticas y de PC, en estudiantes de entre 8 y 9 años, quienes trabajaron tareas aritméticas desde una plataforma con uso del entorno SCRATCH.

Cabe indicar que, en la revisión de la literatura realizada, se evidencian experiencias curriculares y extracurriculares con estudiantes de primaria, que muestran un impacto favorable en las habilidades de PC y de RP al utilizar el software SCRATCH (Moschella y Basso, 2020; Panskyi *et al.*, 2019). En el estudio de Aminah *et al.* (2023), por ejemplo, se presentan resultados sobre el efecto favorable que hay en la habilidad de RP al promover problemas basados en la confección de una calculadora mediante el software SCRATCH. En Basogain y Olmedo (2020) se documentan dos experiencias de aprendizaje colaborativo online con estudiantes de primaria, con la colaboración de expertos que apoyaron el trabajo de los docentes con sus estudiantes. El foco fue estudiar el desarrollo de habilidades de PC y su efecto en la resolución de problemas, a través de la programación con SCRATCH.

En el estudio que realizaron Li *et al.* (2021) se pone de manifiesto que el PC es una competencia fundamental para los estudiantes en el siglo XXI. Los autores aclaran que hay una tendencia a creer que las habilidades de PC y las habilidades de programación son sinónimos. PC es un conjunto más amplio de habilidades cognitivas que, a su vez, forman una parte integral o fundamental de la resolución de problemas complejos. Enseñar y evaluar PC no deben centrarse solo en la programación mediante computadoras. En este mismo estudio, se propuso un enfoque de aprendizaje basado en el diseño (DBL) para mejorar habilidades de PC en estudiantes de un establecimiento educacional. Dicho enfoque no solo incluyó habilidades de programación, también incorporó prácticas de PC en contextos auténticos de la vida real, evaluando el PC de una manera más profunda. Para ello diseñaron un estudio cuasi-experimental para evaluar cómo se asoció la experiencia con las habilidades de PC auto percibidas por los estudiantes, utilizando métodos tanto cuantitativos como cualitativos. En el grupo control, veintitrés estudiantes de entre 9 y 10 años participaron en tres unidades curriculares para crear artefactos que les permitieran resolver problemas específicos del mundo real, utilizando LabPlus electronic kits y el software de programación SCRATCH.

Así, si bien el uso de softwares como SCRATCH y otros, tiene como propósito directo la programación, el PC refiere a habilidades cognitivas más amplias, que se pueden desarrollar a través de actividades conectadas y desconectadas (Tsarava *et al.*, 2017). De hecho, las actividades de programación desconectadas, es decir, que no requieren del uso de dispositivos digitales o cualquier tipo de hardware específico, pueden desarrollar igualmente la capacidad de descomposición, abstracción y comprensión de algoritmos, a través de instrucciones de alto nivel (Curzon *et al.*, 2014). El estudio del pensamiento computacional desde el aprendizaje desconectado es vasto y cubre distintos niveles de enseñanza, con un especial énfasis en estudiantes de menos de 14 años (Rodríguez, Kennicutt *et al.*, 2017; Brackmann *et al.*, 2017; Romero *et al.*, 2022). En general, estas actividades desconectadas consisten en que los estudiantes puedan describir algoritmos utilizando instrucciones paso a paso, o bien, observar patrones para luego descubrir y reproducir el algoritmo que hay detrás. Por ejemplo, Yildiz y Karal (2021) implementan un juego basado en un mapa de ciudad donde los estudiantes de 6to grado debían describir paso a paso las indicaciones para desplazarse desde un lugar a otro; Tonbuluğlu y Tonbuluğlu (2019) implementan, con estudiantes de 5to grado, un proyecto asociado a la codificación desconectada para reconocer y crear algoritmos y describirlos paso a paso.

A lo anterior, se suman investigaciones que han comparado el desarrollo del pensamiento computacional, considerando tanto un enfoque conectado como desconectado, advirtiendo la necesidad de seguir explorando las ventajas de las actividades desconectadas en prácticas de aula (del Olmo-Muñoz, 2020; Romero *et al.*, 2022; Tsarava *et al.*, 2017). Además, el enfoque de pensamiento computacional desconectado es especialmente importante para llevar estas habilidades a las escuelas que no cuentan con los recursos tecnológicos adecuados, conexiones a Internet o incluso energía eléctrica (Brackmann *et al.*, 2017).

En atención a los antecedentes que se han presentado, en esta investigación se hace uso del software SCRTACH, con foco en la programación basada en bloques, para implementar dos tareas de programación, una conectada y otra desconectada, atendiendo a la siguiente pregunta de investigación: ¿cuáles son las habilidades de pensamiento computacional que despliegan escolares de entre 10 y 12 años, al abordar dos tareas matemáticas diseñadas con el software SCRATCH?

2.-MARCO CONCEPTUAL

Para dar respuesta a la pregunta de investigación, se consideraron elementos tanto de la Teoría de las Situaciones Didácticas (TSD) de Brousseau (2007) y los elementos basales que caracterizan las habilidades del pensamiento computacional (Qu y Fok, 2021).

2.1.-La Teoría de las Situaciones Didácticas (TSD)

La TSD es un marco teórico socio-cognitivo, donde el concepto de *situación* —entendida como un modelo de interacción entre los aprendices (los estudiantes) y un cierto *medio* (los recursos)— permite que un profesor/a promueve la mediación de un aprendizaje, el cual se espera emerja a partir de la propia situación, la que puede a no hacer referencia directa a conceptos o nociones matemáticas. Eso último ocurre mediante la activación gradual, no necesariamente lineal, de determinados conocimientos y esquemas, en un sentido piagetiano (Arnon *et al.*, 2014). En esta misma línea, cabe indicar que hay situaciones que tienen una relación directa con algún conocimiento matemático específico, las que se denominan *situaciones didácticas*. En particular, las situaciones didácticas son modelos que describen la actividad tanto del profesor como del estudiante, donde el foco está dado por una tarea matemática particular. Por otro lado, están aquellas situaciones que no están ligadas *a priori* a un contenido matemático, como ocurre en la ya conocida *carrera del 20*. Dichas situaciones se conocen como situaciones *adidácticas* (Brousseau, 1997).

Un aspecto fundamental en la TSD es que se debe observar la evolución de una situación, puesto que el aprendizaje se logra por medio de una adaptación del sujeto que aprende al medio creado por dicha situación. Los conocimientos se manifiestan como instrumentos de control de las situaciones (Brousseau, 2007). Para dicho efecto, resulta fundamental conocer esas fases a las que están sujetas las situaciones *didácticas* y/o *adidácticas*. Se debe tener presente que dichas fases no necesariamente son lineales. A continuación, se presentan tres de las cuatro fases, considerando las características específicas de la tarea/problema que se propone en el estudio (para más detalles de la tarea/problema ver el aparatado metodológico).

Situación de Acción: El estudiante toma decisiones, a medida que éste se involucra y descubre intuitivamente un camino para involucrarse en la tarea y, por ende, tiene razones para elegir sus opciones. Requiere varias partidas para que sean capaces de formular la táctica, justificar y sacar conclusiones. Una estrategia se adopta o rechaza intuitivamente o racionalmente respecto de una estrategia anterior. Cuando se origina una estrategia nueva, se somete a la experiencia para aceptarla o rechazarla según la eficacia que tenga para el alumno. La sucesión de situaciones de acción constituye el proceso por el cual el alumno va a “aprender” un método de resolución de su problema. Surge aquí el modelo implícito donde el alumno toma decisiones a partir de un conjunto de relaciones o reglas sin tener conciencia de ella a posteriori de formularlas.

Situación de formulación: En la segunda fase se observan dos momentos. Primero cuando el representante del equipo está al frente y juega, y el segundo momento cuando el equipo discute. En el primer momento un estudiante que no está al frente registra la información (recoge), pero no puede actuar (intervenir). En el segundo momento el “medio” está constituido por el conjunto de partidas jugadas, en especial por la última. En la situación formulación el niño que está en el pizarrón está en situación acción y el niño que anota está en formulación. Se deben explicar las estrategias, comunicándose.

Situación validación: En esta situación los equipos elaboran y luego proponen por turno un enunciado o intentan argumentar que el enunciado del adversario es falso. Organizan enunciados en demostraciones y construyen explicaciones o justificaciones para sus conjeturas.

Situación Institucionalización: En esta etapa el profesor, en atención a los distintos aspectos que se desplegaron, en las fases anteriores, muestra a sus estudiantes el conocimiento matemático que estuvo en juego en la situación *adidáctica*.

Para Brousseau (2007), la acción, luego la formulación, la validación cultural y la institucionalización parecen construir un orden razonable para la construcción de saberes. La construcción de

conocimiento se ve influenciada por la existencia de obstáculos (Brousseau, 2007). El obstáculo, como un conocimiento ya existente, se opone al nuevo conocimiento y reaparece en el aprendizaje de un nuevo conocimiento, opone resistencia a la adquisición de este nuevo conocimiento. Por este motivo no es conveniente ignorar un obstáculo y se sugiere integrar su negación en el aprendizaje de un conocimiento nuevo, bajo la forma de contraejemplo.

El medio material se refiere a la preparación de la clase, incorporando las reglas, aciertos y errores. Además, debe considerar las interacciones de un sujeto con este medio, al que Brousseau (2007) llama actor objetivo. Esta dupla, medio-actor, constituye la situación objetiva que se propone al alumno para interactuar. En el medio objetivo y ante una situación objetiva, el estudiante se encuentra en posición de sujeto que actúa. El medio objetivo es movilizadado en una situación de acción, donde se motiva al alumno a imaginar un medio ficticio para responder a una pregunta. El sujeto aprende corrigiendo sus acciones y anticipando sus efectos. En la situación es sujeto de aprendizaje, estas situaciones exigen realizar una reflexión en la práctica, para que efectivamente surjan posibilidades de aprender (Brousseau *et al.*, 2014).

2.2.-Pensamiento computacional

Para Selby y Woollard (2014) el concepto de *Pensamiento Computacional* (PC) emerge en repuesta a la RP desde la informática, ya que la RP se da en distintos ámbitos y contextos, incluyendo o no el uso de tecnología. De hecho, algunas investigaciones proponen que el PC es una forma de pensar que está intrínsecamente ligada con la resolución de problemas (Hazzan *et al.*, 2020; Bocconi *et al.*, 2022). Por otro lado, dicha relación está enraizada a las habilidades del siglo XXI, las que a nivel internacional se sugieren para el currículo escolar tanto primario y secundario como universitario. Algunos autores plantean que, en el desarrollo del PC se identifican elementos basales de las habilidades del siglo XXI, los que, a su vez, permiten su caracterización (Wing, 2006 y 2008). Aquellos elementos permiten comprender el rol formalizador y sistematizador del PC, en el sentido que organiza y estructura diferentes nociones, conceptos, procedimientos y técnicas que se dan tanto en el ámbito de la RP como en la informática (Lu y Fletcher, 2009; Barr y Stephenson, 2011; L'Heureux *et al.*, 2012; Voskoglou & Buckley, 2012).

En atención a lo que se ha indicado, se puede decir que, para entender en esencia el concepto de PC, se debe tener en cuenta un *proceso de pensamiento*, el concepto de *abstracción* y el concepto de *descomposición* (Selby y Woollard, 2014). El *proceso de pensamiento* dice relación con la manera en que los seres humanos abordan la resolución de problemas que son complejos, los que provienen tanto del mundo real como de fenómenos de corte más teórico o artificial, en distintos ámbitos del conocimiento. En dicho proceso se pone en juego procedimientos mentales, además de conceptos y técnicas algorítmicas que están ligadas al quehacer de la informática. En relación al concepto de *abstracción*, este hace alusión a la idea de esquema mental o bien de esquema de acción, en un sentido más piagetano (Arnon *et al.*, 2014). En otras palabras, el proceso de pensamiento resulta ser una herramienta heurística que guía la resolución de un problema sofisticado, atendiendo las distintas fases que se han propuesto para dicha habilidad a nivel educacional (Polya, 1990). Cabe indicar que la literatura considera otro tipo de abstracción más específica como es la abstracción de datos (Voskoglou & Buckley, 2012), así como también la abstracción reflexiva y la abstracción empírica (Arnon *et al.*, 2014). Por último, está la idea de *descomposición* que tiene que ver fundamentalmente con el desglose de un problema complejo en otros problemas más elementales, para así buscar similitudes o desplegar heurísticas que permitan comprender y abordar el problema más complejo.

En cuanto a las habilidades básicas del pensamiento computacional, Bocconi *et al.* (2016) establece las siguientes: abstracción, pensamiento algorítmico, automatización, descomposición, depuración y generalización. Siguiendo esta línea, Qu y Fok (2021), con base en una revisión de la literatura, proponen una rúbrica para evaluar el pensamiento computacional en procesos de educación obligatoria, considerando seis indicadores: formulación de problemas, abstracción, pensamiento lógico, uso de algoritmos, análisis e implementación de soluciones y finalmente, generalización y transferencia de problemas. Estas categorías serán usadas como base para el análisis de las habilidades de pensamiento computacional que surjan en los estudiantes al resolver las tareas propuestas.

3.-MÉTODO

En este apartado se comparte el análisis *a priori* de las dos tareas que se presentaron a los 35 estudiantes que participaron en esta investigación —cuyas edades fluctuaron entre los 10 a 12 años— provenientes de una escuela de dependencia municipal en la quinta región, Chile. Lo anterior, con el propósito de documentar aquellas *habilidades de pensamiento computacional* en correlato con los diferentes recursos cognitivos que los estudiantes desplegaron al trabajar las dos tareas. Además, se describen las características de cada tarea y su relación con los aspectos propios del PC. Por otro lado, el estudio tiene un carácter mixto, descriptivo y no correlacional, basado en un estudio de caso instrumental (Stake, 2010). Para implementar las dos tareas en el aula escolar, los investigadores/as y el profesor de la escuela, trabajaron en un *Codiseño* siguiendo sólo las tres primeras fases de la TSD (Brousseau, 2007) —ya que el foco de la investigación se enfoca en el actuar de los estudiantes al desarrollar cada una de las tareas— las que se describieron en el marco conceptual. En lo que respecta a la unidad de análisis, esta quedó conformada por los videos de las clases de los dos cursos, los registros de los estudiantes de la Tarea 1 y los archivos de programación que se generaron en la Tarea 2. Por último, teniendo como referencia las características planteadas por Bocconi *et al.* (2016) y los indicadores de la rúbrica de Qu y Fok (2021), se elaboran seis categorías de análisis, cada una con los respectivos indicadores observables que se presentan en la Tabla 1, con el propósito de orientar la obtención de las categorías de análisis *a priori* al realizar el análisis preliminar de cada una de las dos tareas.

Tabla 1. Categorías del pensamiento computacional que se proponen de la literatura sobre PC.

Categorías	Características	Indicadores
Formulación de problemas (FP)	Sigue una ruta específica de un problema realizando un proceso de descomposición.	i.1. Se enfoca en características particulares del problema, lo simplifica y actúa mediante un proceso inductivo o aleatorio en busca de alguna conjetura.
Abstracción (AB)	Representa datos a través de abstracciones como modelos y/o simulaciones o secuencia de pasos.	i.2. Comunica una conjetura, hipótesis, juicios o ejemplos, los que eventualmente podrían estar acompañados de propuestas para procesos de simulación.
Pensamiento lógico (PL)	Piensa en datos específicos, los organiza lógicamente y, mediante el uso de algún formato reconocible, los analiza.	i.3 Expresa la posición, la orientación o el trazado de objetos usando puntos de referencia mediante lenguaje oral, escrito, gestual y simbólico, secuenciando acciones o conectando con la programación.
Usando algoritmos o programación (AP)	Sugiere mediante lenguaje natural o diagramas, soluciones de automatización a través del uso de algoritmos (una serie de pasos ordenados) o mediante secuencias de programación.	i.4 Aplica operaciones aritméticas paso a paso en correlato con los requerimientos de la tarea.
Análisis e implementación de soluciones (IS)	Identifica, analiza e implementa posibles soluciones con el objetivo de lograr la más eficiente y efectiva combinación de pasos y recursos. Involucra procesos de depuración.	i.5 Explica de manera coherente y, desde algún registro de representación, posibles soluciones al problema.
Generalización (G)	Realiza un proceso de reconocimiento de patrones y transferencia de una solución conocida a otra variedad de problemas similares.	i.6 Utiliza lenguaje natural o gestual para comunicar patrones que son verificables por otros.

Adaptado de Qu y Fok (2021)

3.1.-Las dos tareas y el respectivo análisis *a priori*

A continuación, se presenta un análisis *a priori* de cada una de las tareas que se implementaron en el aula, en primer lugar, damos a conocer los aspectos matemáticos que cada tarea involucra. Luego, los aspectos a tratar en cada una de las fases de la TSD. Finalmente, el proceder de los estudiantes en sintonía con los indicadores de PC.

3.1.1.- Análisis a priori tarea 1

Aspectos matemáticos de la tarea 1

La tarea 1, es una tarea de programación desconectada, donde el software hace las veces de medio para explorar y verificar las figuras resultantes. La tarea requiere que los estudiantes consideren tres números naturales entre 0 y 7, los que corresponderán a la longitud de tres trazos. Así, la aplicación dibuja los trazos en sentido horario, con un giro de 90°, hasta que la figura se cierre. En la Figura 1 se muestra un ejemplo de la figura que se forma al considerar los números 1, 2 y 3.



Figura 1. Figura que se forma con los números 1, 2 y 3 denominada “remolino”.

Plan de clase Tarea 1, según las tres primeras fases de la TSD:

Fase de acción: Los estudiantes deben reproducir la secuencia de pasos para lograr dibujar una poligonal. El rol del profesor, en esta fase, es verificar que las y los estudiantes comprendan la tarea y sigan la secuencia de tres números una vez. Una o dos poligonales de los propios estudiantes en la pizarra puede llevar a preguntas como: ¿Cuál será la longitud de ambas líneas por separado? ¿Qué otra forma tendrá la poligonal?, entre otras preguntas. *Fase de Formulación:* En esta fase se activa el uso del software SCRATCH y es probable que los estudiantes formulen, mediante lenguaje natural, si los números son iguales, la figura cerrada es un cuadrado. Además, si se dan tríos de números al azar, los estudiantes pueden proponer nombres según la forma de cada figura: remolino, aspa, tubería, entre muchos otros. *Fase de Validación:* En esta fase se pueden aceptar las conjeturas mediante la verificación de ejemplos positivos o refutar mediante contraejemplos que salgan de los propios estudiantes.

En la Tabla 2, presentamos un proceder de los estudiantes en sintonía con las habilidades de PC.

Tabla 2. El proceder (p) de los estudiantes en correlato con las categorías de PC asociados a la Tarea 1.

hPC	p	Indicador observable
FP	p1	Propone diferentes ternas de números para generar una figura.
AB	p2	Rotula las figuras con palabras que hacen referencia a objetos cotidianos.
PL	p3	Fija a lo más dos números en una determinada terna.
PL	p4	Invierte el orden de los números en una terna.
AB	p5	Describe, con palabras claves, el patrón de los trazos que dan forma una figura.
AP	p6	Relaciona, mediante una operación aritmética, los números de una terna.
IS	p7	Identifica ternas de números, asociadas a un mismo tipo de figura.
IS	p8	Relaciona el cambio de tamaño de una figura con la idea de múltiplo.
G	p9	Comunica diferentes relaciones aritméticas con su respectiva figura.

Cabe indicar que, como las fases de la TSD no son lineales, los estudiantes pueden transitar entre estas, volviendo a evocar sus propios ejemplos o cambiar a otras conjeturas a medida que se descartan otras.

3.1.2.- Aspectos de la tarea 2

La tarea 2 es una tarea de programación conectada, que consistió en observar una secuencia de bloques, con el propósito de estimular, en las y los estudiantes, la elaboración de conjeturas sobre lo que ocurrirá con el objeto en la interfaz del software SCRATCH y la propuesta de una secuencia de programación por bloques. Se espera con ello que se describa una poligonal, como la que se indica en la Figura 2.

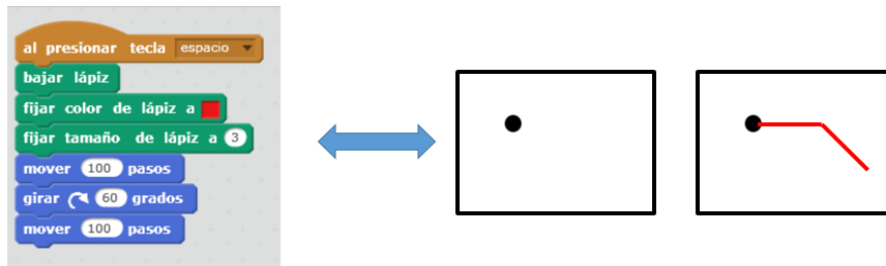


Figura 2. Propósito de la tarea, generar una poligonal.

Plan de clase Tarea 2 según las tres primeras fases de la TSD:

Fase de acción: El profesor propone a los y las estudiantes, que están en una modalidad híbrida, describir mediante el lenguaje natural, lo que eventualmente podría ocurrir con el punto de la interfaz al ejecutar el programa una vez. Idealmente, a uno o dos estudiantes, se les puede pedir que describan un eventual trazado. *Fase de Formulación:* El grupo en modalidad híbrida, con el apoyo de un estudiante que manipula el software, proponen conjeturas de lo que se irá generando al aumentar el número de ejecuciones de la secuencia de bloques, pudiendo resetearse para volver al punto de partida. Se espera que la idea de ángulo y figura se instale en el discurso de los estudiantes. Los y las estudiantes, cambiando el parámetro del ángulo y la magnitud de los trazos deben conjeturar sobre los valores para obtener un cuadrado, un rectángulo y un triángulo rectángulo, entre otras posibilidades, proponiendo una secuencia de bloques. *Fase de Validación:* El grupo curso, acepta o refuta la secuencia de bloque que proponen los otros grupos, argumentando mediante la identificación de alguna inconsistencia.

En la Tabla 3, presentamos un eventual proceder de los estudiantes, en sintonía con las habilidades de PC.

Tabla 3: Proceder (p) de los estudiantes en correlato con las categorías de PC asociados a la tarea 2

hPC	P	Indicador observable
FP	p10	Identifica el rol de cada bloque.
AB	p11	Dibuja figuras utilizando sólo trazos.
AB	p16	Relaciona la forma y/o tamaño de una figura al modificar parámetros de un bloque.
PL	p12	Relaciona el ángulo de giro con un ángulo suplementario.
PL	p15	Descubre el ángulo de giro para formar un triángulo equilátero.
AP	p14	Propone una secuencia de bloques para obtener un rectángulo.
IS	p13	Muestra nuevas figura modificando los parámetros en los bloques.
G	p17	Proponen secuencias de bloques para formar un polígono regular.

Para analizar los datos de las respuestas desplegadas en las tareas, se recurrió a la estadística implicativa (en adelante ASI [Analyse Statistique Implicative] por su sigla en francés) (Gras *et al.*, 2008; Orús *et al.*, 2009), y el uso del software CHIC (Coherive Hierarchical Implicative Classification) versión 7.0 (Couturier, 2008). Una de las características de ASI es que combina dos tipos de análisis estadísticos multivariantes: un análisis de conglomerados (cluster analysis), para descubrir grupos homogéneos de variables, y un análisis de reglas de asociación (association rules), para explorar relaciones no simétricas que permiten establecer relaciones de causa y efecto entre las variables analizadas. La otra característica es que ASI basa sus cálculos en la probabilidad de encontrar una coincidencia entre dos variables (o respectivamente un contraejemplo a una regla) para medir la similitud entre esas dos variables (o la intensidad de una regla que relaciona las dos variables) (Gras *et al.*, 2008; Orús *et al.*, 2009).

El árbol jerárquico de similaridad (dendrograma) se forma de la manera habitual, esto es, comienza con cada variable formando una única clase y agrupando, a cada paso, las dos clases más similares sucesivamente, hasta obtener una única clase que agrupa todas las variables, siendo la interpretación de los grupos hallados el aspecto más importante. Adicionalmente, el árbol de similaridad destaca algunas clases como significativas, por la relación preponderante de los índices de similaridad entre las variables que se hallan reunidas en dicha clase (Orús *et al.*, 2009). Cuando se forma cada clase de variables similares, se puede calcular un índice de contribución de cada informante a la formación de esa clase. Con esos índices, se destaca el grupo de los informantes que más la ha manifestado. Una inspección detallada de ellos, permite ahondar aún más en la descripción de la clase, así como indagar

el posible efecto de otras variables (asociadas a las categorías de análisis) en la formación de dicha clase de variables similares.

El grafo implicativo es la representación gráfica de las reglas de asociación entre parejas de variables. La fuerza de las reglas de asociación se suele medir mediante un índice llamado confianza, esto es, el porcentaje de individuos que cumple la regla de entre los que la podrían cumplir (por verificar el antecedente de la regla), en el sentido clásico de una implicación. Este indicador es útil para predecir si un individuo presenta el rasgo del consecuente, una vez se sabe que presenta el rasgo del antecedente. No obstante, cuando el rasgo consecuente es muy abundante en los informantes, no es evidente que su presencia se deba a un efecto causado por el antecedente de la regla. Así, el grafo implicativo representa las reglas cuyas intensidades superen un determinado límite que determinan los investigadores, y se deben interpretar como asociaciones implicativas estadísticamente significativas. A partir de estas asociaciones inesperadas, se deben distinguir relaciones de causa y efecto que se justifican a la luz de las categorías del análisis a priori, y otras relaciones que no se podrían aprovechar si no hay una justificación adecuada por parte del investigador.

4.-RESULTADOS

Los resultados que se presentan a continuación, están en función de las diferentes instancias que se declararon en el diseño metodológico, a partir de la implementación de las dos tareas. Primero se muestra un panorama general de las categorías de análisis que se plasman en los tres momentos que se han considerado de la TSD, *fase de acción*, *fase de formulación* y *fase de validación*.

4.1.-Resultados Tarea 1

Un aspecto a destacar es que los estudiantes dispusieron de la aplicación en una computadora, en todo momento, para responder a los requerimientos de esta tarea, mediada por una guía de trabajo, como se aprecia en la Figura 3. Sin perjuicio de lo anterior, el profesor mantuvo proyectada la actividad en la pizarra, quien además estuvo acompañado de una asistente de educación para atender a aquellos estudiantes con necesidades especiales.









Figura 3. Distribución de los estudiantes de un curso en la sala de clases

Como estaba previsto, los estudiantes propusieron diferentes tríos de números y pudieron observar distintas figuras que se fueron formando. Ello, les permitió asignar nombres a las distintas figuras que fueron descubriendo, además de relacionar ternas de números con un tipo particular de figura. A continuación, se presentan algunas respuestas que reflejan la activación de algunas de las categorías de *análisis a priori* que se propusieron en el apartado metodológico.

En la Tabla 4, se aprecian distintos nombres que los estudiantes indicaron para distintas ternas de números. Llama la atención los nombres de las figuras que **E1** y **E11** asignaron, pues la disposición de cuatro rectángulos, se espera que se reconociera como un *remolino*. Por otro lado, si bien **E23** reconoce la figura remolino, rotula agregando un atributo más, el tamaño de ésta. Este tipo de respuestas en torno a la figura remolino, puede ser un incentivo, en la *fase de formulación*, para que los estudiantes establezcan una relación entre la forma y su tamaño, con el tipo de número que caracteriza a la nueva figura de la figura referencial —en este caso, remolino con la terna [1 2 3]—. Este es un aspecto que debería utilizarse en la *fase de validación*.

Tabla 4. Nombres que asignaron los estudiantes a las figuras

Estudiante	Nombre	Figura	Estudiante	Nombre	Figura	Estudiante	Nombre	Figura
E22	Margarita [3 2 3]		E35	Flor [3 4 5]		E23	Remolino Grande [1 5 6]	
E1	Ventilador [5 4 1]		E15	Sol [6 1 3]		E11	Hélice [3 5 2]	

En la Figura 4 se aprecia la respuesta que dio E6 para la figura *remolino*, manifiesta que, al cambiar el orden de los números de una terna que da remolino, se logra la misma figura en cuestión. Esta idea fue utilizada por el 57% de los estudiantes. Por otro lado, E6 agrega que la terna [5 1 6], pero no logra notar o comunicar que, al sumar los dos números menores se obtiene el número mayor, que es una regularidad de interés para reconocer este tipo de figuras. Para las dos últimas ternas, fija el primer número, una estrategia a tener presente para alentar a los estudiantes a contar las posibilidades que se tienen a partir de una misma terna de números. No obstante, la idea de fijar números de la terna para indagar en los tipos de figura, como estrategia, no prosperó para ninguno de los estudiantes.


1	2	3	usar números que ya tenemos en diferente orden	
5	1	6		
2	1	3		
2	3	1		

Figura 4. Parte de la respuesta de E6 para indicar ternas asociada a la figura remolino.

Cuando se pide a los estudiantes que propongan ternas asociadas a otra figura que ellos hayan identificado, E6 propone ternas que contienen el mismo número, lo que da origen a un cuadrado. Pese a ello, no logra expresarlo en palabras ni plasmar dicha figura en la cuadrícula. En el trazado, que se observa en la cuadrícula de la Figura 5, E6 cambia el orden en que los trazos se van desplegando, al utilizar la aplicación que dispone en su computador.

Primer Número	Segundo Número	Tercer Número
3	3	3
2	2	2




Figura 5. Parte de la respuesta de E6 para mostrar ternas asociadas a otra figura distinta a *remolino*.

En atención a las respuestas que brindaron los estudiantes a la Tarea 1, se puede ver que la estrategia de simplificar el problema repitiendo los números para configurar una terna, cambiar el orden de los números, utilizar múltiplos para generar una nueva terna o bien atender a la forma y tamaño de un mismo tipo de figura, se asocian con la habilidad de *formulación de problemas*, *abstracción* y *pensamiento lógico*. Por otro, dado que no fue posible observar la necesidad de buscar relaciones aritméticas entre los números y, a su vez, la generalización de casos mediante la comunicación de alguna regla mediante lenguaje natural, se puede establecer que el uso de *algoritmos* y *programación*, así como la *generalización* y *análisis e implementación de soluciones*, son aspectos que no se precian en el trabajo de los estudiantes para esta tarea.

4.2.-Análisis mediante la estadística implicativa

A continuación, se presenta un análisis teórico mediante el ASI y los recursos que provee el software CHIC 7.0. Con dichos recursos, se hará un contraste con los diferentes aspectos de los resultados presentados en el apartado anterior con un enfoque teórico.

4.2.1.-El árbol de similitud

A continuación, como se aprecia en la Figura 6, se presenta el *Árbol de Similitud* que se obtuvo al procesar la matriz de los datos —con la presencia (1) o ausencia (0) de las categorías de análisis *a priori*— utilizando el software CHIC 7.0. En dicho árbol se aprecian 3 clases significativas, las que aparecen destacadas con un trazo de color rojo. La clase 1, es la que presenta una similitud que corresponde a 0.999104. Para las otras dos clases, su índice de similitud es muy bajo para incluirlas en este análisis.

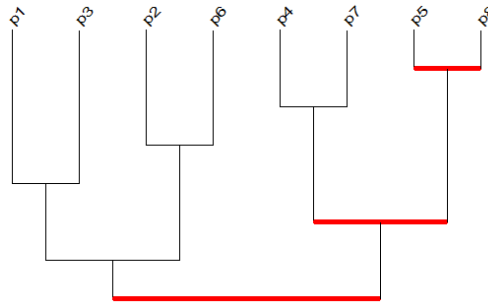


Figura 6. Árbol de similitud sobre las categorías a priori tarea 1.

En la Tabla 5, presentamos los detalles de la clase significativa en cuanto a las categorías que se incluyen en dicha clase, el *índice de similitud* y los estudiantes que *contribuyeron* a que se conformara dicha clase, desde un punto de vista estadístico.

Tabla 5. Clases significativas y su respectivo índice de similitud

Clase	Categorías de la Clase	Similitud	Contribución
1	(p5 p8)	0.999104	E27

En la clase significativa (p5 p8) se pone de relieve el rol que cobra seguir la secuencia de los trazos e invertir el orden de los números de una terna para poder asociar ternas a una misma figura, como se aprecia en la respuesta de **E27**, en la Figura 7.

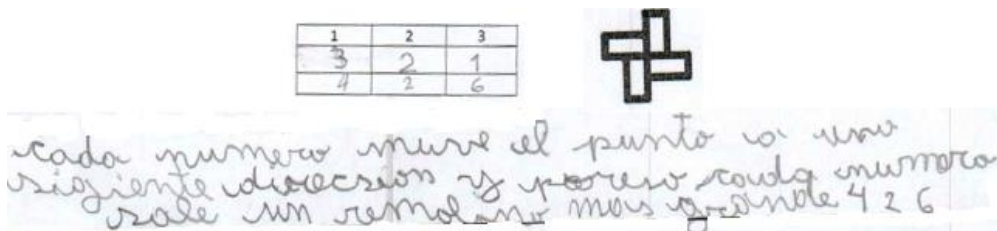


Figura 7. Parte de la respuesta de **E27** para indicar ternas asociadas a la figura remolino.

“Cada número mueve el punto a una siguiente dirección y por eso cada número sale un remolino más grande [4 2 6]”

Por otro lado, **E22** logra dar con dos ternas de una misma figura, similar a remolino, como se aprecia en la Figura 8.

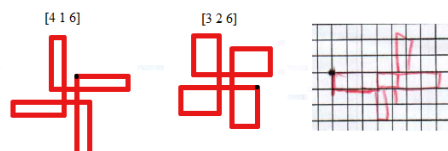


Figura 8. Parte de la respuesta de **E22** para mostrar ternas asociadas a otra figura distinta a Remolino.

4.2.2.-El grafo implicativo

En la Figura 9, se presenta el grafo implicativo. Se aprecia en dicho grafo la implicación de las categorías de análisis con un índice de implicación que se definió mayor e igual a 75. Destaca en él una cadena de implicaciones.

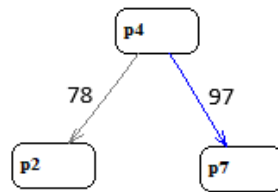


Figura 9. Grafo implicativo de las categorías de análisis a priori de la Tarea 1

De esta cadena de implicaciones destaca, con un índice de implicación del 97%, $p4 \rightarrow p7$, la que pone de relieve una *estrategia* que puede ser representativa en el proceder que tiene este tipo de estudiantes. Con ella se muestra que el cambio de orden de los números en una terna, es un primer camino a tener presente para asociar diferentes ternas a una misma figura, como se observó en la respuesta del **E6**, Figura 4. Sin perjuicio de otras estrategias como repetir un número para cada trío, estrategia de **E6** (Figura 5), o bien usar la idea de múltiplo, como lo planteó el **E27** (Figura 7). Por otro lado la implicación $p4 \rightarrow p2$, pone de relieve que el cambio en el orden de los números en una terna lleva a la clasificación de figuras, en nuestro caso, desde la forma y el tamaño de estas.

4.1.-Resultados Tarea 2

En la Tabla 6 se presenta un panorama general de la manera en que procedieron los estudiantes al trabajar la Tarea 2. Con base en este accionar de los estudiantes, se pudo establecer una relación con las habilidades del PC, considerando para ello las categorías de análisis *a priori* del apartado metodológico. Cabe indicar que no se hizo el análisis estadístico, ya que la tarea requería dibujar, declarar cambio de parámetros y de programar. Los estudiantes no individualizaron su archivo de programación, lo que dificultó asignar la presencia o no de las categorías en esa dimensión.

Tabla 6. Proceder de los estudiantes en la tarea y las habilidades de PC que se asocian.

Proceder de los estudiantes	Habilidades PC
1min 00-min 13: Los estudiantes reciben por parte del profesor ideas y ejemplos de lo que es una figura abierta y cerrada. Aquello, para activar conocimientos previos. A continuación, muestra a los estudiantes lo que genera la secuencia de bloques en la interfaz del software, repitiendo hasta tres veces la misma acción.	AB
min 14-min 19: Los estudiantes describen los elementos geométricos que observan de una figura que va cambiando a medida que aprieta el comando que repite la secuencia de bloques. Primero observan una esquina, dos líneas, un ángulo. Luego, tres ángulos, cuatro líneas, una figura abierta. Acto seguido, cinco ángulos, seis líneas, figura abierta. Por último, los estudiantes perciben que el punto de inicio no coincide con el último trazo y los hace dudar, reconocen que hay ahora 8 trazos.	FP y AB
min 20-min 55: Los estudiantes abren el archivo de la tarea 2, y con un nuevo bloque que permite repetir la secuencia las veces que ellos necesiten, se abocan al desafío de obtener los parámetros para obtener una figura cerrada.	AP
min 56-min: A continuación, los estudiantes organizan los bloques para cuatro secuencias y lograr generar un cuadrado, un rectángulo, un triángulo equilátero y un polígono regular de 5 o más lados.	AP y PL

Se esperaba que los estudiantes, sin ejecutar la aplicación, tenían que dibujar una poligonal con dos trazos y, luego, al pensar en repetir tres veces la secuencia, deberían haber dibujado una poligonal abierta, como se ilustra en la Figura 10, ya que el profesor no activo esa estrategia. En el caso de **E3**, como en la mayoría de los estudiantes, se limitó sólo a dibujar el tipo de figura que se formó al repetir tres veces la programación, como se aprecia en la Figura 11, reconociendo una figura abierta.

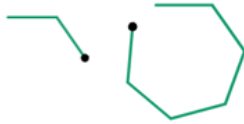


Figura 10. Secuencia de referencia para los estudiantes.

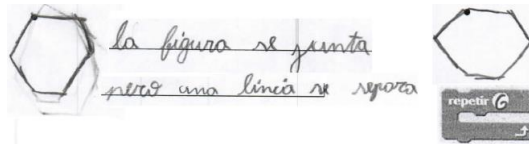


Figura 11. Respuesta del E3 al desafío cerrar la figura abierta que se planteó al inicio de la clase.
“La figura se junta pero una línea se separa”

Con base en la actividad anterior, en la Figura 12, se aprecian algunas de las secuencias de bloques que los estudiantes enviaron en un archivo (sin identificar con su nombre dicho archivo) en relación a las 4 figuras que se les solicitaron.

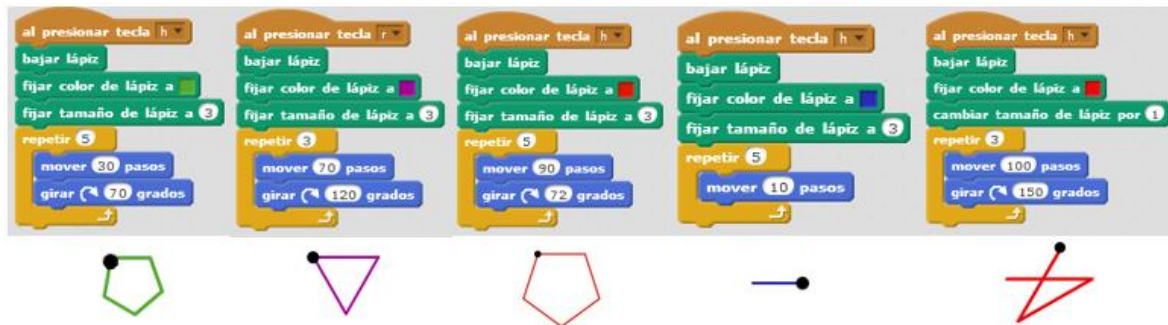


Figura 12. Respuesta de los estudiantes al desafío de generar una secuencia de bloques para dibujar un polígono.

Algunos aspectos a tener presente es que, para el caso de los pentágonos, la propuesta de 70° no es correcta, pero visualmente la figura se ve cerrada —probablemente por el tamaño del punto—. La propuesta de polígono de lado 10 tampoco logra dar con el ángulo interior. Otro aspecto que llama la atención que para el caso del triángulo equilátero no se relaciona el ángulo de 60° y 120° como base para lograr la figura. La secuencia de bloques para el rectángulo no se logró por los estudiantes, probablemente debido a que el bloque de repetir pudo ser un obstáculo a la hora de considerar un polígono de 4 lados, cuyos lados no todos son congruentes.

DISCUSIÓN

Sobre las fases de la TSD

Respecto de la decisión metodológica de considerar las tres primeras fases de la TSD, permitió que le profesor se enfocara en el rol de facilitador del trabajo de los estudiantes, sin perjuicio de que a ratos está la necesidad de dirigir el accionar de los estudiantes. La activación de la fase de formulación y validación, como se sugirió en el *codiseño*, fue condicionada, en parte, por la disposición de los estudiantes en torno a los computadores dispuestos en la sala de computación. Respecto de la opción de activar la fase de institucionalización, fue acertada, ya que los tiempos que requirió la tarea se redujeron, ya que el profesor, para el caso de la tarea 1, enfocarse en las instrucciones y la forma de ir registrando los datos. Para la tarea 2, decide recordar conceptos primitivos de la geometría plana e ideas relacionadas con el concepto de figura plana y de polígono.

Respecto de la Tarea 1

Un aspecto a destacar, en la implementación de la primera tarea, es que permitió a los estudiantes seguir rutinas al utilizar teclas de su computador para ingresar números, dibujar o borrar una figura, agregar datos a una lista dinámica, entre otras. Además, pudieron realizar sus propios registros, con lápiz y papel, en una guía de trabajo. Cabe indicar que algunas actitudes de los estudiantes —como la necesidad de recibir aprobación de su profesor a lo ya realizado o no prosperar más allá de las respuestas parciales a las que llegaron— influyó en la formulación y la validación de reglas (Tarea 1)

o en la obtención y verificación de aquellos parámetros para programar una figura poligonal regular (Tarea 2), en el sentido de lo que plantea Brousseau (2007).

Dar inicio a la clase utilizando la terna [1 2 3] —con la que se dibujó *remolino*—, ayudó a algunos estudiantes a entender uno de los objetivos de la Tarea 1, clasificar figuras según su forma. Aunque, en general, no pudieron comunicar reglas aritméticas, mediante lenguaje natural, algunos estudiantes recurrieron a algunas estrategias que tributaron a dicho objetivo. A modo de ejemplo, la estrategia de **E27** —invertir el orden de los números en la terna [1 2 3]— mantuvo *invariante* la forma *remolino*. Además, **E27**, agregó la terna [4 2 6], cuyos números son los múltiplos de dos de la terna [1 2 3]. Pese a ello, **E27** no hizo alusión a la idea de figura semejante. Cabe hacer notar que, dada la restricción de los números a ingresar, 1 a 6, no aparecieron otros múltiplos de la terna [1 2 3]. Invertir el orden de los números en una terna, como estrategia, permitió que otros estudiantes mostraran que una figura, distinta a *remolino*, también se mantiene invariante en su forma.

En el caso de **E6**, logró agregar la terna [5 1 6], distintas a las que propuso **E27**, pero tampoco pudo establecer una relación aritmética entre los números de una terna para *remolino*, probablemente faltó el espacio para focalizar dichos resultados en la gestión de la clase y, agregando otros ejemplos mediante la búsqueda de la aplicación, los estudiantes pudieron haber formulado alguna conjetura para *remolino*. Otro aporte de **E6** es haber propuesto ternas cuyos números son todos iguales, dejando invariante la forma de la figura, un cuadrado. Pese a ello, no logró expresar ese resultado como una conjetura mediante lenguaje natural.

Respecto de la Tarea 2

Los estudiantes, gracias a lo realizado en la Tarea 1, se familiarizaron prontamente con los requerimientos de la Tarea 2. Lograron comunicar aquellos elementos primarios de la poligonal, la que aumentaba, uno a uno, en el número de trazos, a medida que se ejecutaba la secuencia de bloques. En ese sentido, los estudiantes se enfocaron en describir aspectos como el aumento en el número de trazos y el número de ángulos. Además, se percataron que el punto de inicio no coincidía con el punto del último trazo, en una vuelta horaria. Con ello, se instaló de manera natural el problema de formar un polígono regular. Aquello, dejó de manifiesto dificultades de los estudiantes para coordinar, por ejemplo, los diferentes ángulos de giro con el ángulo de una vuelta completa, los 360° .

En cuanto a la secuenciación de bloques para obtener una figura poligonal, regular o no, si bien no hubo problema con la secuencia para el cuadrado, no pudieron dar con la secuencia de bloques para el rectángulo, pudiendo ser una barrera en dicha programación, el uso del *bloque repetir*. Por otro lado, como se evidenció en la Figura 12, la coordinación entre el número de ángulos de giro, los 360° y los divisores de este, dificultó que los estudiantes determinaran dicho ángulo de giro, como parámetro, para dibujar un pentágono y hexágono regular. Pese a que podían, mediante ensayo y error, aproximarse a la respuesta. Aunque, como se observa en la Figura 12, la primera figura, de izquierda a derecha, muestra que el tamaño del punto incidió en que se determinara el ángulo de giro 70° .

Sobre las habilidades de Pensamiento Computacional (PC)

En atención a la forma de proceder de los estudiantes, en cada una de las tareas, podemos indicar que pensar en casos específicos para una terna, cambiar el orden preestablecido de los números de una terna, verificar mediante la aplicación son aspectos que tributan con la *formulación de problemas* (FP), la *Abstracción* (AB) y el *Pensamiento Lógico* (PL). Por otro lado, la falta de ensayo y error, para lograr cerrar la figura poligonal o establecer el ángulo de giro correcto, son aspectos que imposibilitaron que se activará de manera eficiente el *Uso de Algoritmos o Programación* (AP) y *Análisis e Implementación de soluciones* (IS) y así lograr la *Generalización* (G).

CONCLUSIONES

Las dos tareas consideradas para esta investigación, permitieron estimular habilidades asociadas con el pensamiento computacional. Por otro lado, se observaron algunas *barreras o dificultades*, a tener

en cuenta a la hora del trabajo de un eventual *rediseño* del Codiseño, ya que pudieron tener un efecto negativo en los propósitos que se trazaron al implementar ambas tareas. Una primera barrera es que los estudiantes, en este tipo de tareas, que requieren de un grado de autonomía, se manifestó la necesidad de la aprobación del profesor para continuar con el trabajo que se les propuso. Por otro lado, otra barrera, fue que los estudiantes, dando con una respuesta, no prosperaron en ir más allá del simple requerimiento. En ese sentido, es probable que haya una forma de trabajo, en aulas escolares, que condiciona, en parte, ese trabajo autónomo que se requirió en esta investigación. Otro aspecto es que las fases de la TSD como estrategia didáctica es adecuada, siempre y cuando los estudiantes formulen sus conjeturas grupalmente, para que luego puedan ser socializadas con el resto de los grupos de trabajo, mediante el uso de la aplicación creada con SCRATCH para validar o refutar dichas conjeturas. Sobre el problema del ángulo de giro para programar el dibujo de un polígono regular o no, es una buena tarea para discutir con los estudiantes que la suma de los ángulos exteriores de un polígono es 360° , para luego avanzar a la idea que la suma de los ángulos interiores tiene una fórmula que se puede trabajar de manera inductiva y deductiva, pensando en la fase de institucionalización.

BIBLIOGRAFÍA

Aminah, N.; Sukestiyarno Y.L.; Cahyono, A.N. & Maat, S.M. (2023). Student activities in solving mathematics problems with a computational thinking using Scratch. *International Journal of Evaluation and Research in Education*, 12 (2), 613-621. <http://doi: 10.11591/ijere.v12i2.23308>

Arnon, I., Cottrill, J., & otros cinco autores (2014). *APOS theory: a framework for research and curriculum development in Mathematics Education*, Springer. <https://doi.org/10.1007/978-1-4614-7966-6>

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *Acm Inroads*, 2(1), 48-54. <https://doi.org/10.1145/1929887.1929905>

Basogain, X., & Olmedo, M. (2020). Integración de Pensamiento Computacional en Educación Básica. Dos Experiencias Pedagógicas de Aprendizaje Colaborativo online. *Revista de Educación a Distancia (RED)*, 20(63), 31-36. <https://doi.org/10.6018/red.409481>

Bassi, M.; Busso, M.; Urzúa, S. y Vargas, J. (2012). *Desconectados: habilidades, educación y empleo en América Latina*. Banco Interamericano de Desarrollo - Fondo de Cultura Económica: New York, Estados Unidos.

Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education - Implications for policy and practice*. Technical Report. Publications Office of the European Union. <https://doi.org/10.2791/792158>

Bocconi, S., Chiocciariello, A., Kampylis, P., Dagièné, V., Wastiau, P., Engelhardt, K., ... & Stupurienè, G. (2022). In A. Inamorato dos Santos, Y. Punie, R. Cachia, & N. Giannoutsou (Eds.), *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. Publications Office of the European Union. <https://data.europa.eu/doi/10.2760/126955>

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in

primary school. In E. Barendsen, & P. Hubwieser, (Eds.), *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65-72). <https://doi.org/10.1145/3137065>

Brousseau G (1997). *Theory of didactical situations in mathematics* (eds and trans: Balacheff N, Cooper M, Sutherland R, Warfield V). Kluwer.

Brousseau, G. (2007). *Iniciación al Estudio de las Situaciones Didácticas*. Libros del Zorzal: Buenos aires, Argentina.

Brousseau, G., Brousseau, N., & Warfield, V. (2014). *Teaching fractions through situations: A fundamental experiment*. Springer Science & Business Media. <https://doi.org/10.1007/978-94-007-2715-1>

Cabra Páez, M. L., & Ramírez Gamboa, S. A. (2022). Developing computational thinking and mathematical skills for analysis and problem solving: Learning on Scratch moodle platforms. *Revista Educación*, 46(1), 180-197. <http://dx.doi.org/10.15517/revedu.v46i1.44970>

Centro de Innovación del MINEDUC (2018). <https://www.innovacion.mineduc.cl/inicio>.

Couturier, R. (2008). CHIC: Cohesive Hierarchical Implicative Classification. *Studies in Computational Intelligence*, 41–53. https://doi.org/10.1007/978-3-540-78983-3_2

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In C. Schulte, & M.C, Caspersen (Coords.), *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 89-92). Association for Computing Machinery <https://doi.org/10.1145/2670757>

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832. <https://doi.org/10.1016/j.compedu.2020.103832>

Delors, J. (1996). de la publicación: La Educación Encierra un Tesoro. *Informe a la UNESCO de la Comisión Internacional sobre la Educación para el Siglo XXI*. *Laurus*, 14(26), 136-167.

Fagerlund, J.; Häkkinen, P.; Vesisenaho, M. & Viiri, J. (2020). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29 (1), 12-28. <http://doi.org/10.1002/cae.22255>

Felmer, P., Perdomo-Díaz, J., & Reyes, C. (2019). The ARPA experience in Chile: Problem solving for teachers' professional development. *Mathematical Problem Solving: Current Themes, Trends, and Research*, 311-337. https://doi.org/10.1007/978-3-030-10472-6_14

Gras, R., Suzuki, E., Guillet, F., & Spagnolo, F. (2008). *Statistical implicative analysis: theory and applications*. Springer: Berlín Heidelberg, Alemania.

Hargreaves, A. (2003). Enseñar en la sociedad del conocimiento. La educación en la era de la inventiva. *Perfiles Educativos*, 15 (102), 88-90.

Hazzan, O., Ragonis, N., Lapidot, T., Hazzan, O., Ragonis, N., & Lapidot, T. (2020). Problem-solving strategies. *Guide to Teaching Computer Science: An Activity-Based Approach*, 143-168. https://doi.org/10.1007/978-3-030-39360-1_8

- L'Heureux, J., Boisvert, D., Cohen, R. & Sanghera, K. (2012). It Problem Solving: An Implementation of Computational Thinking in Information Technology. *Proceedings of the 13th Annual Conference on Information Technology Education*, 183-188. ACM. <http://doi:10.1145/2380552.2380606>.
- Li, X., Xie, K., Vongkulluksn, V., Stein, D. & Zhang, Y. (2023). Developing and Testing a Design-Based Learning Approach to Enhance Elementary Students' Self-Perceived Computational Thinking. *Journal of Research on Technology in Education*, 55(2), 344-368, <http://doi:10.1080/15391523.2021.1962453>
- Liang, Z., Zhang, J., Wang, L., Qin, W., Lan, Y., Shao, J., & Zhang, X. (2021). *MWP-BERT: Numeracy-augmented pre-training for math word problem solving*. Findings of the Association for Computational Linguistics: NAACL 2022. <https://doi.org/10.18653/v1/2022.findings-naacl.74>
- Liljedahl, P., Santos-Trigo, M., Malaspina, U., & Bruder, R. (2016). *Problem solving in mathematics education*. Springer Nature. https://doi.org/10.1007/978-3-319-40730-2_1
- Lu, J. J. & Fletcher, G. H. L. (2009). Thinking About Computational Thinking. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 260-264. New York: ACM. <http://doi:10.1145/1508865.1508959>.
- Lubis, M., Khairiansyah, A., Jafar Adrian, Q., & Almaarif, A. (2019). Exploring the User Engagement Factors in Computer Mediated Communication. *Journal of Physics: Conference Series*, 1235(1), 012040. <https://doi.org/10.1088/1742-6596/1235/1/012040>
- MINEDUC (2018). Bases Curriculares de primero a sexto Básico. Unidad de Currículum.
- Moschella, M., & Basso, D. (2020). Computational thinking, spatial and logical skills. An investigation at primary school. *Ricerche di Pedagogia e Didattica. Journal of Theories and Research in Education*, 15(2), 69-89. <https://doi.org/10.6092/issn.1970-2221/11583>
- Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2, 100041. <https://doi.org/10.1016/j.caeai.2021.100041>
- Olivares, S. L., Adame, E., Treviño, J. I., López, M. V., & Turrubiates, M. L. (2020). Action learning: challenges that impact employability skills. *Higher Education, Skills and Work-Based Learning*, 10(1), 203-216. <https://doi.org/10.1108/heswbl-07-2019-0097>
- Orús, P.; Zamora, L. y Gregori, P. (2009). Teoría y aplicaciones del análisis estadístico implicative: primera aproximación en lengua hispana. Pilar Orús, Larisa Zamora, Pablo Gregori (Editores). Castellón: Universitat Jaume I, Departamento de Matemáticas; Santiago de Cuba: Facultad de Matemática y Computación. Universidad de Oriente de Santiago de Cuba, 2009. 516 p. ISBN 978-84-692-3925-4.
- Panskyi, T.; Rowinska, Z. & Biedron, S. (2019). Out-of-school assistance in the teaching of visual creative programming in the game-based environment – Case study: Poland. *Thinking Skills and Creativity*, 34, 1-14. <https://doi.org/10.1016/j.tsc.2019.100593>
- Partnership for 21st Century Skills. Framework for 21st Century Learning (2019). <https://bit.ly/3FS9JBC>

- Pellegrino, J. W. & Hilton, M. (eds.) (2012). *Education for life and work: developing transferable knowledge and Skills in the 21st Century*. National Academies Press.
- Plaza, P., Sancristobal, E., Carro, G., Castro, M., & Blazquez, M. (2018). Scratch day to introduce robotics. In *IEEE Global Engineering Education Conference (EDUCON)*, Santa Cruz de Tenerife (pp. 208-216). <http://doi.org/10.1109/EDUCON.2018.8363230>
- Pólya, G. (1990). *Cómo plantear y resolver problemas*. Trillas.
- Pólya, G. (1945). *How to solve it*. Princeton University.
- Qu, J. R., & Fok, P. K. (2021). Cultivating students' computational thinking through student–robot interactions in robotics education. *International Journal of Technology and Design Education*, 32(4), 1983–2002. <https://doi.org/10.1007/s10798-021-09677-3>
- Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017, March). Assessing computational thinking in CS unplugged activities. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 501-506). <https://doi.org/10.1145/3017680.3017779>
- Rodríguez, M., & Parraguez, M. (2014). Interpretando estrategias en Resolución de Problemas desde dos constructos teóricos: Un estudio de caso. *Revista electrónica de investigación en educación en ciencias*, 9(2), 1-12. <https://doi.org/10.54343/reiec.v9i2.175>
- Rodríguez, M.; Gregori, P.; Riveros, A. y Aceituno, A. (2017). Análisis de las estrategias de resolución de problemas en matemática utilizadas por estudiantes talentosos de 12 a 14 años. *Revista Educación Matemática*, 29(2), 159-186. <http://doi.org/10.24844/EM2902.06>
- Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316-327. <https://doi.org/10.1080/10494820.2019.1612448>
- Romero, M., Viéville, T., & Duflot-Kremer, M. (2022). *Activity for learning computational thinking in plugged and unplugged mode* (Doctoral dissertation, UCA-INSPE Académie de Nice).
- Selby, C. & Woollard, J. (2014). *Refining an understanding of computational thinking* (Monograph). University of Southampton
- Stake, R. E. (2010). *Qualitative research: studying how things work*. The Guilford Press.
- Stanic, G., & Kilpatrick, J. (1989). Historical perspectives on problem solving in the mathematics curriculum. *The teaching and assessing of mathematical problem solving*, 3, 1-22.
- Tonbuloğlu, B., & Tonbuloğlu, İ. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education*, 18(2), 403-426. <https://doi.org/10.15388/infedu.2019.19>
- Törner, G., Schoenfeld, A. H., & Reiss, K. M. (2007). Problem solving around the world: summing up the state of the art. *ZDM*, 39, 353-353. <https://doi.org/10.1007/s11858-007-0053-0>
- Treffers, A. (1987). Integrated column arithmetic according to progressive schematisation. *Educational studies in Mathematics*, 18(2), 125-145.
- Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017, October). Training computational thinking: Game-based unplugged and plugged-in activities in primary

school. In *European conference on games based learning* (pp. 687-695). Academic Conferences International Limited.

Ukobizaba, F., Nizeyimana, G., & Mukuka, A. (2021). Assessment Strategies for Enhancing Students' Mathematical Problem-Solving Skills: A Review of Literature. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(3), em1945.
<https://doi.org/10.29333/ejmste/9728>

Voskoglou, M. G. & Buckley, S. (2012). Problem Solving and Computational Thinking in a Learning Environment. *Egyptian Computer Science Journal*, 36(4), 28-46. <http://doi:arXiv:1212.0750v1>.

Wing, J.M. (2006). *Computational Thinking*. *Communications of the ACM*, 49(3), p. 33-35. <http://doi:10.1145/1118178.1118215>.

Wing, J. (2008). Computational Thinking and Thinking About Computing. *Philosophical Transactions of The Royal Society A*, 366, 3717-3725. <http://doi:10.1098/rsta.2008.0118>.

Yildiz, M., & Karal, H. (2021). A Computer Science Unplugged Activity: CityMap. *International Journal of Computer Science Education in Schools*, 5(2), 14-27. <https://doi.org/10.21585/ijcses.v5i2.110>

Yoshihara, K., & Watanabe, K. (2016, July). Practice of programming education using scratch and NanoBoardAG for high school students. In *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)* (pp. 567-568). IEEE.