

INTRODUCCIÓN AL PENSAMIENTO COMPUTACIONAL

Fernando Bordignon
Alejandro Iglesias

**PROYECTO —
PENSAMIENTO —
COMPUTACIONAL**

Créditos:

Autoridades EDUCAR S. E.

Lic. Laura Marés

Gerenta general de Educar S.E.

Lic. Mayra Botta

Gerenta de Proyectos TIC

Área TIC

Dirección del proyecto: María Laura Costilla

Gestión y desarrollo del proyecto:

Carolina Sokolowicz y Federico Frydman

Edición: Cristina Viturro

Corrección de estilo: Verónica Ruscio

Área de Diseño y Comunicación Visual

Responsable: DG Juan Furlino

Diseño de Marca: DG Manuel Vazquez

Diseño Editorial, tapa y diagramación:

DG Tamara Pereyra

Ilustraciones: DG Tamara Pereyra

Autoridades UNIPE

Adrián Cannellotto

Rector de la Universidad Pedagógica Nacional

Carlos Rodríguez

Vicerrector de la Universidad Pedagógica Nacional

Contenidos: Fernando Bordignon y Alejandro Iglesias

Equipo de colaboradores UNIPE

Lic. Javier Di Salvo

Lic. Mariano Fontao

Dra. Julia Pasín

Bordignon, Fernando

Introducción al pensamiento computacional / Fernando Bordignon ; Alejandro Adrián Iglesias. - 1a edición para el profesor - Ciudad Autónoma de Buenos Aires : UNIPE ; Editorial Universitaria ; EDUCAR S.E., 2020.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-3805-49-3

1. Tecnología de la Información y las Comunicaciones. 2. Educación Secundaria.
3. Tecnología Informática. I. Iglesias, Alejandro Adrián II. Título
CDD 004

ACERCA DE

Colección de tareas para el desarrollo del Pensamiento Computacional en estudiantes de nivel secundario

Este proyecto consistió en la edición y publicación de materiales didácticos diseñados por UNIPE orientados al desarrollo del Pensamiento Computacional en la escuela, con la intención de facilitar el abordaje de los NAP de educación digital, robótica y programación. A partir de la búsqueda y sistematización de experiencias en el nivel secundario relacionadas al Pensamiento Computacional (PC) aplicado a la resolución de problemas, el equipo de UNIPE diseñó y desarrolló materiales educativos orientados a docentes de la educación secundaria. El documento elaborado incluye una introducción al Pensamiento Computacional y a la enseñanza de problemas vinculada a los saberes digitales y una colección de catorce tareas que pueden ser realizadas en el aula con lápiz y papel, inspiradas en las actividades del concurso internacional Bebras. A partir del material elaborado por UNIPE y del trabajo conjunto con Educar S. E., se logró la producción de este sitio web y el libro digital sobre Pensamiento Computacional que lo sustenta.

CONTENIDO

INTRODUCCIÓN

07



CAPÍTULO 1

ENSEÑAR CON PROBLEMAS EN UN MUNDO DIGITAL

15

Sobre los problemas

15

Tipos de pensamiento aplicados a la resolución de problemas

18

La creatividad y los problemas en la escuela

21

CAPÍTULO 2

EL PENSAMIENTO COMPUTACIONAL, UN SABER DE ÉPOCA

24

Capacidades asociadas al pensamiento computacional

29

Técnicas asociadas con el pensamiento computacional

40

Consideraciones finales

46

CAPÍTULO 3

ACTIVIDADES PARA DESARROLLAR EL PENSAMIENTO COMPUTACIONAL

51

Sobre las actividades propuestas

51

Formato de las actividades

51

Metodología de trabajo

52

Organización temática

53

TAREAS

< /

T. 1 ¿Por dónde va? 55

T. 2 Camino a casa 58

T. 3 El escape 62

T. 4 Editando noticias 66

T. 5 Figuras de animales 70

T. 6 Brazaletes mágicos 74

T. 7 Arte con troncos 79

T. 8 Resultados revueltos 85

T. 9 Pociones mágicas 89

T.10 Invitados a la fiesta 93

T.11 Cargando los botes 97

T.12 Agentes secretos 101

T.13 El pintor de paredes 105

T.14 Caminando árboles 108

RESPUESTAS DESAFÍOS

113

REFERENCIAS BIBLIOGRÁFICAS

119

INTRODUCCIÓN

Estamos acostumbrados a estar rodeados de artefactos digitales, aplicaciones y todo tipo de *software* e incluso llevamos, todos los días, una pequeña computadora en el bolsillo que tiene por nombre *teléfono inteligente* o *smartphone*. Sin embargo, muchas personas desconocen cómo es que estos programas funcionan y cómo es que, muchas veces, estos aparatos nos ayudan a solucionar problemas.

Las computadoras son máquinas poderosas, capaces de ejecutar muchas operaciones, sin equivocarse, en muy poco tiempo. Pero, lamentablemente, las operaciones que pueden realizar son sencillas: sumas, restas, multiplicaciones, comparaciones y almacenamiento y recuperación de datos, entre otras. Es responsabilidad de las personas crear programas que puedan utilizarse para resolver problemas reales. Para lograrlo, primero se debe analizar la situación, dividirla en pequeñas partes y luego buscar una solución posible que pueda ser traducida a instrucciones que una computadora pueda ejecutar. Existe hoy una oferta amplia de herramientas que permiten desarrollar *software* con un alto nivel de abstracción y facilitan la tarea. De esta manera, las posibilidades de creación de *software* se expanden y se pueden desarrollar aplicaciones para realizar desde gestiones y operaciones bancarias en un cajero automático hasta simulaciones complejas de mundos virtuales en videojuegos.

Las habilidades necesarias para programar, como se ha hecho notar, no solo están enfocadas en entender cómo funciona una computadora y qué es capaz de hacer, sino también en desarrollar herramientas y técnicas para analizar los problemas y diseñar sus posibles soluciones. Estos conocimientos y habilidades a los que se hace referencia son las que se conocen hoy con el nombre de *pensamiento computacional* o PC, por su sigla.

Este pensamiento, entonces, no debe considerarse como una forma superadora de todos los métodos disponibles para solucionar problemas, sino como una forma complementaria, propia de este tiempo, que se puede utilizar más allá de las tecnologías.

Este documento está dirigido a docentes, en particular a aquellos del primer ciclo de la escuela secundaria, que estén interesados en acercarse a los conceptos asociados al pensamiento computacional y llevar su desarrollo al aula. Para ello se propone primero una reflexión acerca del trabajo escolar con problemas, luego se caracterizan las capacidades de este tipo de pensamiento y, finalmente, se ponen

a disposición una serie de tareas para realizar con estudiantes, que pueden ser resueltas usando solamente papel y lápiz, sin necesidad de conocimientos técnicos previos ni de una infraestructura tecnológica particular.

El trabajo propuesto plantea ir más allá de realizar una serie de ejercicios cerrados. El objetivo de estas tareas es que los estudiantes, reunidos en grupos, resuelvan las actividades elaborando sus propias estrategias y métodos de resolución. Esto les permitirá llegar a diferentes soluciones, que puedan ser argumentadas y discutidas, y, de esta manera, desarrollar capacidades relativas al pensamiento computacional. La labor del docente en esta propuesta será acompañar a los jóvenes en la resolución de estas tareas y, principalmente, generar instancias de reflexión que vinculen lo trabajado con los conceptos básicos del PC, como así también con otros contenidos de interés.

¡Bienvenidos y adelante!

CAP 1

ENSEÑAR CON PROBLEMAS EN UN MUNDO DIGITAL

“Como lo veo, el éxito en el futuro no estará basado en cuanto sepamos, sino en nuestra habilidad para pensar y actuar creativamente”

Mitchel Resnick

SOBRE LOS PROBLEMAS

Un problema es una situación en la cual se pretende llegar a una meta y, en función de lograrlo, se deben hallar y utilizar medios y estrategias. La mayoría de los problemas tienen una serie de elementos compartidos: un estado inicial, una meta a lograr, un conjunto de recursos, unas indicaciones en torno a lo que está permitido hacer y utilizar y lo que no, un dominio sobre el que se aplica y, por último experiencias de casos similares. En términos generales:

“un problema surge cuando existen obstáculos entre una situación dada y la situación a la que se quiere llegar, es querer encontrar un camino para poder llegar del estado actual al estado final, o al que se quiere obtener”

(Torres, 2011, p. 64).

Los problemas son situaciones que ubican a quien los resuelve ante la necesidad de desplegar su actividad cognitiva en un intento de búsqueda de estrategias, de elaboración de conjeturas y de toma de decisiones (Azcue, Diez, Lucanera y Scandrolí, 2006). Cuando se resuelven problemas desde una disciplina, se utilizan estrategias específicas de esta (por ejemplo, en problemas matemáticos se usan estrategias propias del área). Pero también se dispone de estrategias generales: una de ellas es la heurística, que se basa en el uso de reglas empíricas para llegar a una solución.

George Pólya fue un matemático destacado que en gran parte de su vida académica se dedicó a investigar en el área de resolución de problemas. Es considerado pionero en la temática por sus aportes, que se basan en una perspectiva global y para nada restringida a las matemáticas. Pólya enfocó la resolución de problemas desde un punto de vista que permitió plantear una serie de procedimientos que se aplican en la vida cotidiana:

Mi punto de vista es que la parte más importante de la forma de pensar que se desarrolla en matemática es la correcta actitud de la manera de acometer y tratar los problemas. Tenemos problemas en la vida diaria, en las ciencias, en la política, tenemos problemas por doquier. La actitud correcta en la forma de pensar puede ser ligeramente diferente de un dominio a otro, pero solo tenemos una cabeza y por lo tanto es natural que en definitiva haya solo un método de acometer toda clase de problemas. Mi opinión personal es que lo central en la enseñanza de la matemática es desarrollar tácticas en la resolución de problemas.

(Pólya, 1969).

Para Pólya (1945), la heurística es el área que trata de comprender el método que conduce a la solución de problemas y, en particular, se centra en las operaciones mentales útiles en este proceso. La heurística hace uso de la sistematización de la experiencia de resolver problemas a partir de cómo lo hacen los expertos. Para ello se propone una serie de cuatro pasos:

1. Comprender el problema. Reconocer qué se pregunta, identificar lo que hay que resolver y las condiciones asociadas.

2. Elaborar un plan. Se trata de establecer la vinculación entre los datos presentes y el problema a resolver, determinar los recursos que se utilizarán, verificar la similitud con otros problemas previamente resueltos y también la posibilidad de utilizar teorías o modelos útiles, todo esto en función de buscar una manera de resolver el problema.

3. Ejecutar el plan. Desarrollar el resultado de la respuesta, a partir de ejecutar el plan, avanzando y verificando cada paso.

4. Revisar y verificar la solución. Controlar qué hace y que dice el resultado, con vistas a considerar la posibilidad de transferir la solución a otros problemas.



Etapas del método Pólya de resolución de problemas.

Cada una de las etapas anteriores se ejecuta en forma de secuencia. A cada etapa, para su mejor comprensión, se le asocian una serie de preguntas orientadoras:

1. Comprender el problema.

Etapla en la que se identifica qué se pide de modo completamente independiente de las diversas condiciones impuestas y limitaciones asociadas al problema. En particular, se determina cuál es el objetivo de trabajo, los datos con que cuentan las condiciones, y la incógnita u objetivo de trabajo.

- › ¿Es claro el enunciado?
- › ¿Podés replantear el problema con tus palabras?
- › ¿Cuál es la incógnita o el objetivo del trabajo?
- › ¿Cuáles son los datos?
- › ¿Cuál es la condición?
- › ¿Hay suficiente información? ¿Está presentada de manera comprensible?
- › ¿Es suficiente la condición para determinar la incógnita?, ¿es insuficiente?, ¿es redundante?, ¿es contradictoria?
- › ¿Es un problema similar a algún otro que haya resuelto con anterioridad?

2. Elaborar un plan.

Se centra en el cómo o, en otras palabras, qué estrategia se va a utilizar a fin de resolver el problema. Las estrategias pueden plantearse a partir de pruebas de ensayo y error o hasta desarrollar una táctica que permita llegar a la solución. Por ejemplo, idear un plan en matemáticas, en general, se traduce en usar una ley, una definición o un principio que permita obtener la respuesta del problema. La etapa de planificación debe relacionarse con problemas parecidos, del mismo tipo, y también con resultados útiles. Se debe determinar si se pueden usar problemas similares o sus resultados (problemas análogos).

- › ¿Te has encontrado con un problema semejante?
- › ¿Has visto el mismo problema planteado en forma ligeramente diferente?
- › ¿Conocés algún problema relacionado?
- › ¿Conocés algún teorema que te pueda ser útil?
- › ¿Podrías enunciar el problema en otra forma?
- › ¿Podrías plantearlo en forma diferente nuevamente?

3. Ejecutar el plan.

Es la puesta en práctica de lo que el estudiante estableció en la etapa anterior. Se llevan adelante las etapas planteadas. Puede suceder que en el desarrollo se detecte que algo no es pertinente para la solución del problema, lo que implicará replantear la estrategia y volver a comenzar.

- ▶ ¿Se puede ver claramente que los pasos son correctos?
- ▶ ¿Es posible demostrarlo?

4. Revisar y verificar la solución.

Al final del trabajo, es necesario verificar el resultado obtenido y el proceso de solución desarrollado. Se realiza un abordaje metacognitivo.

- ▶ ¿Podés verificar el resultado?
- ▶ ¿Podés verificar el razonamiento?
- ▶ ¿Podés obtener el resultado en forma diferente?
- ▶ ¿Podés verlo fácilmente?
- ▶ ¿Podés emplear el método en algún otro problema?

También a esta etapa se la conoce como *visión retrospectiva*. Para construir conocimiento que sirva en el futuro, es necesario observar y reflexionar sobre qué fue lo que se hizo. Así se obtiene un canal de retroalimentación valioso, que ayudará a resolver problemas futuros. Por otro lado, una vez obtenido el método que lleve a una solución correcta, también se procura ver si hay otros métodos alternativos que respondan al mismo problema. De esta manera, se obtienen distintas perspectivas que pueden aplicarse en el futuro.

Ejemplo de aplicación del método de Pólya:

1. Comprender la situación:

Una persona compró [condición] medio kilogramo de carne [dato], tres cuartos de kilo de papas [dato] y un cuarto kilo de verduras [dato], ¿cuántos kilogramos transportó a su casa [incógnita]?

2. Concebir un plan:

Sumar lo que se compró y el resultado es la cantidad de kilogramos que la persona tuvo que transportar.

3. Ejecutar el plan:

$$(1 / 2) + (3 / 4) + (1 / 4) = 6 / 4 = 1,5 \text{ kg}$$

4. Revisar, verificar y comunicar:

La persona tuvo que transportar 1,5 kilogramos de comestibles hasta su casa.

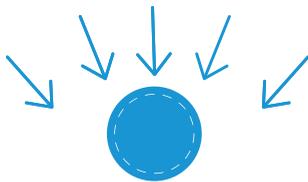
Tipos de pensamiento aplicados a la resolución de problemas

Joy Paul Guilford (1950) propuso un modelo de inteligencia que cambió el enfoque tradicional del análisis de la creatividad: distinguió entre el pensamiento convergente y el pensamiento divergente.

El primer tipo de pensamiento, el **convergente**, se caracteriza por ser analítico y racional, plantea que solo existe una solución correcta para cada problema. Es un pensamiento de tipo cerrado, dado que implica la restricción de las posibilidades y la producción de una única respuesta. Opera basándose en conocimientos previos; la información disponible se ordena de manera lógica para llegar a una solución que resuelve el problema planteado. Se podría decir que en este tipo de pensamiento no se construye el método para obtener esa respuesta, sino que se identifica la correcta (una suerte de reconocimiento de patrones de problemas). Esto implica que su aplicación recurrente no necesariamente garantizaría el desarrollo de la capacidad de resolución de problemas, sino más bien una ejercitación intensiva en detección de tipos de problemas y aplicación de soluciones prediseñadas.

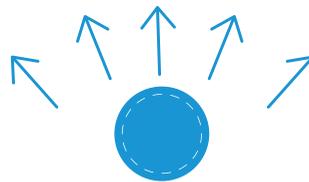
En cambio, el pensamiento **divergente** se caracteriza por mirar desde diferentes perspectivas y encontrar más de una solución frente a un problema. En esencia, contempla varias opciones de solución, que derivan en respuestas múltiples, que a la vez pueden ser todas correctas o no. La diferencia está dada en que en el proceso se flexibilizan posiciones y se producen nuevas conexiones. Por su apertura en la búsqueda de soluciones es una manera abierta de pensar.

CONVERGENTE



SOLUCIÓN

DIVERGENTE



PROBLEMA

Tipos de pensamiento aplicados a la resolución de problemas.

Para Guilford, el pensamiento divergente está formado por una parte importante de creatividad, la cual entendemos como la capacidad de pensar más allá de las ideas admitidas, combinando de una manera original conocimientos construidos, ya que este tipo de pensamiento fuerza la necesidad de generar ideas nuevas. Esto sucede a partir de que, a diferencia del pensamiento convergente, este no se basa en patrones preestablecidos, sino que implica idear un camino nuevo o diferente para resolver el problema. La creatividad implica evitar lo obvio, lo seguro, lo previsible para producir algo que, para su creador, resulte novedoso (Guilford, 1967).

Como explica Guilford, la creatividad no es algo que pocas personas tengan. Por el contrario, es una propiedad compartida por toda la humanidad y está presente

en cada individuo en mayor o menor grado. Está relacionada con aptitudes tales como la flexibilidad y la originalidad.

Más allá de las categorías cognitivas establecidas, en la vida real el pensamiento divergente y el pensamiento convergente se complementan y, con ayuda del método heurístico, contribuyen a desarrollar la capacidad para resolver problemas en las personas.

Edward de Bono es un estudioso de las formas de pensamiento aplicadas a la resolución de problemas que ha trabajado intensamente sobre el pensamiento divergente, al que se refiere llamándolo *pensamiento lateral*. En su definición indica:

"...es una actitud mental y también una cantidad de métodos definidos, la actitud mental implica la disponibilidad para tratar de mirar las cosas de diferentes maneras. Implica una apreciación de que cualquier manera de mirar las cosas es sólo una entre muchas. Implica una comprensión de cómo usa la mente los esquemas para poder pasar a otro mejor."

(De Bono, 1991, p.29)

En un proceso de resolución de problemas, la exploración, el descubrimiento y la comprensión de las estructuras y las relaciones de los elementos que forman parte del problema son una tarea creativa, en la cual se representa la realidad mediante abstracciones, utilizando modelos. Pero, para salir de la tarea única de reconocimiento de problemas anteriores y producir algo divergente, es decir, creativo, se hace necesaria una actitud crítica y activa, que promueva el trabajo desde diferentes perspectivas.

El pensamiento divergente parte de la base de aceptar que, en temas de enseñanza y de aprendizaje, más allá de las prácticas que giran en torno a la reproducción de saberes, también debe promoverse la construcción de estos. Desarrollar la resolución de problemas con heurísticas que contemplen ambos pensamientos es una forma de hacerlo.

A modo de síntesis, se puede decir que la creatividad no es simplemente una habilidad aislada del intelecto que sirve para crear cosas nuevas, sino que, por el contrario, surge como una sinergia entre la concepción que se tiene del mundo y su funcionamiento, los conocimientos adquiridos, una gran capacidad analítica y de observación, una percepción abierta del mundo, una actitud permeable que permita reestructurar y modificar construcciones mentales ya establecidas, la predisposición a la posibilidad de equivocarse y a aprender cosas nuevas, una actitud curiosa y comprometida y, por último, un interés en participar del cambio.

Desarrollar prácticas educativas que fomenten el pensamiento divergente y la resolución de problemas implica, por lo tanto, desarrollar también la creatividad y trabajar con todas las características mencionadas anteriormente.

David Bohm, por su parte, desde una posición crítica, coincide con Pekka Himanen (Himanen, 2002) en que estas actitudes son naturales en los niños y que los métodos tradicionales de aprendizaje trabajan en contra del desarrollo de esta capacidad. Y en particular se afirma en dos críticas fuertes: por un lado, declara que en la educación formal no se da espacio al error y esto es contrario a una actitud adecuada hacia este, en la que es importante correr riesgos y aprender de los experimentos y creaciones fallidas; por el otro, afirma que los métodos de aprendizajes memorísticos y rígidos donde se espera una única respuesta producen un aprendizaje anormal donde se forman estructuras mentales que no son propicias para ser reutilizadas en diversos contextos (Bohm, 2002).

La creatividad y los problemas en la escuela

El término creatividad ha aparecido varias veces en el texto ligado a las situaciones de resolución de problemas. Este hecho dota al concepto de una importancia particular ya que se la percibe como una actitud a desarrollar para lograr el desarrollo del pensamiento divergente. Por otro lado, es considerada actualmente una característica que servirá como herramienta a los estudiantes el día de mañana en su inserción en el ámbito laboral.

El profesor Mitchel Resnick, director del proyecto Lifelong Kindergarten en el MIT Media Lab, afirma:

En el mundo rápidamente cambiante de hoy, la gente debe continuamente crear soluciones creativas a problemas inesperados. El éxito se basa no solo en lo que tú sabes o cuánto sabes, sino en tu habilidad de pensar y actuar creativamente.

(Resnick 2007a)

Por este motivo, desarrollar una actitud creativa resulta de interés también para las instituciones educativas, donde no suelen ejercitarse estas aptitudes. El trabajo escolar se centra, por lo general, en resolver problemas específicos, por lo que luego resulta muy difícil poder aplicar y adaptar estas soluciones como respuesta a problemas inesperados o diferentes.

Así mismo, las nuevas tecnologías proporcionan una oportunidad para desarrollar tal actitud ya que permiten crear producciones en gran variedad de formatos (digitales o físicos) y proporcionan también plataformas digitales donde compar-

tirlas y seguir construyéndolas de manera colaborativa. En su trabajo con jóvenes y adolescentes en el MIT Media Lab, Resnick ha identificado, presenciado y destacado la importancia de jugar y explorar con herramientas digitales como una forma de promover una actitud creativa y activa (Resnick, 2007b). Por otro lado, Stager también destaca que las tecnologías digitales proporcionan un marco propicio para ejercitar esta actitud ya que son requeridas en todos los proyectos emprendidos en las prácticas *maker*¹ (Stager, 2014).

Más allá de los autores mencionados, es relativamente fácil encontrar bibliografía con temáticas vinculadas a la creatividad en el ámbito laboral. Esto se ve potenciado, en parte, por la velocidad de los cambios en el mundo actual que genera la necesidad de una continua adaptación, enfrentando a los ciudadanos a situaciones y oportunidades que no existían en el momento de su formación.

Los procesos de automatización han mejorado la capacidad productiva tanto de la industria como de la agricultura, han influido en las formas y en la cantidad de empleos disponibles, y se espera que su mayor impacto se produzca en las décadas venideras. Se prevé que, en los próximos 20 años, el 47 % de los empleos actuales desaparezcan a causa de la automatización, la inteligencia artificial y la robótica (Fresneda, 2016). Esto no significa, sin embargo, que existirán un 47 % menos de empleos disponibles, sino que las nuevas tecnologías darán paso a nuevos tipos de empleos que requerirán, a su vez, nuevas habilidades y mayor formación. Entre estas habilidades revalorizadas, que son transversales a todas las disciplinas, se destacan: el aprendizaje continuo, la predisposición para trabajar en equipo, la adaptación al cambio y, por sobre todo, la creatividad (WEF, 2016).

El informe *The future of jobs. Employment, skills and workforce strategy for the fourth industrial revolution* del Foro Económico Mundial indica que en el horizonte 2020 se prevén cambios disruptivos en el set de habilidades que los trabajadores necesitarán para poder integrarse en el mundo laboral. Incluso afirma que no será suficiente con adaptar o modernizar contenidos curriculares y cátedras universitarias ya que la vida útil de las habilidades requeridas para un mismo empleo es demasiado corta y los trabajadores deberán estar en un continuo estado de aprendizaje y adaptación. En este contexto, la capacidad de encontrar soluciones a nuevos problemas y de pensar de manera lateral y original no solo será necesaria para la diferenciación del ciudadano de cara al mercado laboral humano, sino también frente a posibles automatizaciones en torno a su empleo.

En este capítulo, se han presentado algunos conceptos básicos que deberían es-

¹ La expresión «prácticas maker» hace referencia a la forma de trabajo que se da en comunidades de personas cuyo interés radica en aprender a crear y construir cosas que los motivan y, a la vez, disfrutar de lo que hacen.

tar presentes en las aulas más a menudo y que hacen hincapié en las posibilidades educativas del hacer, entendido como una forma de construir conocimientos profundos que comienzan en las manos del aprendiz. En el siguiente capítulo, avanzaremos en el tema central de este libro y presentaremos el concepto denominado pensamiento computacional, como así también las capacidades implicadas en su desarrollo. Trataremos de llevar las reflexiones hacia la oportunidad de poder incorporar el PC en el aula con la finalidad de dar a los aprendices nuevas herramientas cognitivas que les permitan comprender y habitar problemas de una mejor manera.

EL PENSAMIENTO COMPUTACIONAL, UN SABER DE ÉPOCA

“A la lectura, escritura y aritmética, debemos agregar el pensamiento computacional en la habilidad analítica de cada niño”

Jeannette Wing

En los últimos años, el término *pensamiento computacional* (PC) ha ganado popularidad. Se lo utiliza para hacer referencia a técnicas y metodologías de resolución de problemas donde intervienen la experiencia y los saberes relacionados con la programación de computadoras. Su aplicación no solamente se restringe a problemas informáticos, sino que se puede utilizar de una manera más amplia, para razonar y trabajar sobre otros tipos de situaciones y áreas de conocimiento. En esencia, es una metodología de resolución de problemas que se puede automatizar (Zapata-Ros, 2015).

Según Jeannette Wing, promotora del concepto, el PC es una habilidad fundamental que debería ser desarrollada por todas las personas y no solo ser exclusiva de los profesionales de las ciencias de la computación.

A la lectura, escritura y aritmética, debemos agregar el pensamiento computacional en la habilidad analítica de cada niño. Así como la imprenta facilitó la difusión de la lectoescritura y el conocimiento matemático, [...] la computación y las computadoras facilitan la difusión del pensamiento computacional. El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática.

(Wing, 2006).

Este sentido que le da Wing se relaciona con la educación al plantear la cuestión de si el desarrollo de las habilidades relacionadas con el pensamiento computacional merece ser tenido en cuenta en los planes de estudio, en particular en los niveles de educación primaria y secundaria.

El PC puede entenderse como una forma alternativa de pensamiento relacionada con el surgimiento de las computadoras, dado que genera un ambiente cognitivo donde se juntan el pensamiento ingenieril, el científico y el lógico matemático (Wing, 2008). Esto implica desarrollar un mayor nivel de abstracción con el fin de resolver problemas concretos del mundo real. Se aplica, preferentemente, al diseño de sistemas (a través de operaciones de modelado) y a la solución de problemas por medio de su automatización en computadoras (pensamiento algorítmico y programación). El pensamiento computacional gira en torno a dos dimensiones: como vínculo entre varias formas de pensamiento (ingenieril, científico y lógico-matemático) sobre la base de utilizar mecanismos computacionales y como recurso orientado al fomento de la abstracción y el análisis de problemas. En el año 2010, Wing y otros autores actualizaron y dotaron de una precisión mayor a su definición de PC, al indicar:

"Pensamiento computacional son los procesos de pensamiento implicados en la formulación de problemas y sus soluciones para que estas últimas estén representadas de forma que puedan llevarse a cabo de manera efectiva por un procesador de información."

(Cuny, Snyder y Wing, 2010, como se citó en Wing, 2011, p. 20).

De la definición anterior surgen dos aspectos que están estrechamente ligados a la educación:

1. Este saber es un proceso de pensamiento y, por lo tanto, independiente de la tecnología.
2. El pensamiento computacional es un tipo específico de método de resolución de problemas.

Esto a su vez implica que las soluciones diseñadas pueden ser ejecutadas por una computadora, un agente humano o una combinación de ambos.

Una serie de trabajos (Villafañe, Rodríguez, Murazzo y Martínez, 2013; National Research Council, 2010 y National Research Council, 2011) afirman la importancia del pensamiento computacional como habilidad básica deseable de desarrollar en los estudiantes. Por ejemplo, para Doderó (2012), su desarrollo ayuda a derribar el mito de que las computadoras hacen magia y de que el informático es una suerte de mago que actúa de mediador entre los usuarios y una gran fuerza oculta. Por la razón anterior, su desarrollo no debería estar limitado a los técnicos, investigadores y profesionales de la informática, sino que cualquier estudiante debería aprender conceptos relacionados con el PC igual que como lo hizo con conocimientos básicos de matemáticas o física.

En una entrevista, el profesor Merelo (Johnbo, 2014) indica que, más que la programación, es necesario que los niños desarrollen el pensamiento computacional, una de cuyas tareas puede ser la programación. Además, recomienda que más allá de resolver problemas, hay que enseñarles una visión no pasiva de la informática. Es decir, que no se perciba solamente la informática como un producto de consumo (principalmente de entretenimiento), sino que se considere a la computadora como una máquina que hará cosas que los niños y jóvenes deseen y le ordenen.

Como se ha mencionado, el pensamiento computacional está sustentado en una serie de conceptos que provienen de las ciencias de la computación. Por ejemplo, para la *International Society for Technology in Education*² (ISTE, 2011) el PC es un proceso de solución de problemas que incluye las siguientes características, sin limitarse a ellas:

- ▶ formular problemas de una manera que permita usar computadoras y otras herramientas para trabajar en pos de su solución;
- ▶ organizar y analizar datos de forma lógica;
- ▶ representar datos de manera abstracta como modelos y simulaciones;
- ▶ automatizar soluciones mediante pensamiento algorítmico (sobre la base de una serie de pasos ordenados);
- ▶ identificar, analizar e implementar posibles soluciones con el objeto de encontrar la combinación de pasos y recursos de manera más eficiente y efectiva;
- ▶ generalizar y transferir ese proceso de solución de problemas a otros problemas.

Asimismo, las actividades que promueven el pensamiento computacional fomentan en las personas el desarrollo de una serie de destrezas particulares, entre las que se incluyen:

- ▶ confianza al trabajar con la complejidad,
- ▶ persistencia al trabajar con problemas difíciles,
- ▶ tolerancia a la ambigüedad,
- ▶ capacidad para lidiar con problemas abiertos y cerrados,
- ▶ capacidad para comunicarse y trabajar con otros para lograr una meta en común.

El profesor Resnick (Brennan y Resnick, 2012) ha propuesto un enfoque alternativo del pensamiento computacional que abarca tres dimensiones:

² La Sociedad Internacional para la Tecnología en Educación (conocida por su sigla en inglés ISTE) es una asociación de docentes que buscan promover el uso eficaz de la tecnología en la enseñanza y la formación de profesores. Su sede está en Washington, Estados Unidos. (www.iste.org).

- a) conceptos computacionales, que son aquellos que emplean los diseñadores en el trabajo de programación;
- b) prácticas computacionales, que son las que desarrollan a medida que programan,
- c) perspectivas computacionales, que son las que los diseñadores construyen sobre el mundo que los rodea y sobre sí mismos.

A partir de esta organización, se aportan maneras de valorar los aprendizajes generados por los jóvenes.

CONCEPTOS	PRÁCTICAS	PERSPECTIVAS
(que se emplean al programar)	(que se desarrollan al programar)	(que se forman sobre el mundo y sobre sí mismos)
<ul style="list-style-type: none"> . algoritmos . programas . secuencias . ciclos . operadores . eventos . paralelismo . datos y paradigmas entre los principales. 	<ul style="list-style-type: none"> . modelado . reutilización . abstracción . evaluación . modularización . documentación y metodología de desarrollo de proyectos. 	<ul style="list-style-type: none"> . expresión . creatividad . trabajo e interacción con pares . pensamiento crítico sobre las tecnologías, sus usos y contextos.

Dimensiones del PC según Resnick (Brennan y Resnick, 2012).

Desde el sector corporativo, también las grandes empresas de informática han puesto su mira en la promoción del pensamiento computacional. En octubre de 2010, Google puso en funcionamiento un portal web denominado *Exploración del Pensamiento Computacional*³, que reúne una importante cantidad de recursos sobre el tema, incluyendo material didáctico para maestros de primaria y secundaria. Por otro lado, Microsoft Research ha financiado el Centro para el Pensamiento Computacional⁴ en la Universidad de Carnegie Mellon, que apoya proyectos de investigación y de divulgación en ámbitos educativos.

En referencia a la enseñanza de la programación, el dilema de hoy parece estar en torno al *software* que se escribe versus la forma en que se piensa para resolver problemas. Raja (2014) indica que, si se empieza por enseñar el pensamiento computacional en vez de la programación, se puede evitar el principio de discriminación que hace que ciertos niños y niñas se inhiban, lo que supone un principio de democratización de los aprendizajes. Por otro lado, en el futuro esos niños, ya en

³ Google, sitio Exploración del Pensamiento Computacional. <http://www.google.com/edu/computationalthinking/index.html>

⁴ Centro para el Pensamiento Computacional. <http://www.cs.cmu.edu/~CompThink>

el rol de trabajadores, van a tener saberes y habilidades relacionadas con la forma de pensar y la resolución de problemas que los harán más competentes en el desarrollo de sus profesiones, cualesquiera que ellas sean. Pero para Raja (ibid) la manera en que hoy se enseñan los temas de ciencias de la computación en los establecimientos de educación secundaria está orientada a aspectos relacionados con la escritura de programas, lo cual refuerza la idea de que los lenguajes son solo para programadores, y en absoluto para artistas, médicos o bibliotecarios.

A fin de contraponer acciones a las realidades comentadas, la Fundación Sadosky, en la Argentina, lleva adelante el proyecto educativo de alcance nacional Program.Ar con el objetivo de enseñar conceptos básicos de las ciencias de la computación a niños y jóvenes. Desde la Fundación, se sostiene:

[La] formación del pensamiento computacional representa una actitud aplicable universalmente y un conjunto de habilidades requeridas actualmente por todos, incluyendo estudiantes y científicos de casi cualquier otra disciplina. Dado que las generaciones actuales se encuentran inmersas en la tecnología, deberíamos intentar enseñar a generaciones de jóvenes las razones detrás de estas nuevas tecnologías. Esto no es alcanzable sin enseñar ciencias de la computación.

(Fundación Sadosky, 2013)

Sin duda, el desarrollo de estas habilidades en nuestros estudiantes les permitirá sacar mejor partido de las ventajas de las transformaciones que los cambios tecnológicos están produciendo en nuestra sociedad y, de alguna manera, contribuirá a la solución de los grandes desafíos que se presentan en el presente. En este sentido, es importante que la capacidad de pensamiento computacional pueda ser desarrollada desde un período temprano de formación. Esto redundará en empoderar a los individuos a través de su desarrollo cognitivo y lograr autonomía para interactuar con el mundo que les toca vivir. En este contexto, el aprender a programar es considerado como un factor que promueve el desarrollo de una forma de pensamiento más abstracta, analítica y eficiente. En estos procesos de enseñanza y de aprendizaje la fuente de motivación es intrínseca y ayuda a promover la creatividad, en un ambiente donde prima el trabajo experimental a partir de la prueba y el error.

Como se puede observar, la temática abordada por el pensamiento computacional se extiende más allá de cualquier visión de orden reduccionista de la informática, que únicamente ve la programación como una técnica industrial de orden menor. En esencia, debemos promover el PC con el objetivo de mejorar la capacidad de entender y solucionar problemas que se puedan automatizar, entendiendo que el beneficio educativo de ser capaz de pensar con la ayuda de estas herramientas cognitivas se transfiere a otros dominios del conocimiento, y refuerza y mejora las habilidades intelectuales de las personas. (Wing, 2011).

A modo de resumen, se puede decir que el pensamiento computacional es un conjunto de saberes que merecen ser desarrollados por nuestros estudiantes, con varios fines:

- a) mejorar la habilidad para solucionar problemas y, como un plus, que estas soluciones puedan traducirse para ser ejecutadas por una computadora;
- b) que dejen de ser consumidores pasivos de tecnología y que pasen a tener una relación más estrecha, activa y fructífera con recursos y herramientas tecnológicas, con el objetivo de comprender mejor el mundo que los rodea;
- c) que los nuevos lenguajes informáticos les permitan ampliar sus posibilidades de expresión y poner en juego toda su capacidad creativa;
- d) que aquellos que ya tengan un principio de vocación por la ingeniería o las ciencias de la computación tengan experiencias de aprendizaje tempranas que les permitan empezar a involucrarse en lo que va a ser parte de su futuro.

Capacidades asociadas al pensamiento computacional

Existe cierto consenso en relación con que el pensamiento computacional va más allá de programar o codificar e implica todo un proceso previo, de formulación y análisis del problema, como así también de diseño y de evaluación de soluciones. Tal consenso empieza a tomar importancia.

En este sentido, entendiendo el pensamiento computacional como un proceso cognitivo que implica un razonamiento lógico aplicado a la resolución de problemas, sus elementos clave son los siguientes (CAS, 2015):

- ▶ capacidad de pensar de forma algorítmica,
- ▶ capacidad de pensar en términos de descomposición,
- ▶ capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones,
- ▶ capacidad de pensar en términos abstractos y elección de buenas representaciones,
- ▶ la capacidad de pensar en términos de evaluación.



Elementos clave del pensamiento computacional (CAS, 2015).

Los elementos clave del pensamiento computacional involucran el desarrollo de un razonamiento lógico. Este permite que los estudiantes puedan dar sentido a las cosas, lo que sucede por medio del análisis y la comprobación de los hechos a través de un pensamiento claro, detallado y preciso. De esta manera, los estudiantes toman sus propios conocimientos y modelos internos para hacer y verificar predicciones y así obtener conclusiones. El razonamiento lógico es la aplicación del pensamiento computacional para resolver problemas (CAS, 2015).

Por ejemplo, en actividades relacionadas con el diseño y la tecnología se aplica el razonamiento lógico —en particular, en el diseño de un objeto, la determinación de su forma y funcionalidad, la elección de los materiales a utilizar y los pasos de fabricación—. La descomposición se aplica al dividir un proyecto o proceso en diferentes partes según un criterio en particular. La generalización sucede cuando, dada una situación particular, el estudiante es capaz de establecer nuevas conexiones y pensar sobre otras aplicaciones o usos en otros contextos. Al diseñar, se están realizando tareas de predicción dado que se suponen comportamientos. El poder simular comportamientos implica evaluar situaciones futuras y así mejorar el diseño asegurando predicciones correctas. Cuando se escribe una solución a

un problema, para que pueda ser implementada por una persona o una computadora, hay tareas relacionadas con la búsqueda y corrección de errores: en esa instancia, también existe un proceso de razonamiento.

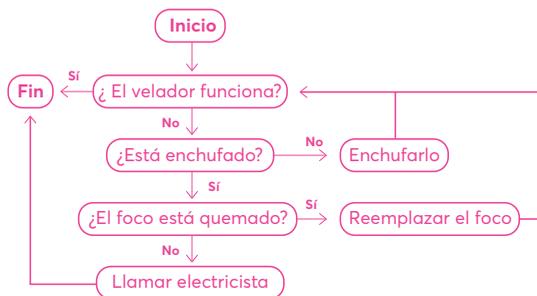
El razonamiento lógico ayuda a explicar por qué sucede algo. Esto es muy importante en ciencias de la computación debido a que las computadoras son predecibles en sus resultados, solo realizan aquello para lo cual están programadas. En virtud de esta cualidad, se utiliza el razonamiento lógico para programarlas y así describir con exactitud las tareas a realizar. Entendido de esta manera, el razonamiento lógico equivale a explicar por qué algo es así.

A continuación, se desarrollarán hasta el final del capítulo, de manera detallada, cada una de las capacidades básicas que componen el pensamiento computacional.

Capacidad de pensar de forma algorítmica

Un algoritmo, en principio, es un objeto de comunicación compuesto por un conjunto finito de instrucciones que especifican una secuencia de operaciones concretas por realizar en un orden determinado para resolver un problema. El pensamiento algorítmico es una actividad cognitiva asociada a la resolución de problemas, a su especificación y a la comunicación de su solución.

Los siguientes son ejemplos de algoritmos que expresan soluciones a distintos problemas. Nótese que el algoritmo puede expresarse de distintas formas, en estos casos como un gráfico y como un texto con órdenes.



ALGORITMO PARA PREPARAR UNA SOPA INSTANTÁNEA EN EL HORNO DE MICROONDAS

1. inicio;
2. destapar el envase de la sopa;
3. agregar una taza pequeña de agua a la sopa;
4. introducir en el horno microondas;
5. programar el horno de microondas por 3 minutos;
7. fin.

Ejemplos de algoritmos.

En general, el pensamiento algorítmico se aplica cuando existen problemas semejantes que tienen que ser resueltos con periodicidad, entonces se analizan en conjunto y se desarrolla una solución general que se aplica cada vez que ocurre el problema.

En nuestra vida cotidiana, recurrimos de manera constante a algoritmos para solucionar problemas y así realizar cosas. Por ejemplo: para resolver una cuenta y

obtener un valor, para cocinar una comida o para realizar una extracción de dinero en un cajero automático. En todos los casos mencionados, seguimos una y otra vez un conjunto ordenado de pasos que están almacenados en nuestro cerebro o en algún soporte externo (como en el caso de la receta de cocina que puede ser tomada de un libro o visualizada en YouTube).

Ejemplos de situaciones donde están presentes algoritmos:

- ▶ Cuando un cocinero escribe una receta para realizar un plato, está creando un algoritmo dado que otros pueden seguir los pasos y así reproducirla.
- ▶ Cuando un amigo anota las instrucciones para llegar a su casa, está especificando una secuencia de pasos (un algoritmo) para que otra persona lo pueda ubicar.
- ▶ Cuando un profesor proporciona un conjunto de instrucciones para llevar a cabo un experimento, está especificando un algoritmo, que es seguido por los estudiantes y así obtienen datos para su análisis y aprendizaje.

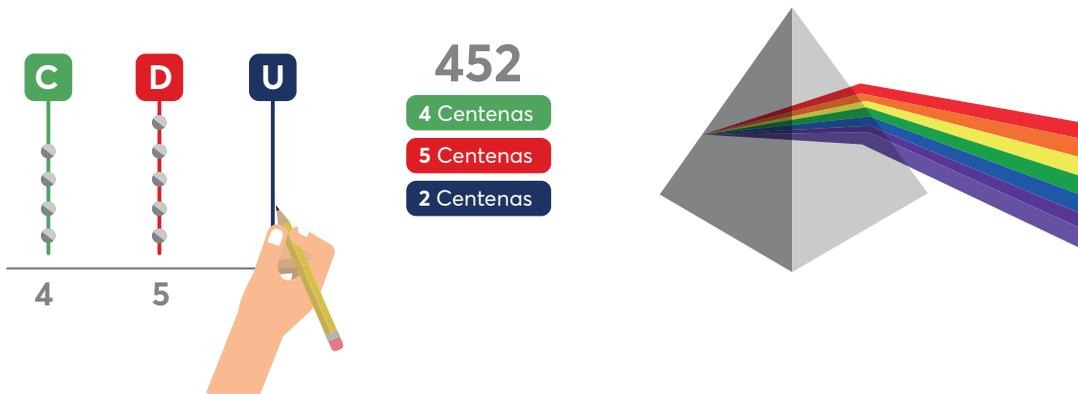
Podemos definir el pensamiento algorítmico como la capacidad de pensar en términos de secuencias y reglas que sirven para resolver problemas (CAS, 2015). Es un conocimiento básico que las personas desarrollan cuando aprenden a escribir sus propios programas de computadora, que no son otra cosa que algoritmos traducidos a instrucciones expresadas en un lenguaje que una computadora pueda comprender y ejecutar (por ejemplo, lenguajes informáticos como Scratch, Python, JavaScript, etc.).

Cuando se habla de software, se hace referencia directa a los datos digitales y a los programas de computadora. Lev Manovich, autor del libro *El software toma el mando*, realiza una descripción de su poder y su presencia en la sociedad actual cuando dice: «El *software* se ha vuelto nuestra interfaz con el mundo, con otras personas, con nuestra memoria e imaginación; un lenguaje universal mediante el cual habla el mundo, un motor universal mediante el cual funciona el mundo» (Manovich, 2013) y, por otro lado, nos advierte: «El *software* juega, hoy en día, un papel crucial en la confección de elementos materiales y de estructuras inmateriales que, aunados, constituyen la “cultura”».

En las ciencias de la computación, el trabajo de los científicos se concentra en encontrar los algoritmos más eficientes. Es decir, aquellos que resuelven un problema involucrando los menores recursos posibles (memoria, comunicaciones, tiempo de procesamiento, etc.) de la manera más efectiva al dar la respuesta correcta o la más cercana a ella.

Capacidad de pensar en términos de descomposición

Según el *Diccionario de la lengua española* de la Real Academia Española, descomponer significa 'separar las diversas partes que forman un compuesto', entre otras acepciones. En los siguientes gráficos, pueden observarse distintas situaciones en las cuales sucede un proceso de descomposición: una es la descomposición numérica —que ayuda a que los estudiantes entiendan la disposición y las relaciones entre los dígitos de un número— y otra, la descomposición de la luz con un prisma —para obtener el espectro que representa al arcoíris—.



Ejemplos de descomposición.

En la escuela es habitual que nos encontremos con actividades de cierta complejidad que tienen que ser descompuestas en tareas más simples para que puedan llevarse a cabo. Algunos ejemplos:

- ▶ Preparar la fiesta de fin de año. Tareas: armado del programa de actividades, difusión del evento, preparación y ensayo de las actividades artísticas, preparación y atención del servicio de cantina, preparación y limpieza del salón de actos, etc.
- ▶ Desarrollar un plan que permita asignarle el rótulo de ecológica a una institución educativa. Tareas: definir una estrategia para almacenar y procesar la basura, establecer un plan para reducir el consumo de electricidad, planificar una campaña de concientización en la comunidad escolar.
- ▶ Planificar la publicación de una revista institucional. Tareas: identificar temas y secciones, asignar roles de los colaboradores y responsabilidades asociadas, planificar tiempos y recursos necesarios a los efectos de llevar adelante el proyecto.

En el ámbito informático, al proceso de dividir un problema en partes más pequeñas o sencillas (y, por ende, más manejables) se lo conoce como **descomposición**. Descomponer un problema es una tarea que facilita su resolución debido a que reduce su complejidad.

Para descomponer hay que pensar en términos de partes y componentes, donde cada pieza se debe comprender, evaluar y solucionar por separado. Por otro lado, la solución asociada a cada una de las partes puede encargarse a una persona o a un equipo de trabajo, con lo cual esto permite poder resolver problemas complejos en tiempos más acotados (Csizmadia et al., 2015).

En esencia, toda descomposición implica:

- ▶ identificación de las partes de algo,
- ▶ división de algo en partes más pequeñas.

Por ejemplo, una actividad en la que se suele aplicar la técnica de la descomposición es la que los detectives policiales realizan ante un crimen a los efectos de esclarecerlo ya que es un proceso complejo, las tareas asociadas son muchas y, en general, son llevadas a cabo por un equipo de agentes. En general, la técnica policial ante un hecho delictivo divide su hacer relacionado con la investigación a partir de una serie de preguntas clave (las cuales se podrían ver como subproblemas): ¿qué tipo de crimen se ha cometido?, ¿cuándo se llevó a cabo?, ¿dónde se cometió?, ¿qué evidencias hay?, ¿hubo testigos?, ¿qué observaron?, ¿quién era la víctima?, ¿hay semejanza con otros crímenes registrados? Luego de obtener las respuestas, el equipo policial puede evaluarlas en su conjunto y empezar a pensar en líneas de investigación a partir de indicios concretos. Como se observa en el ejemplo, la descomposición del problema y la generación de tareas de menor complejidad permiten empezar una resolución efectiva y lógica del problema.

También se puede producir un proceso de descomposición en un proyecto de trabajo: por ejemplo, en la creación de un videojuego en torno al tema de cuidados en el uso de redes sociales. Para facilitar la solución del problema, el equipo responsable realiza un primer análisis y evaluación y luego decide usar esta técnica. Se identifican una serie de partes principales y cada una se asigna a un equipo técnico:

- a) diseño de la narrativa, lógica y reglas asociadas al juego;
- b) diseño de los personajes, objetos y escenarios;
- c) programación del videojuego;
- d) depuración y búsqueda de errores del producto videojuego.

Luego de analizar la situación y de que el gran problema haya podido dividirse en partes de menor complejidad, se asigna la resolución de cada una a una persona

o grupo. A lo largo del desarrollo se ensamblan los resultados de cada equipo en pos de dar solución a la situación problema.

Capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones

Para tratar de pensar el concepto de reconocimiento de patrones, supongamos la siguiente situación: una persona está a cargo de un campo con animales y existe comida suficiente y especial para cada uno. Las instrucciones para que un operario alimente a los animales son simples:

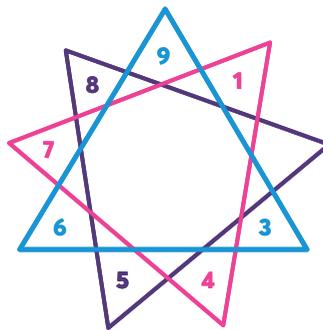
- ▶ Para alimentar al perro, poner la comida del perro en el plato del perro;
- ▶ Para alimentar al pollo, poner la comida del pollo en el plato del pollo;
- ▶ Para alimentar al conejo, poner la comida del conejo en el plato del conejo.

Nótese que hay una estructura subyacente común en cada una de las instrucciones anteriores, es decir un patrón, que podría expresarse de la siguiente manera:

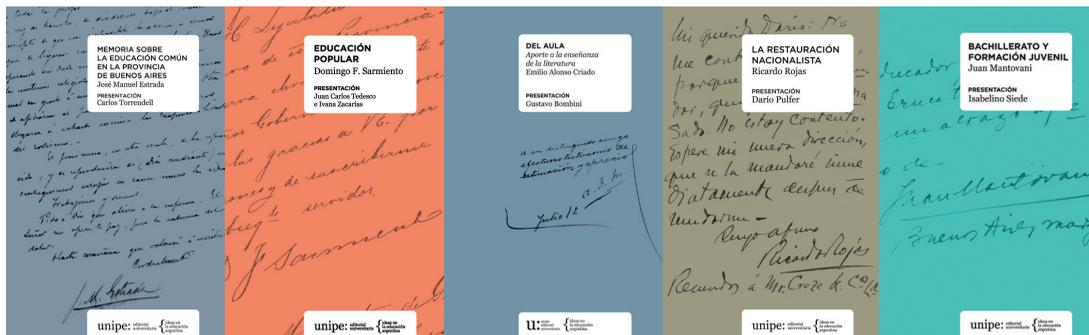
- ▶ Para alimentar al <animal>, poner la comida del <animal> en el plato del <animal>.

Al detectar un patrón en las instrucciones de alimentación de los animales, se pudo realizar luego un proceso de **generalización** que simplificó el protocolo de alimentación a partir de una única instrucción genérica.

Ahora vamos a otro caso: observá la siguiente figura y tratá de detectar cuál es el patrón que generó el dibujo.



Podríamos decir que, a partir de un triángulo equilátero al que se hizo girar dos veces (n grados en cada vez), se obtuvo la imagen anterior, por lo cual el patrón es la rotación del triángulo original n grados.



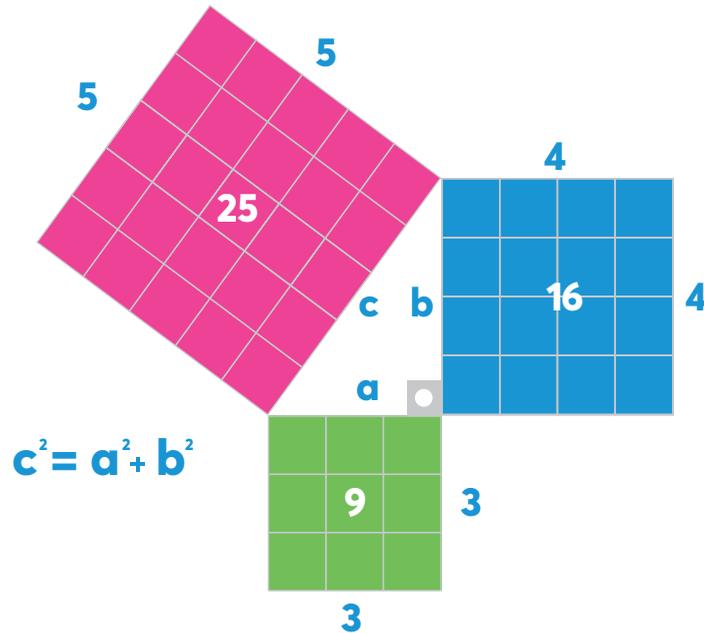
Tapas de la colección de libros Ideas para la Educación Argentina, de UNiPE. ¿Se visualiza algún patrón para identificar?

En la práctica, al **descomponer** un problema complejo se suelen encontrar **patrones** entre los subproblemas que fueron definidos. Los patrones se expresan como características compartidas entre los distintos problemas de menor complejidad. Cuando se los detecta, es posible trabajarlos de manera conjunta y así simplificar la tarea de resolución. Los problemas son más fáciles de resolver cuando comparten patrones, debido a que es posible usar soluciones ya diseñadas con anterioridad y aplicarlas a los subproblemas.

Ejemplos de patrones:

- ▶ Se buscan patrones respecto de la cantidad de personas y el tiempo que demoran al ingresar en un banco y elegir una cola para realizar un pago.
- ▶ Los conductores buscan patrones de comportamiento en el tráfico para decidir cuándo cambiar de carril e ir más rápido.
- ▶ Los inversores buscan patrones en los precios de las acciones para decidir cuándo comprar y cuándo vender.

La generalización es una tarea relacionada con la identificación de patrones, semejanzas y conexiones, para luego realizar una explotación de las características. Se presenta como un método rápido para resolver nuevos problemas sobre la base de clasificarlos como versiones de viejos problemas con solución. Es decir, que se hace uso de la experiencia (Csizmadia et al., 2015). En general, la generalización es una tarea que se realiza luego de la descomposición.



La fórmula matemática del teorema de Pitágoras es un ejemplo de generalización.

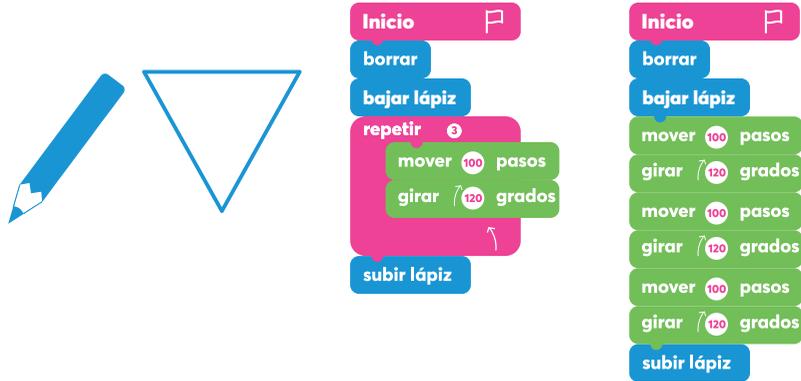
La descomposición y la generalización están relacionadas. Mientras que la descomposición implica dividir un problema en partes más pequeñas, la generalización se basa en combinar esas partes. Así, cuando se realiza la generalización a partir de las partes individuales, no se vuelve a unir el problema de la misma manera en que estaba originalmente. El objetivo de la generalización es observar las partes descompuestas y encontrar formas de facilitar el desarrollo de una solución.

Otro ejemplo de generalización aplicada al desarrollo de programas es el siguiente: supóné que un estudiante tiene que realizar un algoritmo y luego un programa que dibuje un triángulo equilátero.

Una manera de plantear una solución es el algoritmo (escrito en código de bloques del lenguaje Scratch) que se observa en la imagen siguiente a la derecha. Luego de iniciar la tarea, se toma una hoja de trabajo limpia y la herramienta **Lápiz**. Se desplaza la herramienta 100 unidades de longitud (en nuestro caso, son pasos y cada uno tiene una medida fija determinada) y luego se la hace girar 120 grados en el sentido horario sobre su eje central. De esta manera, se ha dibujado un lado y todo está preparado para dibujar el siguiente. Como se ve en la imagen, luego se repiten dos veces los pares de instrucciones mover y girar. Con eso se completan los tres lados y el algoritmo finaliza con la instrucción de levantar el lápiz.

A la izquierda del algoritmo anterior puede observarse otra solución al problema (y más eficiente en términos de redacción de la solución), donde se ha identifica-

do que el par de instrucciones mover y girar son en sí un patrón y, en este caso, se repiten (una vez por cada lado del triángulo). Para hacer más eficiente la solución, cometer menos errores de escritura y comunicar mejor la tarea se utiliza la instrucción repetir 3 y, en cada ciclo de repetición, se solicita el dibujo de una cara. En este caso, a partir de detectar el patrón lado, se ha podido hacer una generalización simple y así obtener un algoritmo más legible, optimizado en su escritura y seguramente con menos errores.

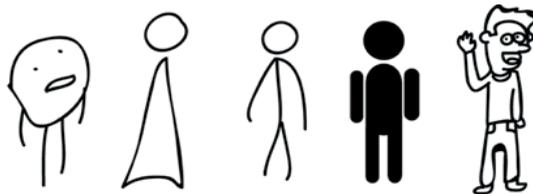


Algoritmos que responden a la solución de dibujar un triángulo equilátero.

Los científicos de las ciencias de la computación desean resolver problemas de forma rápida y eficiente, para lo cual suelen reutilizar métodos efectivos creados con anterioridad. Los patrones son elementos comunes en problemas, se identifican a efectos de crear módulos estándar para que se puedan reutilizar en distintos programas para solucionar diferentes problemas. En general, a partir de un patrón detectado se construye un módulo común una sola vez. En términos asociados a los lenguajes de programación, a estos módulos se los conoce como *procedimientos* o *funciones*.

Capacidad de pensar en términos abstractos, la elección de buenas representaciones

En principio, la **abstracción** es un proceso por el cual se simplifica el entendimiento de una situación. Se basa en identificar lo que es importante de algo, sin preocuparse por los detalles, así, de esta manera, se puede administrar la complejidad de un problema. Todo proceso de abstracción da como resultado la construcción de una vista simplificada, que es la idea principal de algo.



Diferentes representaciones de un ser humano.

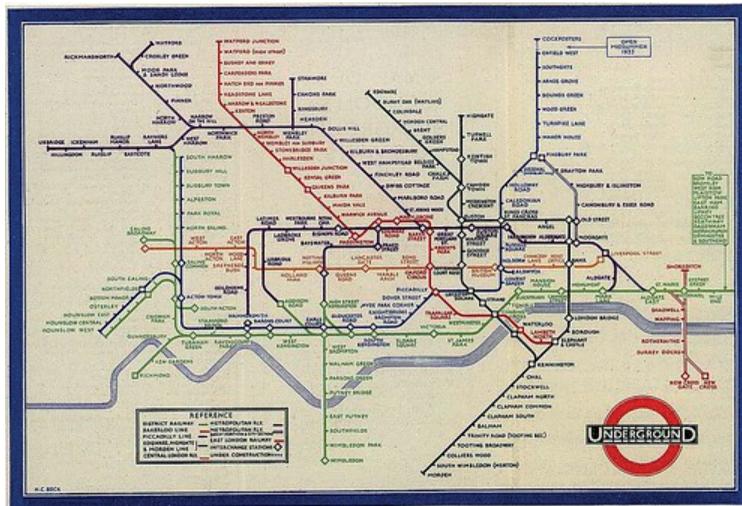
El ser humano ha utilizado la abstracción de muchas maneras al retratarse a sí mismo. En la imagen superior podemos ver un ejemplo de ello. De izquierda a derecha, el primero es un dibujo clásico de un niño de 3 o 4 años, su primer retrato; luego, aparece el bosquejo utilizado en planos de arquitectura para dar idea de las dimensiones en relación a una persona; en tercer lugar, un monigote típico que se puede observar en muchos esquemas; en el cuarto, la silueta de un hombre que a menudo se usa en señalética; finalmente una caricatura que tiene más detalles a fin de comunicar mejor los sentimientos de los personajes. Todas ellas son representaciones diferentes del mismo objeto y son abstracciones.

Jeannette Wing (2006) indica: «en el proceso de abstracción, decidir qué detalles debemos resaltar y qué detalles podemos ignorar, subyace en el pensamiento computacional». Teniendo en cuenta que la idea principal de todo no está en los detalles, sino en los grandes rasgos, al abstraer algo se obtiene como resultado una vista más comprensible. La base del trabajo de abstracción consiste en la observación, la detección y la reducción de detalles, al ocultar cosas no relevantes.

Diferentes puntos de vista ofrecen diferentes abstracciones. Por ejemplo, los mapas mundiales solo muestran las características de importancia para un observador particular. Un mapa para asistir a viajeros muestra las rutas, las distancias y las ciudades de una región. Un mapa de distribución de cultivos, realizado para inversores, realza sobre un área específica las áreas asignadas a la producción de cada cultivo con un color asociado e indica el rendimiento. Pero un mapa de esa misma región destinado a ecólogos podría indicar la distribución de cultivos y los riesgos asociados en relación con el uso de agroquímicos.

Como se ha visto, si bien sobre un mismo problema se pueden generar distintas abstracciones, lo importante para avanzar en su solución es realizar la elección de una buena representación (Csizmadia et al., 2015). Es necesario tener en cuenta que las diferentes representaciones sobre un mismo problema pueden hacerlas cosas más fáciles en algunos casos y más difíciles en otros.

Otro ejemplo clásico de abstracción son los mapas de los servicios de subterráneos de una ciudad. En el año 1931, Harry Beck creó un mapa original de la red de metro de Londres. La característica de este objeto gráfico estaba dada porque solo se mostraban los recorridos de las líneas de trenes, las curvas del río Támesis y, además, las distancias relativas entre las estaciones. Este producto fue realmente innovador dado que su esquema abstracto de trazado sirvió muy bien a la comunicación, tanto que luego siguió siendo utilizado por bastantes décadas. Según cuenta la historia, este producto fue uno entre muchos de los que hizo Beck para resolver el problema de comunicar de forma sencilla el sistema de subterráneos londinense. En el proceso de desarrollo, el nivel de abstracción utilizado fue seleccionado cuidadosamente con la finalidad de incluir solo los detalles necesarios (es decir, eliminando los innecesarios) ya que, si era muy detallado, podía resultar confuso y, si era por demás simplificado, podía obviar información útil.



Mapa de Beck de las líneas de subterráneos de Londres, versión de 1931.

Un proceso de abstracción permite definir la esencia de algo. En la educación, la abstracción suele ser utilizada porque permite mostrar con ejemplos simples sistemas complejos para su estudio y comprensión. Por eso, se aplica en infografías, maquetas, diseños conceptuales, etc.

Otros ejemplos de procesos de abstracción:

- ▶ Una hoja de planificación diaria utiliza la abstracción para representar una semana en términos de días y horas, lo cual la convierte en un objeto útil para organizar el tiempo personal.
- ▶ Un mapamundi es una abstracción del planeta. Su composición, en términos de longitud y latitud, ayuda a describir la ubicación y la geografía de un lugar particular.
- ▶ En matemáticas, se escriben fórmulas generales en términos de variables en lugar de utilizar números. Esto sucede para que puedan ser utilizadas en problemas que involucren diferentes valores.
- ▶ En música, los pentagramas son soportes de escritura de piezas musicales y se los puede entender como abstracciones que representan sonidos en el tiempo.



La notación musical permite escribir de forma abstracta los sonidos de una pieza de música.

Capacidad de pensar en términos de evaluación

Una tarea de evaluación implica hacer juicios sobre algo, de una manera sistemática y sobre la base de criterios previamente definidos para que esta sea lo más objetiva posible. La evaluación es algo que se realiza diariamente: de manera regular hacemos juicios sobre qué hacer, pensando en función de una serie de factores que son parte de un contexto. Por ejemplo, cuando alguien va a comprar un automóvil, en general, antes de hacerlo piensa cuán comfortable es, en qué medida es económico, si es caro o barato su mantenimiento, etc., es decir, evalúa aspectos esenciales antes de decidir si lo adquiere o no.

También evaluamos algo en estas situaciones:

- ▶ Cuando cocinamos, probamos nuestros platos para ajustar su sabor y verificar el estado de cocción.
- ▶ Cuando estamos en un nivel de un videojuego y pretendemos pasar al siguiente, probamos y evaluamos distintas estrategias de acción que nos permitan seguir adelante.
- ▶ Cuando recibimos una devolución de un docente de un trabajo estudiantil, revisamos las notas añadidas para saber qué estamos haciendo bien y qué cosas aún nos falta aprender o hemos comprendido de forma errónea.

En el marco del pensamiento computacional, una vez que se ha diseñado una solución, es necesario asegurarse de que sea adecuada para su propósito. La evaluación es el proceso que se aplica a una respuesta en pos de asegurar que esta

responde a los requerimientos de diseño y que, además, funciona correctamente, sin errores. Cuando se trabaja con programas de computadora, la evaluación es una tarea sistemática y rigurosa ya que se está juzgando su efectividad y eficiencia.

Cuando una solución se traduce a un algoritmo, deberían verificarse una serie de elementos que son parte de una prueba de evaluación:

- ▶ Que se entienda fácilmente: ¿está completamente descompuesto?
- ▶ Que sea eficaz: ¿resuelve el problema?
- ▶ Que sea eficiente: ¿resuelve el problema, haciendo el mejor uso posible de los recursos disponibles?
- ▶ Que cumpla con los criterios de diseño especificados: ¿está en el marco de los requerimientos?

Luego de que un algoritmo supera el proceso de evaluación y cumple con los cuatro criterios anteriores, se entiende que se está ante una solución correcta; por lo tanto, se puede avanzar a la etapa de programación en algún lenguaje informático. La mayoría de las veces programar sin antes evaluar dificulta la tarea de programar, lo cual puede llevar a más errores y, por ende, a mayores costos y tiempos en el proyecto.



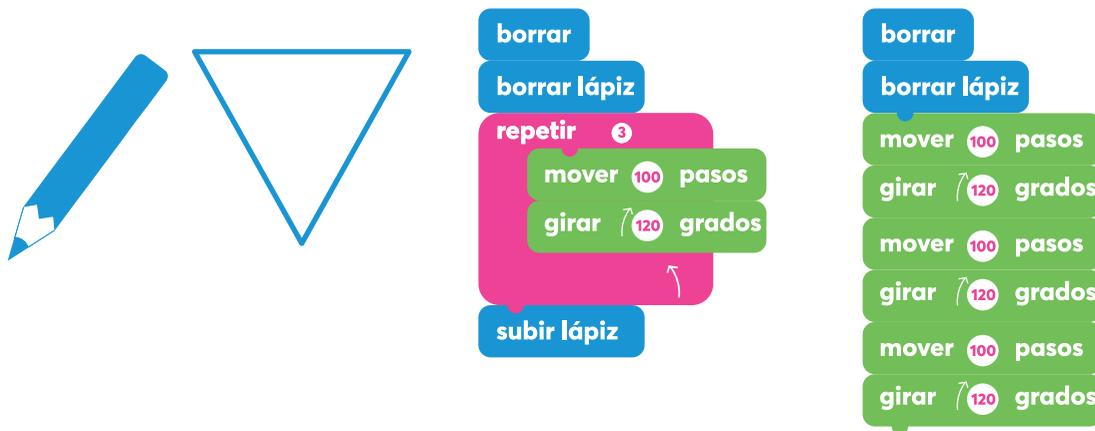
Preguntas típicas para evaluar una solución.

¿Cómo evaluamos nuestra solución? Como hay muchos métodos desarrollados, en principio y a modo de ejemplo, se podría comenzar la tarea con la orientación de las siguientes tres preguntas:

1. ¿Se comprende, de manera completa, cómo se ha resuelto el problema? Es decir, ¿la solución construida está explicitada en su totalidad? Si todavía algo no se sabe, aún no se tiene una solución completa. En caso de que no

se sepa claramente cómo hacer algo para resolver el problema, hay que volver a la etapa anterior. Es necesario verificar que todo se haya descompuesto correctamente y que cada parte tenga una solución.

2. ¿La solución cubre todas las partes del problema? Aquí se busca validar que la solución propuesta y desarrollada satisfaga plenamente el objetivo a cumplir y que, además, lo haga en el marco de las restricciones impuestas.
3. ¿La solución optimiza la repetición de tareas? En caso de respuesta negativa, se debe preguntar si existe alguna forma de reducir tal repetición, para lo cual hay que regresar a la etapa de desarrollo de la solución y eliminar las repeticiones innecesarias.



Algoritmo optimizado que utiliza un procedimiento llamado figura.

En ciencias de la computación, se conoce como **depuración** o **debugging** el proceso de búsqueda y corrección de errores en un programa. Es una tarea de cierta complejidad y tiempo debido a que primero hay que entender el problema y la solución elegida para luego trabajar analizando las instrucciones que la conforman.

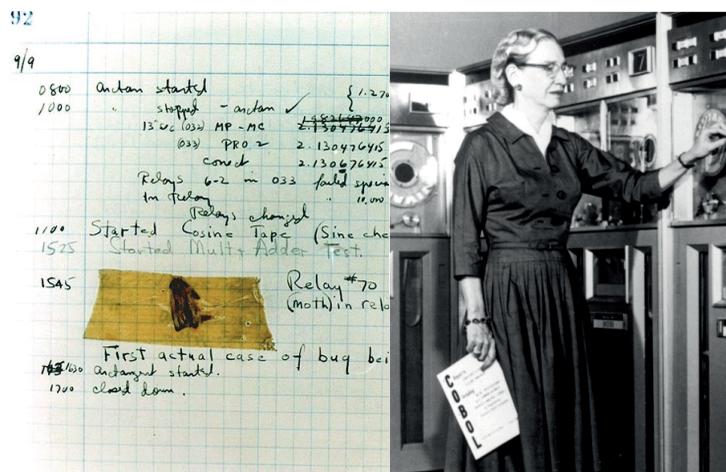
Para corregir errores o *bugs* en programas de computadora, en principio, se sugiere seguir una secuencia de tres pasos:

- ▶ predecir lo que debe suceder;
- ▶ averiguar exactamente qué sucede;
- ▶ trabajar nuevamente donde algo salió mal hasta arreglarlo.

En un programa de computadora suele haber dos tipos de errores: los de codificación (se escribió mal alguna instrucción, por ejemplo, se usó la orden INPRIMIR en

vez IMPRIMIR) y los de lógica (el plan asociado a la solución está equivocado).

Para finalizar con el tema de la evaluación, la historia del primer *debugging*. Fue en la era de las primeras computadoras analógicas y se trató, específicamente, de la Mark III (construida por la Universidad de Harvard para estudios militares). En esa computadora, la ingeniera contraalmirante de la Marina de los Estados Unidos Grace Hopper encontró, en 1947, el primer *bug* o error en la historia de la informática. Resulta que la máquina no funcionaba bien y, haciendo una revisión física, halló que en el relé 70 de un panel había una polilla que impedía todo contacto eléctrico. De ahí en más se asoció el error informático con la palabra *bug* (en inglés, bicho).



Extracto del informe del primer error informático documentado (izquierda) y Grace Hopper operando la computadora Mark III. Fuente: Wikipedia.

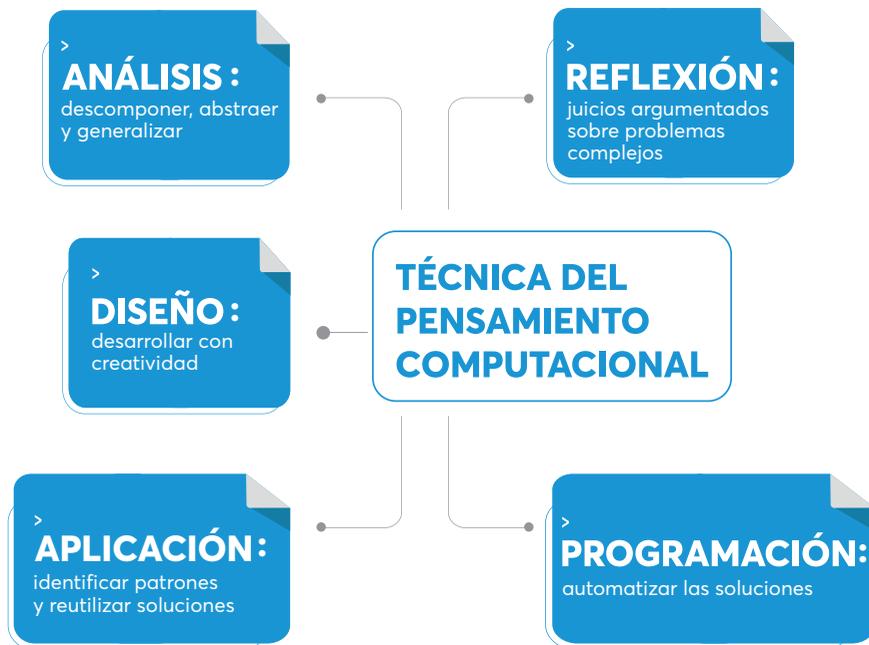
Técnicas asociadas con el pensamiento computacional

Más allá de las capacidades que configuran el pensamiento computacional, de manera complementaria existen una serie de técnicas asociadas a su desarrollo cuya función es ordenar y favorecer el trabajo cognitivo. Son las siguientes:

- ▶ **Reflexión:** se define como la habilidad de llevar a cabo juicios argumentados sobre situaciones que tengan cierto grado de complejidad. Desde la perspectiva de la informática, las técnicas que promueven la reflexión utilizan criterios sobre la comprensión de las especificaciones de los productos y las necesidades de los usuarios.
- ▶ **Análisis:** para analizar una situación problemática, se acude a una serie de herramientas auxiliares como dividir en partes de menor complejidad los componentes

de un problema (descomponer), reducir aspectos de complejidad (abstraer), identificar los procesos y buscar elementos patrones (generalizar). Se trata de utilizar la capacidad analítica, apelando a la lógica, para comprender plenamente el problema y así avanzar en su solución.

- ▶ **Diseño:** es una técnica que se aplica al proceso de desarrollo de una solución efectiva y eficiente de un problema. El diseño implica creatividad debido a que es necesario ver desde distintos puntos de vista un mismo problema, satisfacer las necesidades o deseos y adaptarse a un contexto que la hace posible y sustentable.
- ▶ **Programación:** todo diseño de una solución a un problema debe ser programado para que una computadora pueda automatizar el proceso. A esta traducción de la solución, realizada utilizando los recursos de un lenguaje computacional, se le debe aplicar un proceso de revisión técnica funcional que la evalúe en función de garantizar su funcionamiento correcto en todas las condiciones de trabajo. Asociada al proceso, está la operación de depuración, entendida como la actividad de corrección de errores que solo afectan la programación, debido a que no provienen de etapas previas.
- ▶ **Aplicación:** se basa en la adopción de soluciones existentes para satisfacer las necesidades de otro contexto. Esta habilidad implica el uso de la capacidad de generalizar debido a que se deben identificar patrones y realizar conexiones a efectos de adoptar lo preexistente.



Técnicas de pensamiento computacional.

Por otro lado, existen una serie de **actitudes** adicionales que ayudan a promover el desarrollo del pensamiento computacional. Estas están en relación con aspectos de quienes resuelven los problemas ya que les permitirán enfrentar tales situaciones de una manera más natural, reduciendo la tendencia a la frustración por resultados erróneos o no deseados.

Actitudes vinculadas al pensamiento computacional:

► **Perseverancia:** Es una de las capacidades extras por desarrollar junto con el pensamiento computacional ya que estamos tratando con problemas y, utilizando nuestra creatividad, debemos pensar en soluciones. Así, perseverar se entiende como la actitud de continuar sin ceder, ser resistente y tenaz en una tarea.

En muchas situaciones, al resolver problemas complejos, se necesita voluntad para superar, en ciertas ocasiones, las ganas de abandonar un proyecto y dejarlo inconcluso. La capacidad de perseverar es común a los profesionales de las ciencias de la computación dado que es frecuente que el trabajo intelectual se vea acompañado de fallas y errores que deriven en situaciones de frustración sino se tiene la capacidad mencionada. La tolerancia a los errores, su análisis y comprensión son parte de la disciplina que deben acompañar a todo profesional.



► **Experimentación:** Otra capacidad por desarrollar en los estudiantes tiene que ver con **experimentar**, lo que sucede frecuentemente a partir de probar cosas (en inglés, *tinkering*). Es una forma de intervención, asociada a no conformarse con lo establecido y ya hecho, sino con querer dar un valor agregado a las cosas a partir de una mirada y un hacer crítico que genera nuevas versiones de los objetos en estudio.

Desde la infancia, realizamos actividades de *tinkering*, a partir de experimentar probando cosas. En los niños pequeños, esta forma de indagación se presenta como una etapa basada en el juego y la exploración con el fin de aprender sobre los objetos del mundo. El **deconstruir** cosas es una actividad básica, desarmarlas para ver cuáles son sus componentes, materiales, uniones, conexiones, etc. es un juego de *hackeo*, donde el experimentador deja de ser un consumidor pasivo y pasa a ser alguien que se cuestiona su propio mundo y lo quiere intervenir.

Las comunidades de aprendizaje que experimentan con actividades basadas en el *tinkering* tienen la libertad de explorar y, en general, lo realizan en un entorno lúdico y libre de riesgos, lo que les da confianza en sí mismas a partir de las posibilidades de creación que aparecen. Estas actividades están asociadas con el razonamiento lógico; en la práctica, a medida que los estudiantes prueban cosas comienzan a acumular experiencias de tipo causa y efecto (por ejemplo: si muevo esto, entonces sucede aquello).

Las modificaciones a un objeto, derivadas de una actividad de *tinkering*, ayudan a ver que este abordaje de la tecnología pasa por desarrollar la comprensión en vez de obtener respuestas prearmadas y actividades totalmente cerradas. La experimentación brinda a niños y a jóvenes la posibilidad de enfocar algo desde distintas perspectivas, lo cual favorece una comprensión holística de las situaciones y suele traer de la mano soluciones no tradicionales.

► **Creatividad:** El trabajo creativo comprende aspectos de originalidad. Estos se ponen en juego al **crear** algo valioso y significativo para quien está involucrado directamente en el proceso. La creatividad se aplica tanto a las cosas tangibles como a las que no lo son: hoy los medios y las herramientas digitales han expandido significativamente el espacio de creación del hombre. En una actividad típica donde se crea algo, existen etapas asociadas que, en general, tienen que ver con la planificación, el hacer y la evaluación de lo creado. Por ejemplo, la programación de por sí es un proceso creativo.

Ken Robinson ha definido la creatividad como «el proceso de tener ideas originales que tienen valor» (Aprendemos Juntos, 2018). Por eso, toda actividad de aprendizaje que implique creación debe ser original, debe nacer de los estudiantes, no debe ser algo donde se copie o se complete sobre la base de instrucciones. Por otro lado, una vez terminado el trabajo, al compartirlo, el creador se comunica con otros y va desarrollando la confianza en sí mismo. La originalidad y el valor agregado cooperan en la construcción del conocimiento dado que, al tener fluidez en el manejo de técnicas, herramientas y materiales, se desarrollan habilidades, saberes e independencia. En síntesis, es una manera lúdica, experimental y útil de expresar las ideas.

Consideraciones finales

De acuerdo con Zapata-Ros (2015), hay una serie de conceptos que representan la esencia del pensamiento computacional y que merecen ser tenidos en cuenta cuando se piensa su aplicación y desarrollo en contextos educativos, en particular. Son los siguientes:

a) En el pensamiento computacional se conceptualiza, no se programa. Esto se debe a que se piensa como un científico de la computación y, por ende, se requiere pensar apelando a múltiples niveles de abstracción.

b) En el pensamiento computacional son fundamentales las habilidades no memorísticas o no mecánicas. Para programar hace falta una mente imaginativa que haga uso del pensamiento divergente.

c) En el pensamiento computacional lo importante son las ideas, no los artefactos. En esencia, el objetivo es ayudar a que se desarrolle un estudiante activo, creador de artefactos con tecnología, más que un usuario pasivo que se entretiene con objetos tecnológicos.

El desarrollo del pensamiento computacional en el currículum educativo puede verse como una oportunidad (Gurises Unidos, 2017) de ayudar a que niños y jóvenes construyan saberes significativos y necesarios para este tiempo. En particular, las oportunidades pueden estar en relación con las siguientes acciones:

- ▶ **Actualizar la alfabetización.** Desarrollar usos efectivos de las herramientas (Gurstein, 2003) y tener fluidez digital (Resnick, 2002) se ha vuelto tan importante como leer, escribir y hacer cuentas. Esto, por un lado, permite conectar a los estudiantes con su entorno, con su realidad diaria, y, por otro lado, al requerir de estudiantes activos, promueve una relación activa y crítica con tal entorno. Aplicar el pensamiento computacional en proyectos resolviendo problemas, generando prototipos de soluciones y analizando críticamente lo hecho permite superar con creces el «modelo de enseñanza bancario», según lo definió el pedagogo Paulo Freire, que critica el modelo educativo basado exclusivamente en la transmisión no dialógica. Los estudiantes pasan a ser coautores y no solamente usuarios pasivos. También, el pensamiento computacional puede colaborar para ayudar a comprender qué es lo que ocurre detrás de todo dispositivo digital, ya sea en su faz tecnológica o social.

Potenciar el trabajo colectivo. La mayoría de los objetos tecnológicos que nos rodean son el resultado de procesos de resolución de problemas que involucraron mucha inversión económica y de recursos humanos. En general, no fueron hechos por una única persona en un garaje o en un laboratorio aislado; por lo contrario, son el resultado de la acumulación de muchas soluciones aportadas por distintas personas. El trabajo en equipo es algo fundamental, tanto en la época en que se es estudiante, como a lo largo de toda la vida.

Los equipos permiten trabajar aspectos colaborativos y cooperativos. Los colaborativos se dan cuando los participantes trabajan todos a la vez sobre una situación determinada, en un ida y vuelta, tratando de enfocar sus esfuerzos en una sola cosa (por ejemplo, diseñar una solución a un problema o evaluar una solución construida). En cambio, el trabajo cooperativo sucede en un equipo cuando las tareas se dividen y cada participante o subgrupo del equipo realiza las que acordó hacer para lograr un determinado resultado (por ejemplo, en un diseño de solución alguien podría desarrollar un *hardware*; otro, fabricar un esqueleto de robot, y un subgrupo, programar las acciones acordadas). Para que se dé un trabajo efectivo en equipo, es necesario construir un marco para la colaboración, donde cada individuo participe en función de sus motivaciones, intereses y capacidades.

- ▶ **Conectar los saberes teóricos con la práctica.** El desarrollo efectivo del pensamiento computacional implica que los estudiantes, necesariamente, tienen que articular saberes prácticos con saberes teóricos. No es ni práctica vacía de sentido ni repetitiva, ni teoría desvinculada de la realidad de cada persona. Para diseñar y construir tecnología, en efecto, hay que conectar los elementos del hacer tecnológico con las bases del conocimiento teórico.
- ▶ **Apostar a la no directividad.** Es decir, intentar la superación del viejo mandato escolar que sostiene: «todos deben aprender lo mismo y al mismo tiempo». En este sentido, la no directividad implica tratar de encontrar nuevos modos de relación con el estudiante donde surja una actitud comprensiva del otro. Así, se intentan encontrar, a partir de una búsqueda compartida, los intereses de los estudiantes y canalizar la energía que tienen para tratar de aumentar sus posibilidades de autonomía, de confianza en sí mismos y de responsabilidad.
- ▶ **Promover la participación genuina y la autonomía.** El trabajo por proyectos, donde las situaciones problemáticas son el centro, es la base de un hacer pedagógico que promueve la construcción genuina de conocimientos ya que existe una apropiación de estos a lo largo del proceso y, también, en el objeto realizado. Pero, para que este proceso sea efectivo, debe estar conectado con las motivaciones e intereses de los educandos. De esta manera, al promover un ambiente de trabajo donde esté presente el manejo autónomo y participativo, se podrán lograr aprendizajes ricos.

En síntesis: hemos presentado el **pensamiento computacional** como una forma alternativa de trabajo cognitivo enriquecida por el pensamiento ingenieril, el pensamiento científico y el pensamiento lógico matemático, que se enfoca en el estudio de los problemas y sus soluciones, apuntando a que estas últimas puedan traducirse para que puedan ejecutarse por un procesador de información.

El método de trabajo asociado al PC plantea que todo problema complejo a trabajar debe descomponerse (**proceso de descomposición**) en subproblemas, lo cual implica que habrá más problemas, pero más pequeños y manejables, cuyas soluciones combinadas aportarán a la solución final del problema original.

El proceso de **abstracción** permite centrar la atención en las características más importantes del problema, al captar su esencia. Esto se logra al invisibilizar las características no fundamentales y, a la vez, destacar los rasgos más relevantes. Como resultado de la abstracción, se logra desarrollar una representación simplificada del problema.

Luego, se acude a la experiencia, al conocimiento compartido, cuando se procede a realizar el **reconocimiento de patrones**. Se buscan similitudes entre los distintos subproblemas dentro de una misma situación. De esta manera, se utiliza el conocimiento de problemas similares, que han sido resueltos con anterioridad y se lo aplica directamente (**generalización**). Cuantos más patrones se reconozcan, será más fácil y rápido el desarrollo de un proyecto.

Luego, cuando se tiene una solución al problema se la traduce en forma de plan de acción a ejecutar (**algoritmo**) para que sea realizado por una máquina o un agente humano. Finalmente, se realiza una **evaluación** profunda del algoritmo a los efectos de verificar que, por un lado, dé respuesta efectivamente al problema original, y, por otro lado, no contenga errores en sus instrucciones.

En el capítulo siguiente, se presentan una serie de recursos didácticos, en forma de tareas, para ayudar a desarrollar el pensamiento computacional en estudiantes de primer ciclo de la escuela secundaria.

ACTIVIDADES PARA DESARROLLAR EL PENSAMIENTO COMPUTACIONAL

“Más confío en el trabajo que en la suerte”

Anónimo

SOBRE LAS ACTIVIDADES PROPUESTAS

Se presentan 14 actividades para estudiantes de primer ciclo de la escuela secundaria orientadas a desarrollar conceptos y capacidades relacionadas con el pensamiento computacional. Están basadas en las tareas del concurso Bebras (<https://www.bebas.org>), que es una iniciativa educativa de alcance internacional para la promoción del pensamiento computacional, cuya meta principal es lograr que los estudiantes de la escuela primaria y secundaria se interesen en temas relacionados con las ciencias de la computación y los apliquen a la resolución de problemas.

La metodología Bebras es adecuada para trabajar en clases del primer ciclo de la escuela secundaria ya que no evalúa conocimientos previos o habilidades específicas sobre ciencias de la computación aprendidas en la escuela. Solo se presentan a los estudiantes actividades de resolución de problemas en forma de tareas, de modo que puedan aprender conceptos y desarrollar capacidades en torno al pensamiento computacional.

Formato de las actividades

Las actividades, como se dijo anteriormente, se presentan en forma de tareas, cada una puede resolverse en alrededor de cuarenta minutos y se compone de las siguientes secciones:

- ▶ **Presentación de la tarea.**
- ▶ **Respuesta y explicación de esta.**

- ▶ **Para saber más:** reflexiones e información ampliatoria para docentes y estudiantes. En esta sección, se recuperan temas presentados en los capítulos 1 y 2 para que puedan ser compartidos directamente con los estudiantes.
- ▶ **Desafíos:** para seguir profundizando en la resolución de tareas similares.

Metodología de trabajo

Las tareas que se presentan toman elementos del aprendizaje por indagación (Dostál, 2015). La idea principal es que los estudiantes estén motivados por desafíos y preguntas y, a partir de la guía docente, puedan desarrollar conceptos y reflexiones significativas. Esto les permitirá construir y evaluar respuestas a preguntas más complejas, como así también comunicar los conocimientos a los que han llegado.

En síntesis, se pretende que los estudiantes encuentren soluciones a un problema investigando sobre la base de un trabajo cooperativo. Luego, en una puesta en común de ideas, fundada en un proceso de reflexión general sobre lo trabajado, se presentan distintas perspectivas y se argumentan posiciones. Así, trabajando bajo un enfoque por indagación, se realiza una participación activa de los estudiantes en la construcción de conocimientos y, a la vez que se aprende a trabajar con problemas, se desarrolla el pensamiento crítico.

Estas catorce tareas componen un recurso didáctico introductorio a conceptos relacionados con el pensamiento computacional. Cada actividad está diseñada para ser trabajada en clase, de manera colaborativa, en un período de entre 30 y 40 minutos. En cada caso, se sugiere una resolución en grupos de no más de 3 o 4 integrantes.

Cada clase debería comenzar con una breve revisión de la resolución de la última actividad trabajada, discutiendo el desafío que quedó planteado. Es importante que la voz de los estudiantes esté presente en esta parte y, luego, una vez que se discutieron los distintos enfoques y soluciones, como cierre de esta primera parte, el profesor debe repasar los conceptos teóricos presentados en la clase anterior. En total, para este segmento de la clase se podrían dedicar alrededor de 15 minutos.

Una vez finalizado, el profesor debe entregar una hoja con la tarea del día a cada grupo de estudiantes, asignando 10 minutos para su resolución. No es necesaria una presentación de la actividad dado que las tareas son autoexplicativas. Si durante el proceso de resolución surgen dudas, se pueden dar pistas mínimas. Otro recurso interesante puede ser el intercambio de opiniones entre grupos de estudiantes.

Pasados los 10 minutos asignados a la resolución grupal, se invita a los integrantes de un grupo a explicar la solución alcanzada, haciendo hincapié en cómo fue el proceso de comprensión del problema y el diseño de la solución. Luego, se invita a participar a otros grupos para mostrar la existencia de alternativas de trabajo y así enriquecer el proceso desde otras perspectivas. En esta etapa, en ningún momento debe el docente dar la solución correcta dado que esto se entiende como un proceso de construcción por parte de los estudiantes.

A continuación, el docente le entrega a cada grupo el material con la segunda parte de la tarea. Les pide que realicen, por grupo, una lectura atenta de la solución y de la sección «Para saber más».

Luego, para finalizar la clase, el docente realiza una presentación formal de los conceptos abordados en el mencionado apartado tratando de conectarlos con las actividades recientes. La clase finaliza con la propuesta de resolución, en sus hogares o en el tiempo libre, de las actividades denominadas desafíos asociadas a la tarea trabajada.

Organización temática

Se presenta una tabla con la colección de tareas sugeridas para el desarrollo del pensamiento computacional. Junto a cada tarea, se indican las dimensiones principales asociadas y los conceptos teóricos por desarrollar.

TAREA	DIMENSIONES PRINCIPALES ASOCIADOS A LA TAREA	CONCEPTO DESARROLLADO EN EL APARTADO " PARA SABER MÁS"
1	Pensamiento algorítmico (PA)	PA. Concepto de algoritmo y secuencia
2	Pensamiento algorítmico, descomposición (DE)	PA. Concepto de programa y computadora. Determinismo
3	Pensamiento algorítmico, generalización (GE)	PA. Estructura secuencial, concepto de ciclo
4	Abstracción (AB), evaluación (EV)	AB. Concepto de abstracción
5	Abstracción	AB. Diferentes representaciones de un problema, seleccionar la mejor representación
6	Generalización, descomposición	GE. Concepto de patrón y generalización
7	Generalización, abstracción	GE. Reconocimiento de patrones, clasificación
8	Pensamiento algorítmico, evaluación	PA. Lógica, razonamiento lógico
9	Pensamiento algorítmico, evaluación	PA. Lógica y computadoras
10	Descomposición, evaluación	DE. Concepto de descomposición
11	Descomposición, evaluación	DE. Reducción de la complejidad de un problema
12	Evaluación, generalización	EV. Concepto de evaluación
13	Evaluación, generalización, abstracción	EV. Búsqueda de errores y mejora del uso de recursos
14	Pensamiento algorítmico, abstracción, generalización	Pensamiento computacional

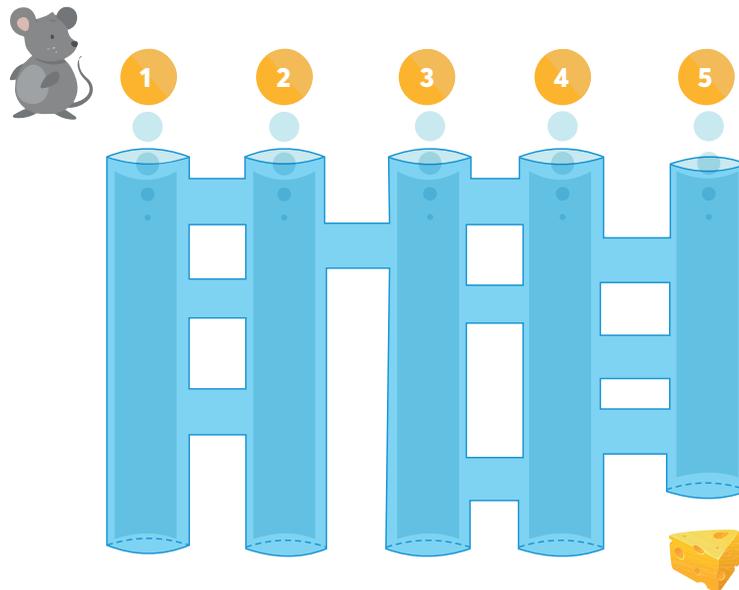
¿POR DÓNDE VA?

TAREA 1

TEMA: pensamiento algorítmico

Un ratón de laboratorio, llamado XC4, ha sido entrenado por científicos. En un experimento, está situado en la entrada de un sistema de cañerías y el objetivo es que llegue al queso que se encuentra al final del quinto caño. Estas son las instrucciones que siempre sigue XC4:

1. Bajá por el tubo hasta que aparezca un túnel nuevo.
2. Cada vez que te encuentres con un túnel nuevo, debés atravesarlo.
3. Volvé a la instrucción 1.



PREGUNTA

¿En cuál entrada debería ingresar el ratón para llegar al queso?

1. Buscá una escalera y ubicala en posición de alcanzar la lamparita rota.
2. Buscá una lamparita nueva.
3. Cortá la energía eléctrica.
4. Subí la escalera con la lamparita nueva.
5. Desenroscá la lamparita rota.
6. Enroscá la lamparita nueva.
7. Bajá la escalera.
8. Volvé a conectar la energía eléctrica.
9. Comprubá que la lamparita nueva funcione.
10. Si funciona, fin del proceso; sino, volvé al paso 2.

Como puede observarse en el algoritmo anterior, las instrucciones son claras y precisas, están ordenadas por una secuencia lógica que debe respetarse y hay especificado un inicio y un fin. De manera más formal, un algoritmo es una **secuencia finita y ordenada de pasos para llegar a hacer una tarea determinada**. Ejemplos de algoritmos en la vida cotidiana pueden ser una receta de cocina, un manual de un artefacto que indica cómo ensamblarlo (por ejemplo: un mueble) o un listado de acciones por realizar en caso de que se produzca una emergencia en un edificio.

En la escuela, los estudiantes también suelen trabajar con algoritmos:

- ▶ en ciencia, cuando se siguen las instrucciones para realizar un experimento;
- ▶ en matemáticas, cuando se aplica un procedimiento para resolver un ejercicio (por ejemplo, el cálculo de un promedio);
- ▶ en tecnología, cuando se siguen una serie de instrucciones para ensamblar algún artefacto;
- ▶ en arte, cuando se aplica alguna técnica de fabricación o composición (por ejemplo, la fabricación de una pieza de cerámica).

DESAFÍO 1. Reentrenando al ratón

Los científicos están buscando la forma de entrenar nuevamente al ratón para que, ingresando por cualquier tubo, salga siempre por el número cinco, ¿podés escribir un algoritmo que resuelva el problema?

TAREA 2. Camino a casa**RESPUESTA**

El camino más corto es: A, I, A, A, A, A, I, A, A.

EXPLICACIÓN

La tarea por realizar es simple y se resuelve mirando el plano. Si bien desde las distancias en el plano hay dos posibles soluciones que respondan al camino más corto, no es así desde el criterio de menor cantidad de instrucciones. Es importante que los algoritmos sean eficientes, así se utilizan menos recursos.

PARA SABER MÁS

Las **computadoras** sirven para realizar **tareas bien definidas**, para ello hay que **programarlas**. La programación se realiza mediante un **lenguaje computacional** que las máquinas son capaces de **interpretar y ejecutar**. Este lenguaje es diferente del que utilizan las personas ya que está compuesto de **instrucciones claras, concisas e inequívocas**.

Un **algoritmo** se convierte en **programa** cuando un programador lo traduce a instrucciones que una **computadora puede interpretar y ejecutar** (en este caso, la del automóvil autónomo).

Las computadoras siempre ejecutan las instrucciones de un programa al pie de la letra desconociendo las intenciones del programador y el problema por resolver. En la tarea anterior, si el programa es correcto (no contiene errores), el automóvil hará lo que se le ha indicado y así resolverá el problema (en este caso, conducir a Heriberto a su hogar). Ahora bien, si el programa guía el automóvil a otro lugar, aunque sigue siendo un programa válido, no resuelve el problema. Finalmente, si el programa tiene instrucciones no reconocibles (recordar que solo acepta I, D y A), la computadora no sabrá que hacer y entonces dará un mensaje de error.

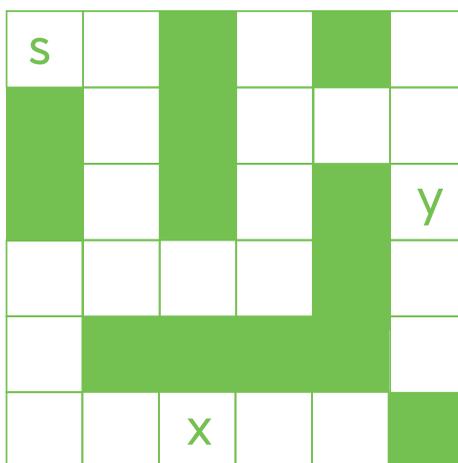
Las computadoras, independientemente de su forma y propósito (de escritorio, teléfonos inteligentes, robots de entretenimiento o industriales, etc.), siguen las instrucciones que les indican los programas que se les cargan. Los programas de computadora son **deterministas**, es decir, si siempre se ingresan los mismos datos, el programa realizará los mismos cálculos y entregará los mismos resultados.

Uno de los trabajos de los científicos de las ciencias de la computación es encontrar los algoritmos más **eficientes**. Es decir, aquellos que resuelven un problema en el menor tiempo, usando la menor cantidad de recursos (por ejemplo: memoria, acceso a disco rígido, uso de red, etc.).

Cuando hacemos búsquedas en internet, los buscadores utilizan algoritmos para recuperar la información solicitada y seleccionar y visualizar los resultados en el menor tiempo posible, entendiendo que la eficiencia en estos casos está lograda al recuperar y analizar un gran volumen de datos en pocos segundos.

DESAFÍO 2. El robot en su laberinto

El siguiente laberinto contiene dos tesoros marcados como X e Y. Los bloques negros muestran dónde están ubicadas las paredes y los bloques blancos indican los caminos por donde podría viajar un robot.



Las instrucciones que le podés dar al robot son las siguientes:

- ✓ Ax: avanzá x bloques.
- ✓ D: girá a la derecha 90°.
- ✓ I: girá a la izquierda 90°.
- ✓ T: recogé tesoro.

Inicialmente, el robot está en la posición S y está mirando hacia la derecha del mapa. El robot solo puede recoger el tesoro si está en la misma casilla del mapa que el tesoro. Por ejemplo, en el siguiente algoritmo se muestra cómo recogería el robot el tesoro X:

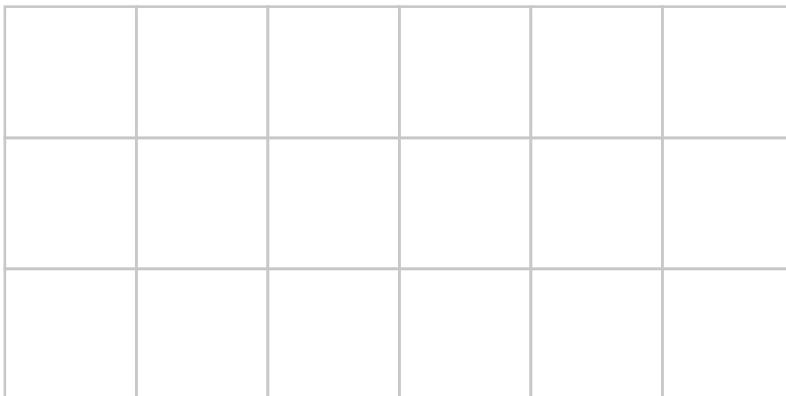
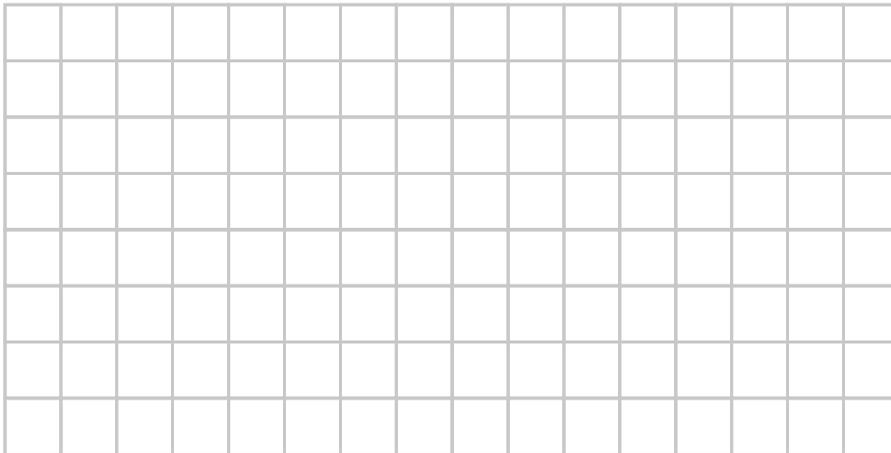
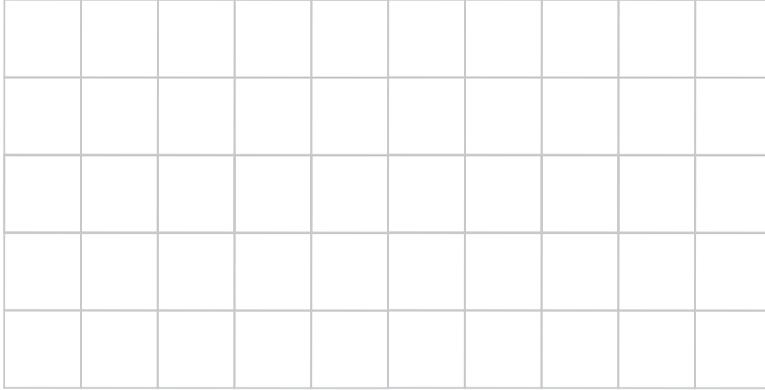
A1, D, A3, D, A1, I, A2, I, A2, T.

PREGUNTA

¿cuál es el algoritmo necesario para programar al robot (saliendo de la casilla S con el objetivo de recoger el tesoro)?

DESAFÍO 2 (yapa). Tu propio laberinto

¿Te animás a crear tu propio laberinto con un tesoro para que un compañero, la próxima clase, pueda desarrollar el algoritmo?



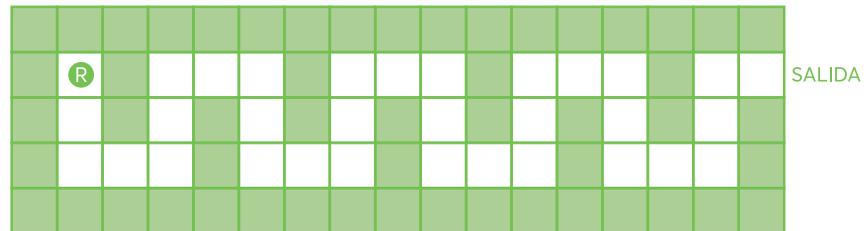
EL ESCAPE

3

TAREA

TEMAS: pensamiento algorítmico y generalización.

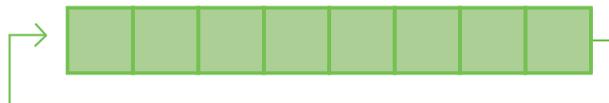
Es necesario ayudar a que el robot BAUN3 escape del laberinto.



Las instrucciones básicas que entiende el robot son las siguientes:



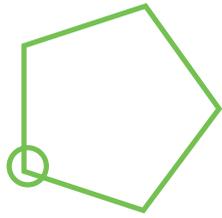
El robot está preparado para repetir cuatro veces una secuencia de ocho instrucciones (la que vos armes).



4X

PREGUNTA

¿cuál es la secuencia de 8 instrucciones, que se repetirá cuatro veces, necesaria para que el robot escape?



```
repetir 5 veces  
hacer  
  avanzar 100  
  girar a la derecha 72
```

Estructura de ciclo con indicación de cantidad de repeticiones.

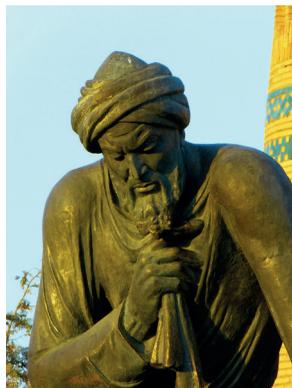
En el ciclo planteado en el algoritmo de esta tarea, el robot repite cuatro veces las mismas instrucciones. Pero puede haber otros casos de ciclos donde la cantidad de repeticiones no esté determinada por un número fijo, sino por una condición. Por ejemplo, repetir hasta que no queden bolitas en la caja o repetir mientras no se llegue a una pared.



```
repetir hasta  
hacer  
  avanzar  
  si hay camino a la izquierda  
  hacer girar a la izquierda
```

Estructura de ciclo con condición de finalización.

Curiosidad: la palabra **algoritmo** deriva del nombre de un matemático árabe, **AlJuarismi**, quien vivió en la ciudad de Bagdad alrededor del año 800 d. C. Se dedicó a estudiar álgebra y astronomía. Es reconocido por sus procedimientos para resolución de ecuaciones y un tratado sobre números —*Algoritmi de numero indorum*, título de su trabajo traducido al latín.⁵



Estatua de AlJuarismi, conocido como el padre del álgebra

⁵ Más información, en Wikipedia (<https://es.wikipedia.org/wiki/Algoritmo>).

DESAFÍO 3. Escribir el algoritmo I

En el cuento «Instrucciones para subir una escalera» de Julio Cortázar, se puede detectar un patrón de repetición de instrucciones. ¿Cuáles son las instrucciones que se repiten?

Las escaleras se suben de frente, [...] los brazos colgando sin esfuerzo, la cabeza erguida, aunque no tanto que los ojos dejen de ver los peldaños inmediatamente superiores al que se pisa, y respirando lenta y regularmente. Para subir una escalera se comienza por levantar esa parte del cuerpo situada a la derecha abajo, envuelta casi siempre en cuero o gamuza, y que salvo excepciones cabe exactamente en el escalón. Puesta en el primer peldaño dicha parte, que para abreviar llamaremos pie, se recoge la parte equivalente de la izquierda (también llamada pie, pero que no ha de confundirse con el pie antes citado), y llevándola a la altura del pie, se le hace seguir hasta colocarla en el segundo peldaño, con lo cual en este descansará el pie, y en el primero descansará el pie. [...]

Llegando en esta forma al segundo peldaño, basta repetir alternadamente los movimientos hasta encontrarse con el final de la escalera [...].

(Cortázar, 1962)

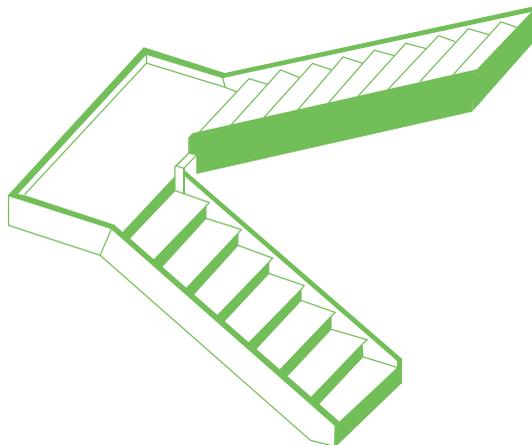
DESAFÍO 3 (yapa). Escribir el algoritmo II

PREGUNTA

¿podés escribir un algoritmo que permita que un robot suba la siguiente escalera?

Si necesitás más instrucciones que las propuestas en la tarea «El escape», podés definir las: por ejemplo, alguna del tipo «mientras no haya llegado [...]».

Recordá que las instrucciones se escriben en modo imperativo.



EDITANDO NOTICIAS

TAREA 4

TEMAS: abstracción y evaluación.

En un periódico hay un equipo de diez periodistas. Todos los días escriben o editan sus propios artículos en determinados horarios en los que asisten a la redacción.

Las marcas, en el siguiente calendario, muestran cuándo los periodistas necesitan una computadora (son todas iguales). Durante una hora, solo un periodista a la vez puede trabajar en una computadora.

		HORAS						
		8.00	9.00	10.00	11.00	12.00	13.00	14.00
PERIODISTAS	1							
	2							
	3							
	4							
	5							
	6							
	7							
	8							
	9							
	10							

PREGUNTA

¿cuál es el número mínimo de computadoras necesarias para que todos los periodistas trabajen de acuerdo con el plan que se muestra arriba?



TAREA 4. Editando noticias

RESPUESTA

La respuesta es 5 computadoras.

EXPLICACIÓN

	8.00	9.00	10.00	11.00	12.00	13.00	14.00
1		PC 3	PC 3				
2			PC 1	PC 1	PC 1	PC 1	
3	PC 1	PC 1					
4					PC 3	PC 3	PC 3
5		PC 4	PC 4				
6				PC 2	PC 2		
7			PC 5				
8		PC 5					
9	PC 2	PC 2	PC 2				
10						PC 2	PC 2

A las 9 h y a las 10 h, cinco periodistas necesitan una computadora, no podemos resolver el problema con menos de cinco computadoras.

Para analizar importantes cantidades de datos y encontrar relaciones de interés, es mejor crear un método de representación, por ejemplo, una tabla o un diagrama. Según sea de la información de interés, podemos representar los mismos datos de manera diferente.

PARA SABER MÁS

En esta tarea, se utilizó el gráfico calendario para mostrarte de manera clara cómo se distribuyen las computadoras a lo largo del día. El calendario es una **herramienta que permite resaltar la información de interés para un problema y, al mismo tiempo, ocultar aquella que no lo es**. A esto se denomina *abstracción*.

La abstracción es parte de la esencia del pensamiento computacional, es un proceso que ayuda a simplificar las cosas al identificar lo que es importante sin preocuparse demasiado por los detalles. De esta forma, al poder aplicar la abstracción, estamos **gestionando la complejidad**.

El gráfico de calendario es un buen ejemplo de abstracción debido a que solo interesa comunicar quién trabaja y a qué hora. Todos los detalles accesorios no se muestran (por ejemplo: de qué tipo de máquinas se trata, qué sistema operativo tienen, nombres y apellidos de los periodistas, etc.) porque están demás y no son de interés en la comunicación.

Un ejemplo de abstracción puede darse en los mapas. Cuando el interés está en las divisiones políticas, se suelen ignorar casi todos los otros posibles detalles. Ahora,

si lo que nos interesa son formaciones rocosas y relieves, debemos acudir a una representación diferente, donde se las realce y se escondan otros tipos de atributos.

MAPA POLÍTICO



MAPA OROGRÁFICO

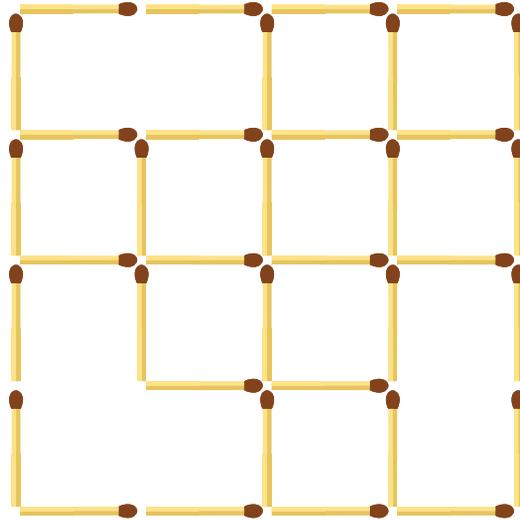


Distintos mapas de la República Argentina. Fuente: Educ.ar.

Otros ejemplos de la escuela en los que se usa la abstracción:

- ▶ En música, la partitura de piano de una canción es una abstracción para esa pieza musical.
- ▶ Cuando se diseña un mapa mental, se apela a un proceso de abstracción ya que se están dejando de lado los detalles que no se necesitan para comunicar las ideas principales.
- ▶ Un resumen de un libro es una abstracción.
- ▶ En prácticas del lenguaje o en arte, el guion de una obra es una abstracción ya que resume lo esencial cuadro a cuadro.

Las abstracciones a veces se representan como capas o jerarquías, lo que nos permite ver las cosas con diferentes grados de detalle. La naturaleza de poder ocultar la complejidad dentro de «cajas», que a la vez van dentro de otras «cajas», hace que la abstracción sea una herramienta poderosa ya que no debemos preocuparnos por los detalles de lo que ocurre dentro de cada «caja». Por ejemplo, en informática, la abstracción se utiliza para gestionar la complejidad de gran parte de lo que se diseña y crea.

DESAFÍO 4. Contar cuadrados**PREGUNTA**

¿cuántos cuadrados (de todos los tamaños) hay en el dibujo?

DESAFÍO 4 (yapa). Próximo vuelo

La empresa Aerolíneas B. utiliza un código único para identificar sus vuelos

- ▶ El vuelo SLA 12 sale a las 06:15.
- ▶ El vuelo SLA 15 sale a las 08:25.
- ▶ El vuelo SLA 04 sale a las 12:10.
- ▶ El vuelo SLA 08 sale a las 15:20.

PREGUNTA

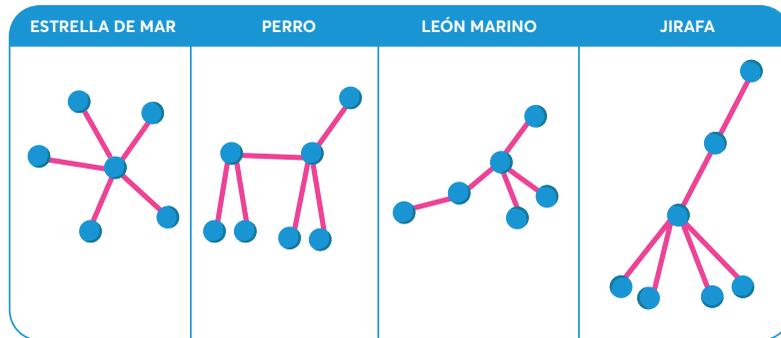
¿cuál es el código para el vuelo que sale a las 18:00? ¿Cómo se construye un código de vuelo?

FIGURAS DE ANIMALES

TAREA 5

TEMAS: abstracción y generalización.

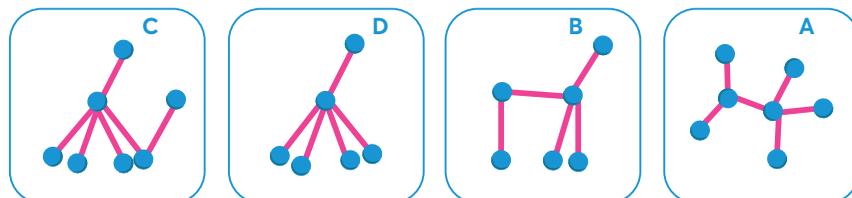
Gerardo estaba jugando en el bosque y usó frutos y palitos para crear cuatro simpáticos animales.



La hermana de Gerardo deformó a los animales sin quitar ninguno de los palitos y Gerardo se enojó porque realmente le gustaba la figura del perro.

PREGUNTA

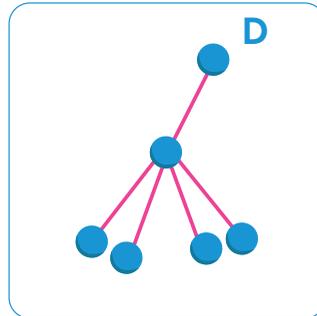
¿cuál de las siguientes figuras se puede reacomodar para volver a ser la figura del perro?





TAREA 5. Figuras de animales

RESPUESTA



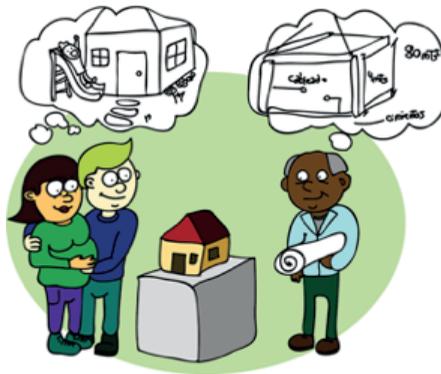
EXPLICACIÓN

Cada animal puede ser descrito por sus partes (frutas) y sus conexiones (palitos). Las posiciones específicas de las partes y los ángulos de las conexiones pueden cambiar mientras se juega con ellas, pero eso no cambia la esencia de la estructura del animal (ya que sigue teniendo las mismas partes conectadas de igual manera). Para responder la pregunta, es necesario determinar qué imagen tiene la misma estructura que la del perro. Por ejemplo, no puede ser la estrella de mar ya que posee una estructura formada por una parte central y cinco brazos.

Con estos ejemplos de animales hechos con frutas y palitos, **abstraemos** características como la piel, color, tamaño, etc. Se representa al animal solo por la estructura de su cuerpo; el resto no es importante.

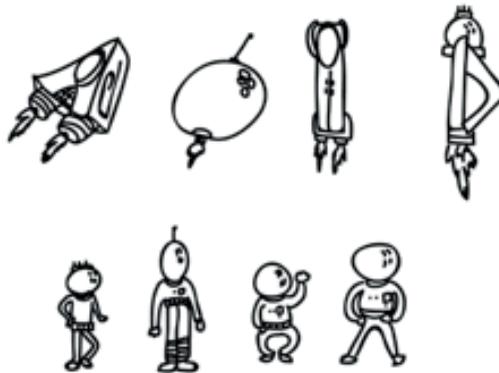
PARA SABER MÁS

En un proceso de **abstracción** sobre un problema dado, se decide **qué detalles se deben resaltar y qué detalles se pueden ignorar**. Teniendo en cuenta que la idea principal de algo no está en los detalles, sino en los grandes rasgos, al abstraer se obtiene como resultado una representación más comprensible del problema que se desea resolver.



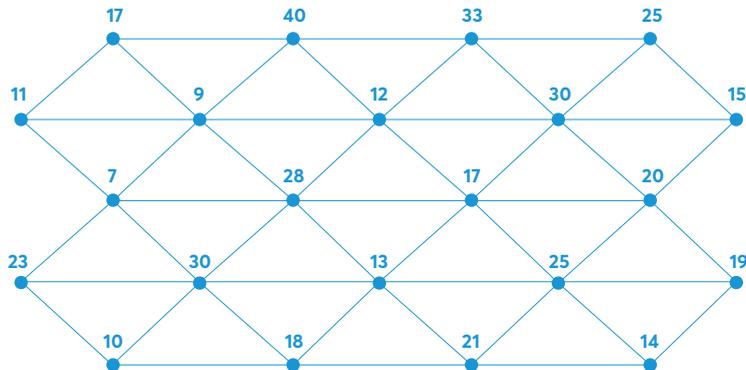
En la imagen anterior, podemos ver a una pareja y a un arquitecto analizando un proyecto de casa. Ambos tienen intereses distintos: por un lado, la pareja ve imágenes futuras acerca de cómo será su hogar, mientras que el arquitecto ve las estructuras y los materiales que se requerirán en la construcción. Este ejemplo nos muestra que, sobre un mismo problema, se pueden generar distintas abstracciones, cada una centrada en las necesidades particulares de cada persona. Lo importante, para avanzar en la solución a un problema, es elegir una buena representación —es decir, la que mayor información aporte— a partir de resaltar los elementos de interés y ocultar los que no lo son.

¿Podrías indicar en el siguiente ejemplo qué nave pertenece a qué astronauta?



Este ejemplo nos invita a analizar las características comunes (reconocer patrones) de ciertos objetos para establecer una asociación entre elementos de dos conjuntos. Esta tarea requiere, por lo tanto, analizar uno a uno los componentes de la imagen, compararlos y buscar posibles pistas y, al mismo tiempo, nos obliga a generar conjeturas. Por ejemplo, si analizamos los cascos, podemos ver algunas similitudes con las naves, pero no se puede establecer una relación con todas ellas. En cambio, si examinamos la posición de las piernas de los astronautas, vemos que coinciden con la forma de sus naves.

DESAFÍO 5. ¿Dónde están los lagos?



En cada punto numerado en el diagrama anterior, se encuentra una fuente de agua. El número indica la altitud sobre el nivel del mar. Las líneas entre los puntos numerados son canales que conectan las fuentes. Si el agua corre a una fuente y no puede seguir, se estanca y forma un lago. Por ejemplo, desde el punto 11 el agua corre hacia las fuentes 7 y 9 y no a hacia la 17.

PREGUNTAS

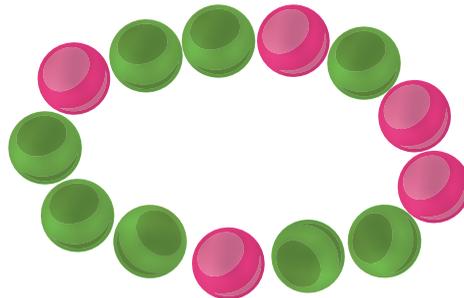
¿en qué puntos se formarán los lagos? ¿Qué tienen en común todos estos puntos?

BRAZALETE MÁGICO

TAREA 6

TEMAS: generalización y descomposición.

Un superhéroe posee un brazalete mágico:



El brazalete se mezcló con otros tres (sin poderes) y el superhéroe necesita recuperarlo.

PREGUNTA

¿cuál de los cuatro brazaletes siguientes es el que tiene poderes mágicos?

- 1 
- 2 
- 3 
- 4 



TAREA 6. Brazaletes mágico

RESPUESTA

La respuesta es el brazaletes 2.



EXPLICACIÓN

El brazaletes es un ejemplo de una secuencia de objetos. Las bolas están dispuestas en un cierto **orden** que se llama **patrón**. Para identificar el brazaletes correcto, hay que identificar el patrón y luego revisar cada opción para ver dónde hay coincidencia.

Cuando resolvemos problemas, es útil poder **identificar patrones**, y así luego encontrar objetos similares en diferentes lugares, ya que esto simplifica la tarea.

PARA SABER MÁS

Un **patrón es algo que se repite de manera continua** en una secuencia o en un conjunto de elementos. Cuando se intenta resolver un problema, en la etapa de análisis, descubrir qué patrones asociados existen permite **entender mejor las distintas partes que lo componen** y cuáles deben ser tenidas en cuenta en la etapa de diseño de soluciones. Es decir, que saber detectar patrones y descubrir las reglas que los definen ayuda a entender mejor los problemas y diseñar sus posibles soluciones.



Ejemplo de una serie de números con un patrón asociado: el número siguiente es el número actual más cuatro

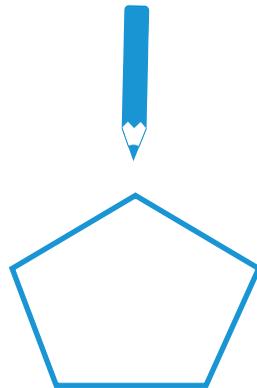
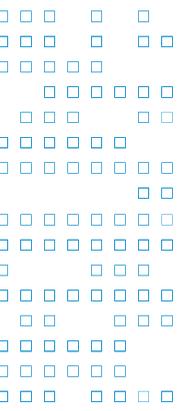
Las situaciones que involucran patrones habitan nuestra vida diaria. Por ejemplo, se usan patrones climáticos para crear los modelos de pronósticos meteorológicos; las mascotas descubren patrones de respuesta de sus dueños a sus acciones y así adaptan su comportamiento a una manera esperada la próxima vez; cuando un adulto maneja un automóvil, observa el patrón de conducta del conductor del vehículo que se encuentra adelante, lo que le permite tomar precauciones ante posibles maniobras.

Así, al identificar patrones, podemos hacer predicciones, crear reglas y resolver problemas más generales. El método de buscar un enfoque general para una clase

de problemas se llama *generalización* y es una actividad que realizamos al mismo tiempo que analizamos patrones. Por ejemplo, para la serie numérica presentada anteriormente, la generalización en modo de fórmula matemática es $(n-1) + 4$ para calcular el valor n de la serie.

¿Por qué son importantes los patrones? Los informáticos se esfuerzan por resolver problemas de manera rápida y eficiente, para lo cual muchas veces **reutilizan los métodos** ya creados. Entonces, si ven un patrón común en un problema, buscarán crear una única solución común para poder **volver a utilizarla muchas veces**. Esto significa que solo tienen que **diseñar y construir la solución común una sola vez, en lugar de hacerlo para cada caso**.

A continuación, se presentan dos programas que resuelven un mismo problema: dibujar un pentágono. En el programa 1, se utiliza únicamente una estructura secuencial (que se caracteriza por que cada instrucción se ejecuta una sola vez). Si se observan con detalle las instrucciones del programa, se puede descubrir un patrón de dos instrucciones que se repite cinco veces (avanzar 80 pasos y girar 72 grados). El programador que realizó el programa 2, al descubrir el patrón, utilizó una estructura repetitiva para simplificar la solución (es decir, aplicó la técnica de generalización), escribiendo menos instrucciones para realizar la misma tarea.



Inicio 

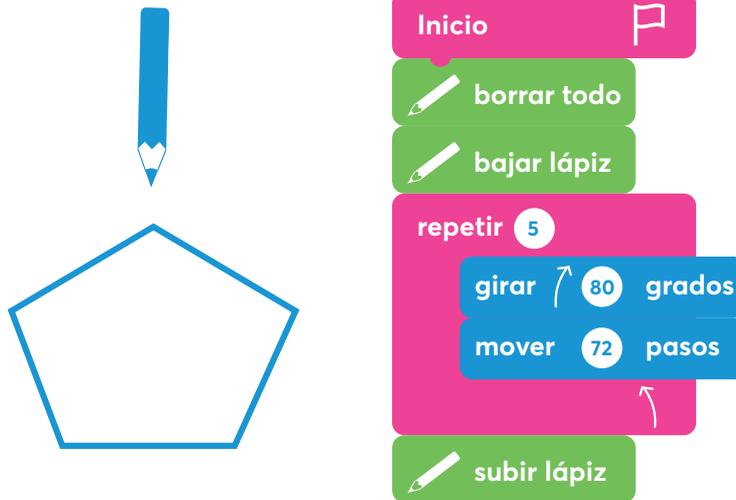
 borrar todo

 bajar lápiz

mover 80 pasos

girar ↗ 72 grados

 subir lápiz



Programas escritos utilizando un lenguaje de programación con una interfaz visual denominada de bloques (en este caso, Scratch), que son muy simples de leer y también de escribir.

DESAFÍO 6. Próximos

PREGUNTA 1:

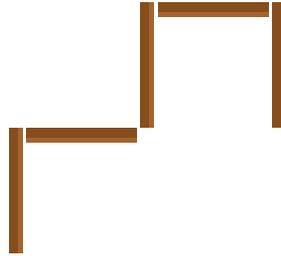
¿qué número sigue en esta secuencia?
2, 4, 8, 16, 32, 64, 128,...

PREGUNTA 2:

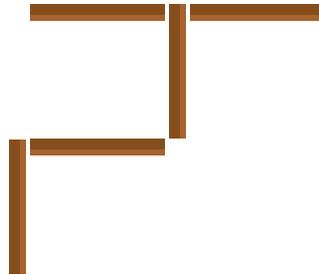
¿qué letra sigue en esta secuencia?
Z, Y, Z, Y, X, Z, Y, X, W, Z, Y, X, W, ...

DESAFÍO 6 (yapa). Cinco palitos

Ana tiene cinco palitos. Los pone sobre la mesa y crea esta figura.



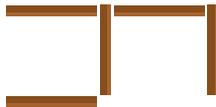
Norma llega a la mesa, toma un palito y lo pone en un lugar diferente.



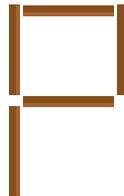
Luego Braulio se acerca a la mesa, también toma un palito y lo pone en un lugar diferente.

PREGUNTA :

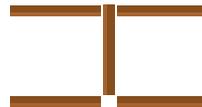
¿qué forma no puede haber hecho Braulio?



A



B



C



D

ARTE CON TRONCOS

TAREA 7

TEMAS: generalización y abstracción.

Cuando los castores roen los árboles, disfrutan colocando las piezas de una manera especial. Comienzan con un tronco, lo parten en pedazos y generan una forma (la que se observa en el estado 1). Luego, toman cada uno de los troncos del estado 1 y los vuelven a roer, manteniendo la forma con elementos más pequeños (el resultado de esto se observa en el estado 2).

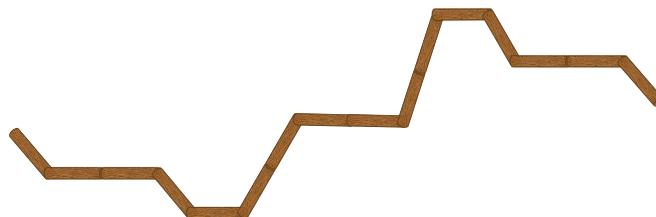


Aquí hay tres ejemplos. En cada uno se puede ver cómo comenzó el castor, el resultado después de la etapa 1 y el resultado después de la etapa 2.

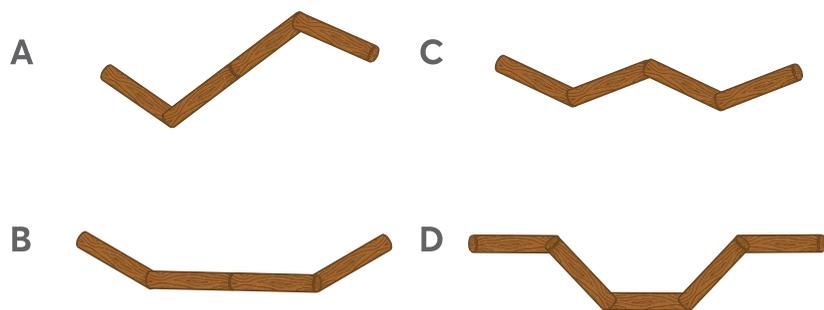
	TRONCO ORIGINAL	ESTADO 1	ESTADO 2
EJEMPLO 1			
EJEMPLO 2			
EJEMPLO 3			

PREGUNTA

si el resultado de la segunda etapa es el siguiente,



¿cuál fue la primera etapa?



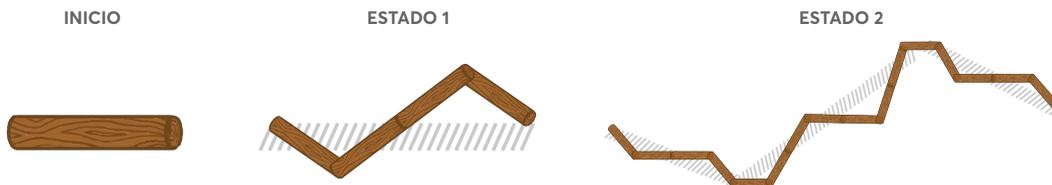
TAREA 7. Arte con troncos

RESPUESTA

La respuesta es la figura A.



EXPLICACIÓN



Como ya se vio, los programas representan un conjunto de reglas que se pueden ejecutar en una computadora. Incluso las reglas muy simples aplicadas reiteradamente pueden conducir a un comportamiento complejo.

En esta tarea, presentamos las reglas de construcción de imágenes **fractales**, que son gráficos con una forma autosimilar, estructurada infinitamente. Incluso las reglas simples pueden dar lugar a gráficos complejos y hermosos.

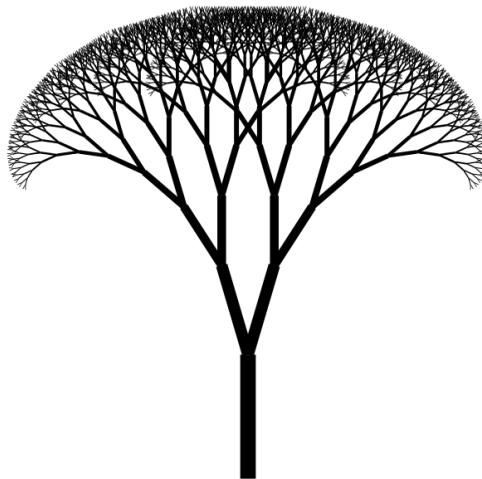


Imagen de un árbol generada a partir de fractales.

PARA SABER MÁS

En este caso, acudimos a la historia: en 1854, en Londres hubo un brote de cólera que cobró la vida de 616 vecinos. El médico John Snow⁶ realizó una investigación sobre la enfermedad y las muertes. Entrevistó a familiares de las víctimas sobre sus rutinas diarias y realizó un mapa que vinculó la enfermedad con el agua que tomaban. Al mirar un plano, donde dispuso los datos recolectados de las entrevistas, se dio cuenta de que todos los enfermos bebían de una bomba de agua que estaba cerca de una cloaca con filtraciones. Mucho antes que las computadoras aparecieran, el médico utilizó la técnica de **reconocimiento de patrones** para resolver un problema grave.



Mapa del médico John Snow en el que se observa la localización de la bomba de agua (punto celeste) y, con marcas, la ubicación de las viviendas de los vecinos fallecidos.

Como hemos visto, al identificar patrones podemos hacer predicciones, crear reglas y resolver problemas más generales. En computación, el método de buscar un enfoque general para una clase de problemas se llama **generalización**. En la escuela, la generalización es una tarea común, por ejemplo, en la clase de matemáticas se aprenden fórmulas y estas son generalizaciones.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Fórmula general para resolver ecuaciones de segundo grado.

⁶ Más información en Wikipedia, en la página dedicada al médico John Snow(https://es.wikipedia.org/wiki/John_Snow).

Veamos un caso simple y detallado de generalización. Supongamos la siguiente situación: una persona está a cargo de un campo con animales y existe comida suficiente y especial para cada uno. Las instrucciones para que un operario alimente a los animales son muy simples:

- ▶ Para alimentar al perro, poné la comida del perro en el plato del perro.
- ▶ Para alimentar al pollo, poné la comida del pollo en el plato del pollo.
- ▶ Para alimentar al conejo, poné la comida del conejo en el plato del conejo.

Nótese que hay una estructura subyacente compartida en cada una de las instrucciones anteriores, es decir, un patrón común, que podría expresarse de la siguiente manera:

- ▶ Para alimentar al <animal>, poné la comida del <animal> en el plato del <animal>.

Al detectar un patrón en las instrucciones de alimentación de los animales, se pudo realizar luego un proceso de generalización que simplifica la tarea de alimentación a partir de una sola instrucción genérica.

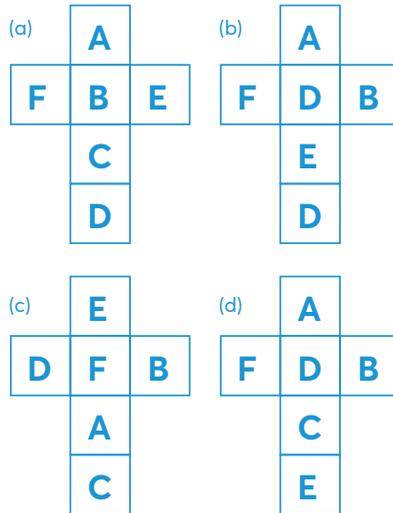
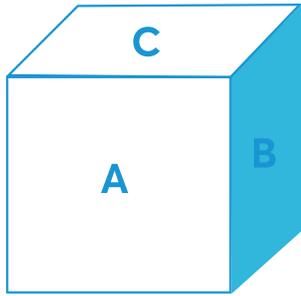
También se trabaja con patrones al realizar actividades de **clasificación** ya que en ellas se deben descubrir características comunes para crear asociaciones posibles. Cuando se realiza una tarea de este tipo, se deben examinar uno a uno los elementos presentados en el problema para **crear reglas** y luego clasificarlos.

¿SE PUEDEN ASOCIAR ESTAS IMÁGENES EN GRUPOS?, ¿CUÁLES?	POSIBLES SOLUCIONES	
	COMUNICACIÓN: 	COMIDA: 
	CON PERSONAS: 	SIN PERSONAS: 

DESAFÍO 7. Armar el cubo

PREGUNTA :

¿cuál de las figuras(a, b, c o d) creará el siguiente cubo cuando se pliegue?

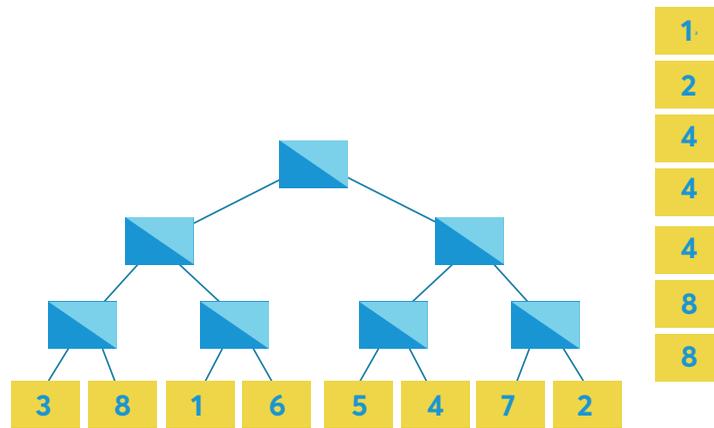


RESULTADOS REVUELTOS

TAREA 8

TEMAS: pensamiento algorítmico y evaluación.

Ana observó un campeonato de esgrima y registró a los ganadores de cada etapa en el tablero que se muestra a continuación. Los competidores llevaban los mismos números, del 1 al 8, durante todo el campeonato. Ana usó tarjetas numeradas para representar a cada competidor.



Cuando finalizó el campeonato, el hermano menor de Ana, Agustín, mezcló todas las tarjetas, excepto las de la primera ronda del campeonato.

PREGUNTA

¿Es posible reconstruir el resultado de las competencias (casillas celestes) a partir de observar las tarjetas desordenadas que se encuentran a la derecha del gráfico?

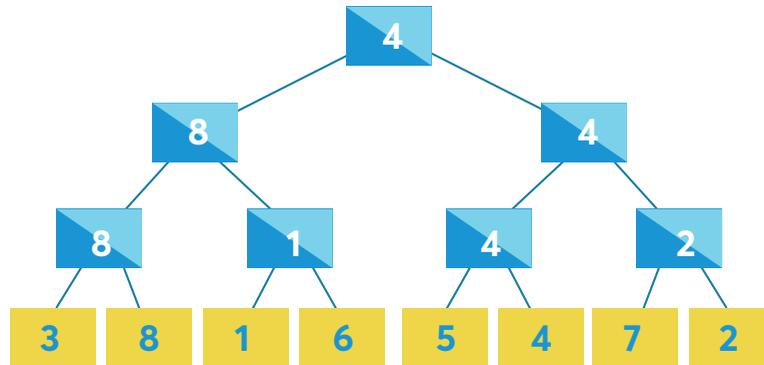
Completá las casillas celestes.



TAREA 8. Resultados revueltos

RESPUESTA

La disposición total de todas las tarjetas es la siguiente:



EXPLICACIÓN

Las tarjetas desordenadas corresponden a participantes que ganaron al menos una competencia. Por lo tanto, para resolver la segunda ronda solo hace falta ver quién ganó cada duelo particular, escribir su identificación en la casilla correcta y tacharlo de la lista. De esta manera, cuando se llega a completar la última casilla celeste, se conoce al ganador del campeonato.

PARA SABER MÁS

La **lógica** estudia los principios de la demostración de los argumentos mediante la **comprobación de la validez de expresiones**, que pueden ser evaluadas como: **ciertas/verdaderas** o **falsas/no verdaderas**. Así, la lógica estudia las inferencias y el pensamiento humano.

Ejemplos de expresiones:

- ▶ Hoy llueve.
- ▶ Hace frío.
- ▶ Es de color azul.
- ▶ Ana es inmortal.

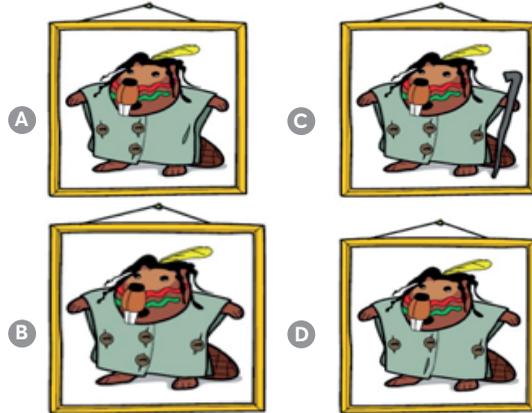
Como observamos, cada una de las cuatro expresiones anteriores puede tener un valor de verdad asociado (verdadero o falso). Ese valor dependerá de cada situación particular donde se aplique un proceso de razonamiento.

En computación, la lógica se utiliza tanto en el análisis como en la resolución de problemas, por ejemplo, al momento de diseñar reglas para clasificar o para tomar decisiones. **El razonamiento lógico ayuda a explicar por qué sucede algo.**

Esto es muy importante en ciencias de la computación: las computadoras son predecibles en sus resultados ya que solo realizan aquello para lo cual están programadas. En virtud de esta cualidad, se utiliza el razonamiento lógico para programarlas y así describir con exactitud las tareas por realizar.

En las imágenes siguientes, se presenta un ejemplo de lógica. Se debe elegir un cuadro de acuerdo con dos condiciones que deben cumplirse a la vez:

- 1) que el personaje de la foto no tenga ningún bastón;
- 2) que todos los botones de su saco estén abrochados.



Aquí tenemos dos expresiones («sin bastón» y «botones abrochados»), las cuales, en conjunto, deben resultar verdaderas. Ahora, para resolver la tarea, debemos observar los cuatro cuadros y determinar en cuál de ellos se cumplen ambas expresiones (es decir, son verdaderas). Al observar con detalle, vemos que el cuadro C es la respuesta.

	Sin bastón	Botones abrochados		Sin bastón	Botones abrochados
A 	Verdadero	Falso	C 	Verdadero	Verdadero
B 	Falso	Verdadero	D 	Verdadero	Falso

Este ejemplo está relacionado con la lógica booleana: hay dos condiciones (las llamaremos p y q) y ambas deben cumplirse, por lo que se aplica el operador booleano Y de conjunción (también escrito con el símbolo \wedge), que indica que, a partir de dos entradas verdaderas, se obtiene una salida verdadera.

p	\wedge	q
V	V	V
V	F	F
F	F	V
F	F	F

DESAFÍO 8. El algoritmo

PREGUNTA :

¿podés escribir el algoritmo que corresponde al problema «Resultados revueltos»? Recordá utilizar instrucciones simples en lenguaje imperativo.

DESAFÍO 8 (yapa). ¿Hay otro camino?

POCIONES MÁGICAS

TAREA 9

TEMAS: pensamiento algorítmico, abstracción y evaluación.

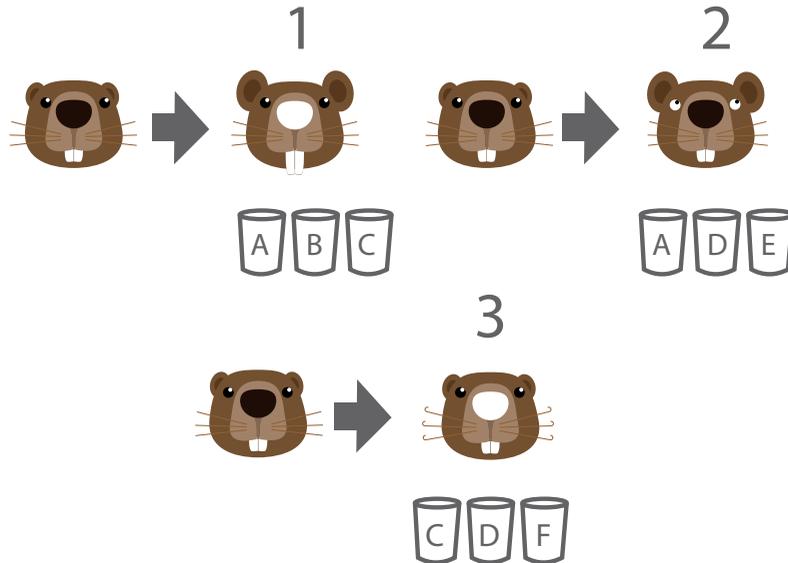
El mago Marcos ha descubierto cinco pociones mágicas nuevas:

- ▶ Una hace que las orejas se alarguen.
- ▶ Otra hace que los dientes se alarguen.
- ▶ Otra hace bigotes rizados.
- ▶ Otra hace que la nariz se vuelva blanca.
- ▶ Y la última hace que los ojos se vuelvan blancos.

El mago vertió cada poción mágica en un vaso distinto y puso agua pura en otro vaso, así que hay seis recipientes en total.



El problema es que el mago hizo tres experimentos con tres castores diferentes y se olvidó de registrar cual poción está en cada vaso.



PREGUNTA :

¿qué vaso contiene agua pura?

TAREA 9. Pociones mágicas

RESPUESTA

Vaso D.

EXPLICACIÓN

Este problema usa elementos de la teoría de conjuntos y la lógica. Una manera de abordar la solución es a partir de analizar en qué experimentos intervino el vaso de agua. Esto puede hacerse a partir de buscar los resultados donde solamente hubo dos efectos (experimentos 2 y 3). Al observar la composición de los dos experimentos, vamos a descubrir que el vaso que se repite es el D, por lo cual inferimos que es el que contiene agua.

PARA SABER MÁS

En este problema tenemos una recopilación de datos que nos ayuda a resolverlo usando la **abstracción** y el **razonamiento lógico**. Como se ha dicho, las computadoras son máquinas de comportamiento predecible: hacé una prueba y configurá dos computadoras de la misma manera, dándoles el mismo programa y los mismos datos de entrada. Se puede garantizar que obtendrás la misma salida de información. Esto significa que las computadoras trabajan exactamente tal como se las programa, no hay factores intrínsecos que las hagan funcionar de manera diferente según se sientan o cuán cansadas estén (como les sucede a las personas).

Las computadoras procesan información sobre la base del uso intensivo de la lógica en sus operaciones internas. En el corazón de cualquier computadora (sea PC, tableta, teléfono inteligente, etc.), existe un componente denominado unidad central de procesamiento (UCP). Cada instrucción de trabajo que procesa una computadora se reduce a un conjunto de **operaciones aritmético lógicas**. Como se ha dicho, todo lo que realiza una computadora está controlado por la lógica, lo cual nos indica que debemos saber cómo trabajar con ella para poder resolver problemas, desarrollar algoritmos que aporten soluciones y luego, opcionalmente, traducirlos a programas en algún lenguaje informático que la computadora sea capaz de interpretar y ejecutar.

El razonamiento lógico en la escuela es utilizado más allá de las computadoras:

- ▶ En ciencia, se debe explicar cómo se ha llegado a conclusiones a partir de experimentos realizados.
- ▶ En historia, se deben discutir las conexiones lógicas de causa y efecto.
- ▶ En tecnología, se reflexiona sobre qué material es el más adecuado para cada parte de un proyecto.
- ▶ En discusiones sobre filosofía, se usa el razonamiento lógico para analizar argumentos.

DESAFÍO 9. ¿Qué hay en los vasos?

¿Podés identificar qué contienen los otros vasos?

DESAFÍO 9 (yapa). Nuevas claves

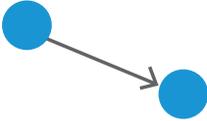
En una sala de computadoras de una escuela, los estudiantes tienen que establecer nuevas contraseñas a sus cuentas personales. Se les permite utilizar letras minúsculas y mayúsculas y los dígitos del 0 al 9. Cada contraseña debe tener un orden para ser aceptada.

a-z: cualquier letra minúscula del alfabeto.

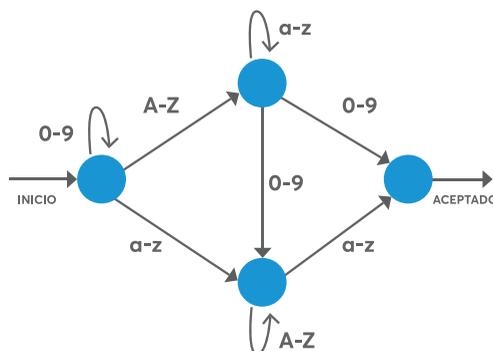
A-Z: cualquier letra mayúscula del alfabeto.

0-9: cualquier dígito.

Hay ciertas reglas que deben cumplirse para que la contraseña sea aceptada.

 <p>A-Z</p>	<p>En un bucle cualquiera cantidad de letras o números se pueden utilizar varias veces. Este bucle, a los estudiantes, les permite usar ninguna, una o varias letras mayúsculas.</p>
 <p>a-z</p>	<p>Una rama implica que los estudiantes tienen que utilizar exactamente una letra o un dígito. Por ejemplo, esta rama indica que se utiliza una minúscula.</p>

En el siguiente esquema, están todas las reglas necesarias para construir las claves seguras en la institución. Cada nodo y arista representa una regla y a la vez indican un camino en la construcción de la contraseña.



Por ejemplo la clave 02Aabc9 es una clave válida porque responde a uno de los caminos de reglas. Comienza con un dígito, luego tiene una letra mayúscula, tres letras minúsculas y finaliza con un número (en el gráfico, correspondería a tomar el camino superior).

PREGUNTA :

¿Cuáles de las siguientes contraseñas no serán aceptadas siguiendo las reglas determinadas por el esquema anterior?

1. Peter3ABCd
2. bENNOZzz
3. 2010Beaver4EVER
4. 123aNNa

INVITADOS A LA FIESTA

TAREA 10

TEMAS: abstracción y descomposición.

Para organizar una cena, Sara necesita hablar con cinco amigos: Alicia, Beti, Carolina, David y Emilio.

Sara puede hablar con Emilio de inmediato. Sin embargo, para hablar con sus otros amigos, hay algunos puntos que considerar:

1. Antes de hablar con David, debe hablar con Alicia.
2. Antes de hablar con Beti, debe hablar con Emilio.
3. Antes de hablar con Carolina, debe hablar con Beti y David.
4. Antes de hablar con Alicia, debe hablar con Beti y Emilio.

PREGUNTA :

¿en qué orden debería Sara hablar con todos sus amigos si quiere hablar con todos ellos?

TAREA 10. Invitados a la fiesta

RESPUESTA

Emilio - Beti- Alicia - David - Carolina

EXPLICACIÓN

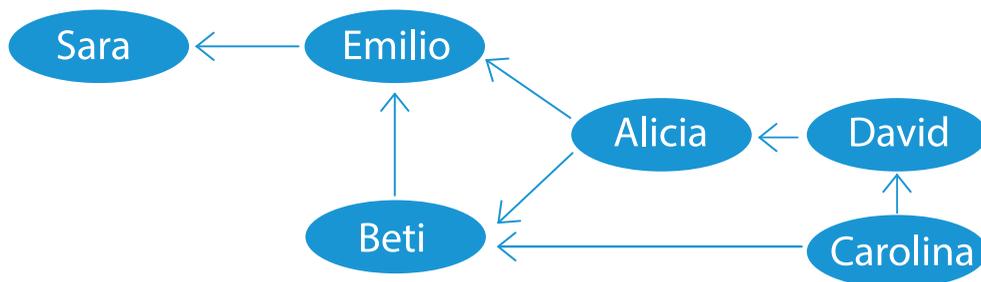
Esta tarea tiene que ver con las dependencias, veamos:

- ▶ Emilio es el único amigo sin dependencias, por lo que debe ir primero.
- ▶ Beti solo depende de Emilio, por lo que va segunda.
- ▶ Alicia depende de Beti y Emilio, por lo que va en tercer lugar.
- ▶ David depende de Alicia, por lo que va a continuación.
- ▶ Finalmente, Carolina depende de Alicia y David, por lo que va en la última posición.

Ninguna otra secuencia puede cumplir las dependencias.

Satisfacer dependencias es un problema computacional que se trata con frecuencia. Este tipo de problemas implica trabajar con reglas de dependencia y orden, las cuales se pueden representar con un esquema llamado grafo, que simplifica el análisis y la comprensión de la situación.

Un grafo se compone de dos elementos básicos: nodos (en este caso representando los amigos) y aristas, que unen los nodos entre sí por alguna relación definida. En este caso, la arista (dirigida) significa X requiere hablar primero con Y. Así, según el siguiente grafo, Alicia necesita primero que se hable con Emilio y con Beti (la arista que parte de su nodo los apunta a ellos) y, a su vez, David requiere que se hable primero con Alicia (la arista del nodo de David llega al nodo de ella).



Una representación del problema utilizando grafos.

PARA SABER MÁS

Cuando se resuelven problemas, al proceso de dividir uno en partes más pequeñas (y, por ende, más manejables) se lo conoce como *descomposición*. Descomponer un problema es una tarea **que facilita su resolución debido a que reduce su complejidad**.

Para descomponer un problema, hay que pensar en términos de partes y componentes, donde cada parte se debe comprender, evaluar y solucionar por separado. Por otro lado, la solución asociada a cada una de ellas puede encargarse a una persona o a un equipo, lo que permite resolver problemas complejos en tiempos más acotados. Toda descomposición implica:

- ▶ identificación de las partes de algo,
- ▶ división de ese algo en partes más pequeñas.

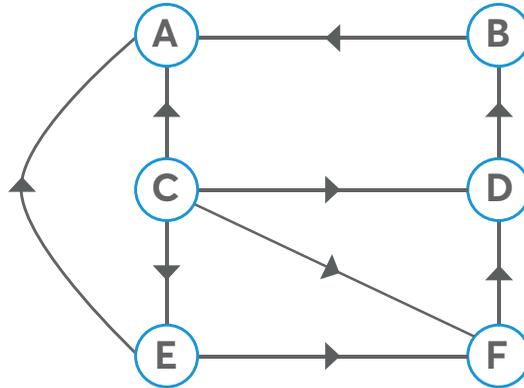
Un ejemplo de cómo se suele aplicar la técnica de la descomposición es la actividad que realizan los detectives policiales cuando se analiza la escena de un crimen. Esclarecer un crimen es un proceso complejo; las tareas asociadas son muchas y, en general, son llevadas a cabo por un equipo de agentes. Por eso, en general, ante un hecho delictivo la técnica policial es dividir el trabajo de investigación a partir de una serie de preguntas clave (las cuales se podrían ver como subproblemas):

- ▶ ¿Qué tipo de crimen se ha cometido?
- ▶ ¿Cuándo se llevó a cabo?
- ▶ ¿Dónde se cometió?
- ▶ ¿Qué evidencias hay?
- ▶ ¿Hubo testigos?
- ▶ ¿Qué observaron?
- ▶ ¿Quién era la víctima?
- ▶ ¿Hay semejanza con otros crímenes registrados?

Luego de obtener las respuestas, el equipo policial puede evaluarlas en su conjunto y empezar a pensar en líneas de investigación a partir de indicios concretos. Como se observa en el ejemplo, la **descomposición del problema** y la generación de tareas de **menor complejidad** permiten empezar una resolución efectiva y lógica del problema.

DESAFÍO 10. Manos y contramanos

La Municipalidad de la Ciudad Ballena Feliz ha indicado la mano de todas las calles entre A, B, C, D, E y F (calles de única mano). Sin embargo, algunas de ellas apuntan en el sentido equivocado.



PREGUNTA :

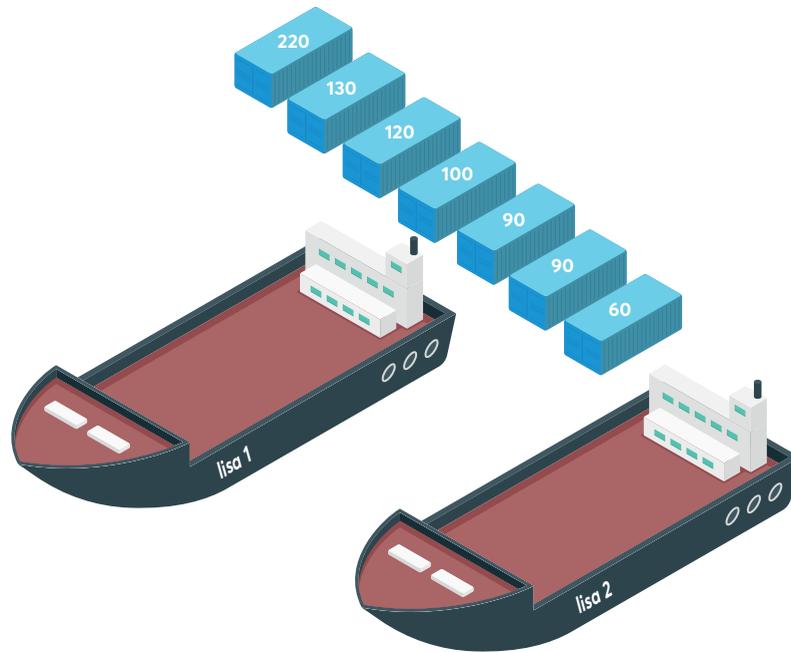
Para asegurarnos de que se pueda llegar a cada punto desde cualquier otro punto, ¿cuál es el número mínimo de calles que deben invertir su dirección?

CARGANDO LOS BOTES

TAREA 11

TEMAS: descomposición y evaluación.

Carlos posee dos botes, llamados Lisa 1 y Lisa 2. Cada embarcación puede llevar una carga máxima de 300 kg. Carlos recibe barriles llenos de pescado para que los transporte; en cada uno de ellos, hay un número que indica su peso en kilogramos.



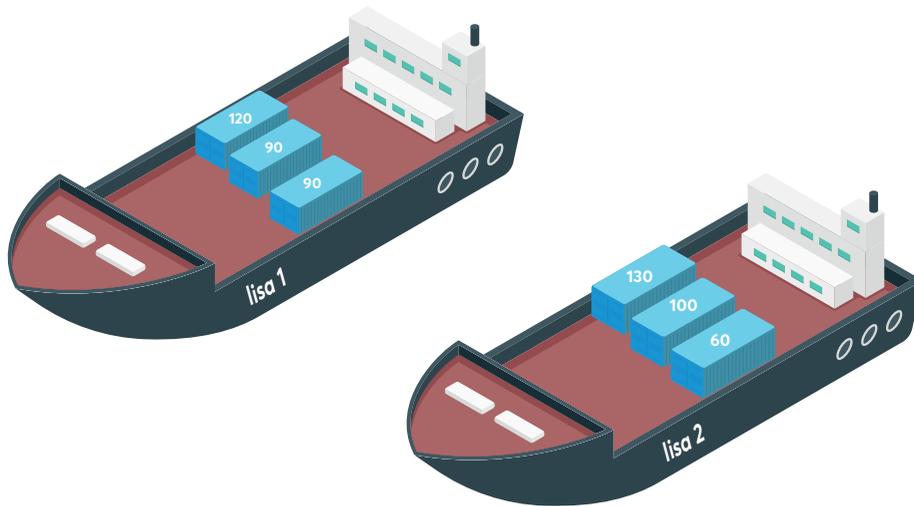
PREGUNTA :

¿cuál es la mejor distribución de la carga para que ningún bote lleve sobrepeso?



TAREA 11. Cargando los botes

RESPUESTA



EXPLICACIÓN

Es posible obtener los barcos cargados con 590 kilos:

$120 + 90 + 90 = 300$ kilos en un bote, $130 + 100 + 60 = 290$ kilos en el otro.

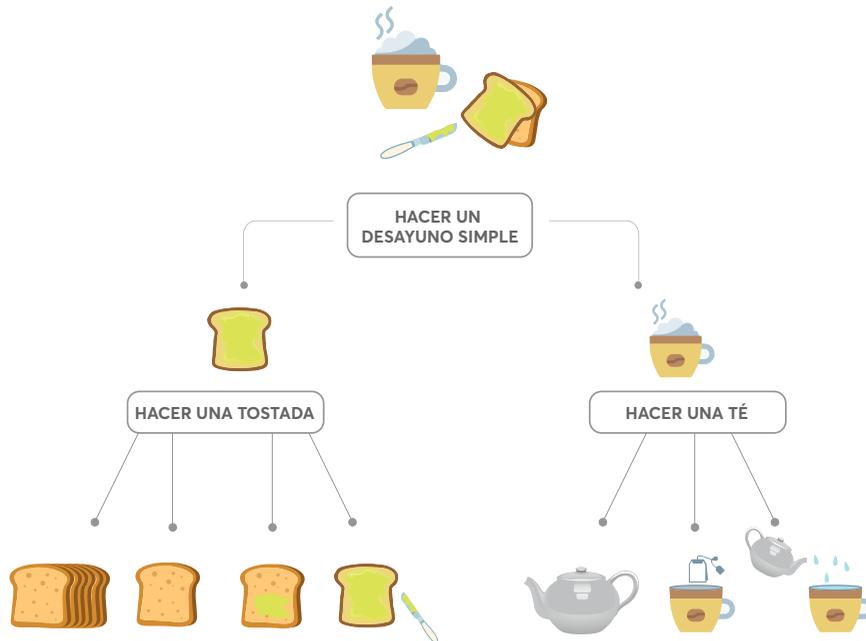
Si intentás cargar barriles pesados primero, terminarás con $220 + 60 = 280$ kilos y $130 + 120 = 250$ kilos, lo que representa un total de 530 kilos.

Además, no es posible llevar más de 590 kg. De hecho, si se quisiera llevar más, habría que llenar ambos barcos con 300 kg, pero hay una forma de hacerlosolo en uno de ellos: $120 + 90 + 90$.

PARA SABER MÁS

Como se ha visto, el proceso de descomponer un problema en partes más pequeñas y manejables ayuda a resolver cuestiones complejas y a gestionar proyectos de envergadura. En general, los grandes problemas son desalentadores y es más fácil trabajar con un conjunto de tareas más pequeñas y relacionadas entre sí. Esto posibilita que cada subtarea puede ser abordada por personas o equipos de trabajo, donde cada uno aporta sus propios conocimientos, experiencia y habilidades.

En el gráfico siguiente, se presenta un ejemplo simple de descomposición.



Puede observarse, de manera clara, la descomposición de la situación problema a través de la generación de tareas de menor complejidad.

También en la escuela se da el proceso de descomposición de tareas, por ejemplo:

Al desarrollar un plan que permita implementar buenas prácticas ecológicas y energéticas en una escuela. Tareas: definir una estrategia para almacenar y procesar la basura, definir un plan para reducir el consumo de electricidad, planificar una campaña de concientización y capacitación en la comunidad escolar.

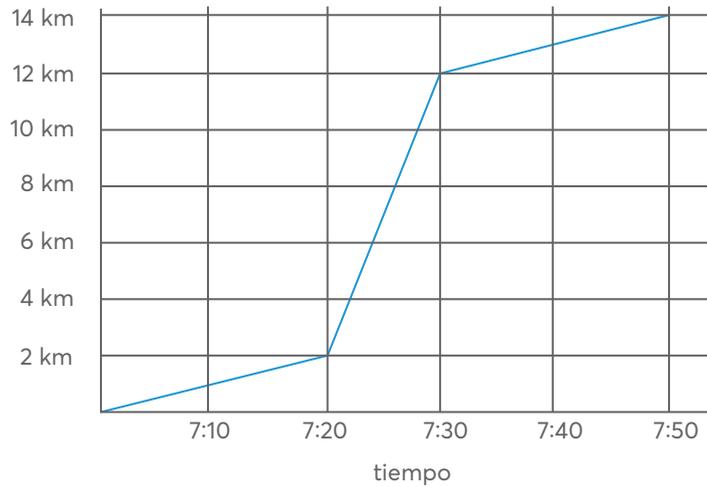
Al planificar una revista institucional. Tareas: identificar temas y secciones, definir los roles de los colaboradores y responsabilidades asociadas, planificar tiempos y recursos necesarios a los efectos de llevar adelante el proyecto.

Quando se organiza una fiesta escolar, se realiza una planificación de las actividades; en particular, se definen tareas como diseño del evento, definición de partes, selección de actores, definición de roles y responsabilidades, necesidades de logística, etc.

Como se ha podido observar, detrás de estos ejemplos hay una serie importante de tareas asociadas. También resulta evidente que la descomposición de un problema en partes más pequeñas no es un proceso exclusivo de la informática, también es común en otros ámbitos como, por ejemplo, la ingeniería, el diseño y la gestión de proyectos.

DESAFÍO 11. El viaje

Todos los días Belen sale de su casa y camina hacia la estación de tren, luego toma un tren hasta una estación cercana a su escuela y, finalmente, camina hacia esta. Su progreso se registra en el siguiente gráfico:



PREGUNTA :

- a) ¿A cuántos kilómetros de distancia se halla su escuela?
- b) ¿Qué tan rápido (en km/h) camina Belén?
- c) ¿Cuál es la velocidad media (en km/h) del tren?

AGENTES SECRETOS

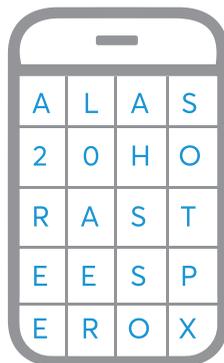
TAREA 12

TEMAS: evaluación y generalización.

Los agentes Boris y Berta se comunican mediante mensajes secretos. Boris quiere enviarle a Berta el siguiente mensaje:

ALAS20HORASTEESPEROX

Boris escribe cada carácter en una cuadrícula de 4 columnas de izquierda a derecha y fila por fila a partir de la parte superior. Pone una X en la casilla no utilizada. El resultado se muestra a continuación:



A	L	A	S
2	O	H	O
R	A	S	T
E	E	S	P
E	R	O	X

Luego crea el mensaje secreto leyendo los caracteres de arriba a abajo y columna por columna, comenzando desde la izquierda:

A2REELOAERAHSSOSOTPX

Berta luego usa el mismo método para responder a Boris. El mensaje secreto que ella le envía es:

OSELNKTEPT-ANUEEREEX

PREGUNTA :

¿Qué mensaje le devuelve Berta a Boris?

TAREA 12. Agentes secretos

RESPUESTA

OK-ESTAREENELPUENTEX

EXPLICACIÓN



Luego la lectura de izquierda a derecha y fila por fila da:

OK-ESTAREENELPUENTEX

PARA SABER MÁS

Una tarea de **evaluación** implica hacer juicios sobre algo de una **manera objetiva** y sistemática. La evaluación es algo que realizamos cotidianamente, haciendo juicios sobre qué hacer y pensando en función de una serie de factores que son parte de un contexto. Evaluamos algo en estas situaciones:

- ▶ cuando cocinamos y probamos nuestros platos, para ajustar su sabor y verificar el estado de cocción;
- ▶ cuando estamos en un nivel de un videojuego, queremos pasar al siguiente y probamos y evaluamos distintas estrategias de acción que nos permitan seguir adelante;
- ▶ cuando recibimos la devolución de un docente de un trabajo estudiantil y revisamos las notas que añadió para saber qué estamos haciendo bien y qué cosas aún nos falta aprender o hemos comprendido de forma errónea.

En el marco del pensamiento computacional, una vez que se ha diseñado **una**

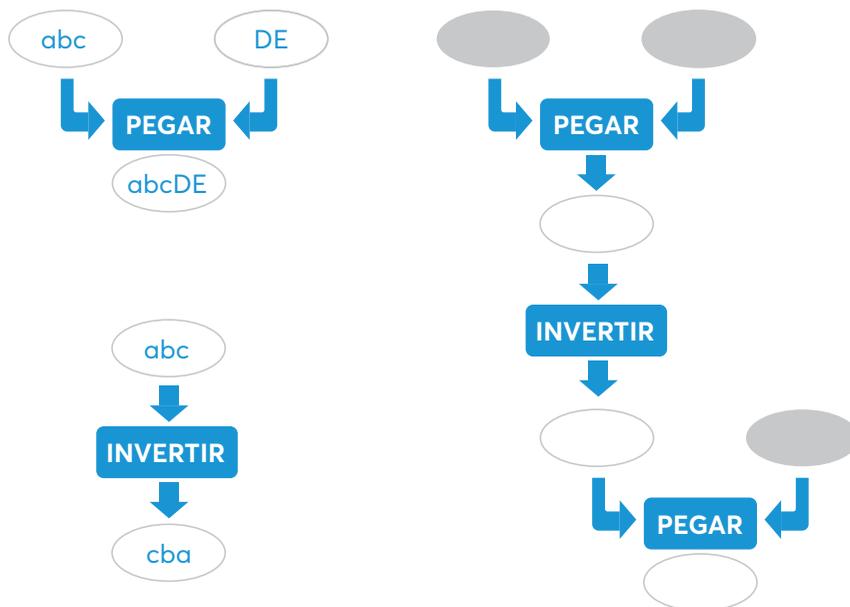
solución, es necesario asegurarse de que sea adecuada para su propósito. La evaluación es el proceso que se aplica a una solución en pos de asegurar que responde a los requerimientos de diseño y que, además, funciona correctamente, sin errores.

Esta tarea está relacionada con temas de seguridad informática. No siempre queremos que los mensajes enviados a través de las redes se lean si son interceptados. Estos mensajes pueden contener una contraseña u otra información privada. Por lo tanto, un mensaje a veces se **cifra**, lo que significa que se convierte en un mensaje secreto. Para que esto funcione, el destinatario debe poder **descifrar** este mensaje secreto y descubrir el mensaje original. Sin embargo, no debería ser posible que alguien ajeno que encuentre el mensaje secreto también recupere el original.

Hay muchos tipos de cifrado, el que se usa en este ejemplo se denomina **cifrado de transposición** ya que cambia de filas a columnas y de columnas a filas cuando el mensaje se coloca en una cuadrícula. El estudio de estos temas se denomina **criptografía** y es un área de investigación moderna que toma importancia a partir del avance de las tecnologías de la comunicación. Su desarrollo implica el trabajo con problemas matemáticos complejos.

DESAFÍO 12. La máquina de pegar e invertir

Tenemos dos tipos de máquinas para procesar textos. La máquina Pegar toma dos trozos de texto y los junta uno con otro (dibujo superior izquierdo). La máquina Invertir toma un trozo de texto y lo invierte (dibujo inferior izquierdo).



Combinando dos máquinas Pegar y una máquina Invertir (como las de la columna izquierda del dibujo), obtenemos una máquina de texto más compleja (la de la columna derecha). Nuestra nueva máquina de texto compuesta necesita tres textos para trabajar, señalados con elipses grises, es decir, para procesarlos y producir un texto en la elipse inferior de salida.

PREGUNTA :

¿qué textos deben ser introducidos en la máquina si queremos que el resultado que produzca sea la palabra PREGUNTA?

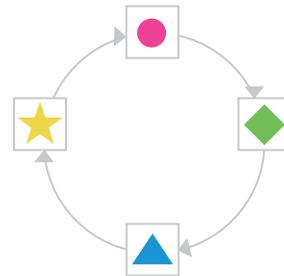
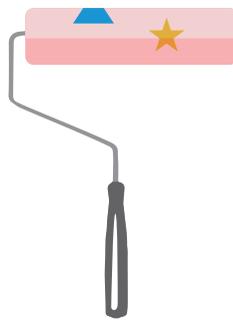
1. GERP- NU -TA
2. TA - GERP- NU
3. GERP- TA - NU
4. NU-GERP -TA

EL PINTOR DE PAREDES

TAREA 13

TEMAS: evaluación, abstracción y generalización.

Un pintor ha comprado un rodillo con algunas propiedades particulares. El rodillo reemplaza una forma existente en una pared por otra. El siguiente gráfico muestra cuáles cambios realiza.



Por ejemplo, cuando el pintor usa el rodillo para pintar una pared, que tiene las figuras que se ven a la izquierda, obtiene la pintura de la derecha:



PREGUNTA :

¿cómo se verá la siguiente pintura después de usar el rodillo mágico?



RESPUESTA



EXPLICACIÓN

Para resolver la tarea hay que aplicar un algoritmo. La secuencia B es la que queda después de aplicar el rodillo mágico. Con respecto a las otras secuencias, la A y la C tienen un segundo símbolo incorrecto, que debe ser un círculo, y solo los primeros dos símbolos de la D son correctos.

PARA SABER MÁS

Cuando una solución se traduce a un algoritmo, **deberían verificarse una serie de elementos** que son parte de una prueba de **evaluación**:

- ▶ Que se entienda fácilmente: ¿está completamente descompuesto?
- ▶ Que sea eficaz: ¿resuelve el problema?
- ▶ Que sea eficiente: ¿resuelve el problema, haciendo el mejor uso posible de los recursos disponibles?

Cuando un algoritmo pasa el proceso de evaluación y cumple con los tres criterios anteriores, se entiende que se está ante una solución correcta; por lo tanto, puede avanzar a la etapa de programación en algún lenguaje informático. Programar sin antes evaluar, la mayoría de las veces dificulta la programación, lo cual puede llevar a más errores y, por ende, a mayores costos y tiempos en el proyecto.

En informática, la evaluación de los programas es sistemática y rigurosa: se trata de juzgar la calidad, la eficacia y la eficiencia de las soluciones, sistemas, productos y procesos. La evaluación comprueba que las soluciones hacen el trabajo para el que están diseñadas y son adecuadas para su propósito.

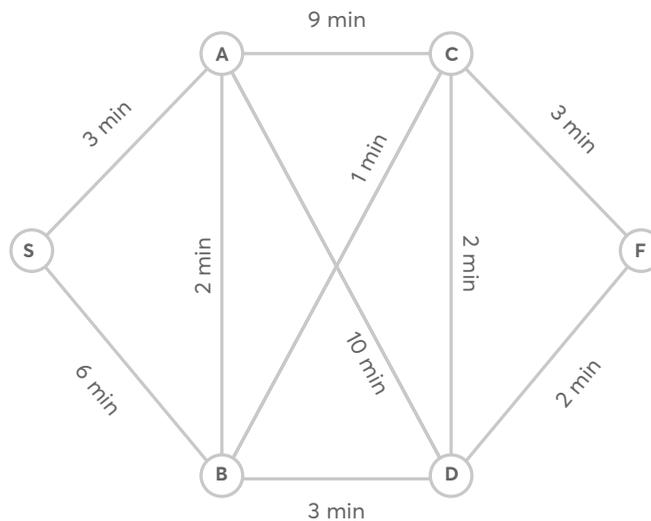
¿Cómo evaluamos nuestra solución? Hay muchos métodos desarrollados. En principio y a modo de ejemplo, se podría comenzar la tarea con la orientación de las siguientes tres preguntas:

- ▶ ¿Se comprende, de manera completa, cómo se ha resuelto el problema? Es decir, si la solución construida está explicitada en su totalidad. Si algo no se sabe todavía, entonces no se tiene una solución completa. En caso de que no se sepa claramente cómo hacer algo para resolver el problema, hay que volver a la etapa anterior para verificar que todo se haya descompuesto correctamente y que cada parte tenga una solución.

- ▶ ¿La solución cubre todas las partes del problema? Aquí se busca validar que la solución propuesta y desarrollada satisfaga plenamente el objetivo por cumplir y que, además, lo haga en el marco de las restricciones impuestas.
- ▶ ¿La solución optimiza la repetición de tareas? En caso de respuesta negativa, se debe preguntar si existe alguna forma de reducir tal repetición. Para esto, hay que regresar a la etapa de desarrollo de la solución y eliminar las repeticiones innecesarias.

DESAFÍO 13. ¿Por dónde voy?

La ruta más corta no siempre es la más rápida, por lo menos así lo indican los atascos de tránsito que producen congestiones importantes.



PREGUNTA :

¿cuál es la ruta desde S a F que insumirá menos tiempo?

CAMINANDO ÁRBOLES

TEMAS: pensamiento algorítmico, abstracción y descomposición.

Las instrucciones para armar un árbol n°1 son:

1. Da un paso adelante y hacé una huella, luego volvé a la posición de inicio.



Cuando sabés cómo hacer un árbol n°1, podés aprender a hacer un árbol n°2:

1. Da 2 pasos hacia adelante para hacer dos huellas.
2. Girá a la izquierda y hacé un árbol n°1.
3. Girá a la derecha y hacé un árbol n°1.
4. Volvé al inicio.



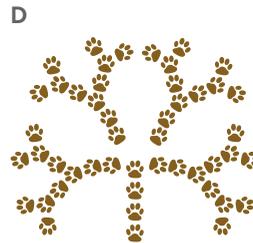
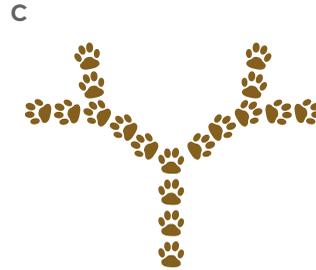
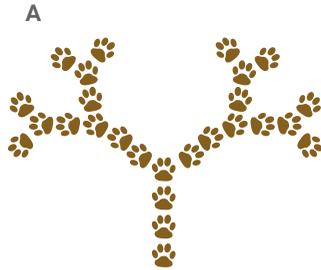
Y, si sabés hacer un árbol n°2, podés hacer un árbol n°3:

1. Da 3 pasos hacia adelante para hacer tres huellas.
2. Girá a la izquierda y hacé un árbol n°2.
3. Girá a la derecha y hacé un árbol n°2.
4. Volvé al inicio.



PREGUNTA :

De manera similar podés crear un árbol n°4. ¿Cuál de los siguientes árboles es un árbol n°4 ?



TAREA 14. Caminando árboles

RESPUESTA

Árbol A.

EXPLICACIÓN

El árbol A es de tipo árbol₄, hecho de 4 pasos más dos estructuras árbol₃.

El esquema es tan compacto como inteligente y funciona para cualquier tamaño imaginable de un árbol.

Árbol X = X pasos hacia adelante +2 árboles (X-1)

Árbol X-1 = X-1 pasos hacia adelante +2 árboles (X-2)

En el pensamiento computacional, cuando una tarea se explica a través de una versión más simple de la tarea en sí se denomina **recursión**. Finalmente, la recursión reduce la tarea a una tarea base, como en este caso «generar un árbol 1», con lo cual, para hacer un árbol X, solo tenés que recordar la forma de combinar los árboles (X-1) y cómo hacer un árbol 1.

PARA SABER MÁS

Las tareas y los desafíos que has realizado te han permitido desarrollar capacidades para comprender y tratar de resolver problemas de una mejor manera. Has trabajado con elementos del pensamiento computacional, que es una herramienta mental que te ayuda a abordar, de manera creativa, tareas o situaciones que involucren problemas.

En particular, en las tareas anteriores, has desarrollado una serie de capacidades:

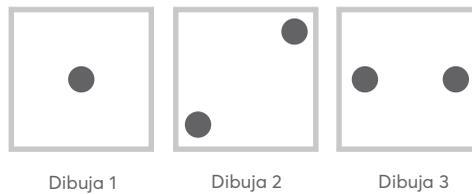
- ▶ pensar de forma algorítmica, para poder comunicar una solución a un problema;
- ▶ pensar en términos de descomposición, para dividir un gran problema en pequeños
- ▶ problemas de menor complejidad;
- ▶ pensar en generalizaciones, al identificar y usar patrones, para ofrecer soluciones más eficientes;
- ▶ pensar en términos abstractos, para elegir las mejores representaciones, es decir, aquellas que resalten la información de interés y oculten los detalles sin importancia;

- ▶ pensar en términos de evaluación, para saber si algo funciona y si puede ser mejorado en términos de rendimiento.

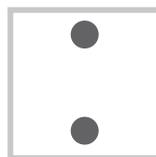


DESAFÍO 14. Datos programados

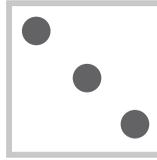
Tenemos tres instrucciones, `dibuja_1`, `dibuja_2` y `dibuja_2A`, para obtener los siguientes modelos:



La instrucción `gira_90` gira el dibujo hecho. Por ejemplo, si utilizamos juntas las instrucciones `dibuja_2A` y `gira_90`, conseguiremos este dibujo:

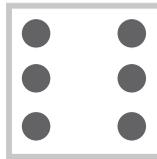


Juntando las instrucciones se pueden hacer combinaciones distintas de puntos. Por ejemplo, las instrucciones `dibuja_1`, `dibuja_2` y `gira_90` dan el siguiente resultado:



PREGUNTA :

¿Cuál es la sucesión de instrucciones que se ha utilizado para dibujar esta figura?



- 1: `dibuja_2A`, `gira_90`, `dibuja_2`, `dibuja_1`
- 2: `dibuja_2A`, `dibuja_2`, `gira_90`, `dibuja_2`
- 3: `dibuja_2`, `dibuja_2A`, `gira_90`, `dibuja_2`
- 4: `dibuja_2`, `gira_90`, `dibuja_2`, `dibuja_2A`

DESAFÍOS

RESPUESTAS

DESAFÍO 1. Reentrenando al ratón

RESPUESTA

1. Avanzá.
2. Cada vez que encuentres un túnel a la izquierda, debés atravesarlo.
3. Volvé al paso 1.

-

DESAFÍO 2. El robot en su laberinto

RESPUESTA

A1, D, A3, I, A2, I, A2, I, A2, D, A2, D, A1, T.

-

DESAFÍO 3. Escribir el algoritmo I

RESPUESTA

Las instrucciones que se repiten son:

se comienza por levantar esa parte del cuerpo situada a la derecha abajo, envuelta casi siempre en cuero o gamuza, y que salvo excepciones cabe exactamente en el escalón. Puesta en el primer peldaño dicha parte, que para abreviar llamaremos pie, se recoge la parte equivalente de la izquierda (también llamada pie, pero que no ha de confundirse con el pie antes citado), y llevándola a la altura del pie, se le hace seguir hasta colocarla en el segundo peldaño, con lo cual en este descansará el pie, y en el primero descansará el pie.

Esto, expresado en instrucciones de un algoritmo, sería:

- Levantá el pie derecho.
- ▶ Apoyá el pie derecho en el escalón superior.
- ▶

- ▶ Levantá el pie izquierdo.
- ▶ Apoyá el pie izquierdo en el escalón inmediatamente superior al del pie derecho.

DESAFÍO 3 (yapa). Escribir el algoritmo II

RESPUESTA

El algoritmo que debe seguir el robot para subir la escalera contiene dos ciclos. En el primero, sube de la planta inferior al descanso y, en el segundo ciclo, al piso superior. Nótese que las instrucciones de ambos ciclos son las mismas.

Mientras no hayas llegado al descanso,

- ▶ (levantá la pata derecha y apoyala en el escalón superior,
- ▶ levánta la pata izquierda y apoyala en el escalón inmediatamente superior al de la pata derecha).

Posícinate al inicio del segundo tramo de escalera.

Mientras no hayas llegado al piso superior,

- ▶ (levantá la pata derecha y apoyala en el escalón superior,
- ▶ levánta la pata izquierda y apoyala en el escalón inmediatamente superior al de la pata derecha).

-

DESAFÍO 4. Contar cuadrados

RESPUESTA

Los cuadrados son 15.

DESAFÍO 4 (yapa). Próximo vuelo

RESPUESTA

SLA 09 es el código para el vuelo que sale a las 18:00. La parte numérica del código se arma sumando todos los dígitos de la hora de partida.

-

DESAFÍO 5. ¿Dónde están los lagos?**RESPUESTA**

Se deben buscar fuentes de agua cuyas vecinas estén todas a mayor altura. Así, los lagos se formarán en los puntos 7, 10, 13, 14 y 15

DESAFÍO 6. Próximos**RESPUESTA 1**

Son potencias de 2 y el número siguiente es 256.

RESPUESTA 2

Sigue la V dado que se agrega una letra del alfabeto en sentido inverso a la serie con cada repetición a partir de Z.

-

DESAFÍO 6 (yapa). Cinco palitos**RESPUESTA**

La figura D dado que implica más movimientos.

-

DESAFÍO 7. Armar el cubo**RESPUESTA:**

La figura B.

-

DESAFÍO 8. El algoritmo**RESPUESTA**

1. Tomá una casilla verde vacía cuyos competidores estén definidos.
2. Buscá en la lista de tarjetas desordenadas el competidor que debe asignarse a la casilla verde.
3. Tachá de la lista de tarjetas desordenadas el competidor encontrado y colocalo en la casilla vacía.
4. Volvé al paso 1 mientras existan casillas vacías.
5. El ganador es la última casilla definida.

DESAFÍO 8 (yapa). ¿Hay otro camino?**RESPUESTA**

1. Buscá en la lista de tarjetas desordenadas el competidor que más se repite.
2. El ganador es el competidor hallado en el paso 1.

-

DESAFÍO 9. ¿Qué hay en los vasos?**RESPUESTA**

En A, la poción que alarga orejas; en B, la que alarga dientes; en C, la que vuelve blanca la nariz; en D, agua; en E, la que vuelve blancos los ojos; en F, la que riza bigotes.

Para deducir lo anterior, hay que revisar qué vasos y efectos se repiten entre pares de experimentos.

DESAFÍO 9 (yapa). Nuevas claves**RESPUESTA**

La clave 2 no será aceptada porque finaliza con más de una letra minúscula.

-

DESAFÍO 10. Manos y contramanos**RESPUESTA**

Se debe invertir el sentido de tránsito de la calle que conecta las esquinas A y C.

-

DESAFÍO 11. El viaje**RESPUESTA**

La escuela está a 14 km. Belén camina a un ritmo de 6 km por hora. El tren viaja a 60 km por hora.

-

▶

DESAFÍO 12. La máquina de pegar e invertir

RESPUESTA

Los textos a introducir son los de la opción 4.

-

DESAFÍO 13. ¿Por dónde voy?

RESPUESTA

La ruta más corta en tiempo lleva 9 minutos: S, A, B, C, F. Hay otra ruta que involucra menos nodos, pero lleva más tiempo.

-

DESAFÍO 14. Datos programados

RESPUESTA

El algoritmo de la opción 4 es el correcto.

-

REFERENCIAS BIBLIOGRÁFICAS

Aprendemos Juntos (12 de noviembre, 2018). Versión Completa. Enseñar es un arte. Ken Robinson, educador y escritor. [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=WP8WSK-6Pj0>

Azcue, M., Diez, M. L., Lucanera, V., Scandrolí, N., (agosto de 2006). Resolución de un problema complejo utilizando un elemento de naturaleza heurística. *Revista Iberoamericana de Educación*, 37(6). Recuperado de <https://rieoei.org/RIE/article/view/2684/3667>

Bohm, D. (2002). *Sobre la creatividad*. Barcelona: Kairós.

Brennan, K. y Resnick, M. (2012). *Entrevistas basadas en artefactos para estudiar el desarrollo del Pensamiento Computacional (PC) en el diseño de medios interactivos*. Vancouver: American Educational Research Association.

CAS (2015). *Pensamiento Computacional. Guía para profesores*. Computing At School. Recuperado de <http://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Guía-para-profesores.pdf> [En línea 31/10/19]

Cortázar, J. (1962). *Instrucciones para subir una escalera. Historias de cronopios y de famas*.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. y Woollard, J. (2015). *Computational thinking. A guide for teachers*. Recuperado de <https://community.computingatschool.org.uk/files/8550/original.pdf>

De Bono, E. (1991). *Aprender a pensar*. México DF: Editorial Paidós.

Dodero, J. (julio, 2012). *Pensamiento computacional para no informáticos. Re Visión*, 5(1). Recuperado de <https://bit.ly/2QoO3E2> [En línea 29/19/19]

Dostál, J. (julio, 2015). *The definition on the term "Inquiry-based instruction"*. *International Journal of Instruction*, 8.

Fresneda, C. (2016). *El 47 % de los empleos está en "alto riesgo" de ser automatizado*. *El Mundo*. Recuperado de <https://bit.ly/1P4Wcl4> [En línea 29/19/19]

Fundación Sadosky (2013). *CC – 2016. Una propuesta para refundar la enseñanza de la computación en las escuelas argentinas*. Recuperado de <http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf> [En línea 29/10/19]

Gardner H. (2000). *Inteligencia: múltiples perspectivas*. Buenos Aires: Aique.

Guilford, J. (1950). Creativity. *American Psychologic*, 5(9), pp. 444-454.

Guilford, J. (1967). Creativity: yesterday, today and tomorrow. *Journal of Creative Behavior*, 1(1), pp. 3-14.

Gurises Unidos (2017). *Pensamiento Computacional. Un aporte para la educación de hoy*. Fundación Telefónica, Movistar.

Gurstein, M. (2003). Effective use: A community information strategy beyond the Digital Divide. *FirstMonday*, 8(12).

Himanen, P (2002). *La ética del hacker y el espíritu de la era de la información*. Barcelona: Destino.

Hocevar, D. (1980). Intelligence, divergent thinking, and creativity. *Intelligence*, 4(1), pp. 1-95.

ISTE (2011). Operational definition of computational thinking for K-12 education. NSF, CSTA, ISTE. Recuperado de <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf> [En línea 29/10/19]

Johnbo. (2014, 29 de mayo). La enseñanza del pensamiento algorítmico debe empezar en Primaria. Entrevista a Juan Julián Merelo. Genbeta: dev. Recuperado de <https://www.genbeta.com/desarrollo/la-ensenanza-del-pensamiento-algoritmico-debe-empezar-en-primaria-entrevista-a-juan-julian-merelo> [En línea 29/10/19]

Manovich, L. (2013). *El software toma el mando*. Barcelona: UOC.

National Research Council (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. The National Academies Press. Washington D. C.

National Research Council (2011). *Report of a Workshop on the Pedagogical Aspects of Computational Thinking*. The National Academies Press. Washington D. C.

Pólya, G. (1945). *How to Solve It*. Princeton University Press.

Pólya, G. (1969). The goals of mathematical education. Recuperado de <http://blk.mat.uni-bayreuth.de/aktuell/db/20/polya/polya.html> [En línea 31/10/19]

Raja, T. (16 de junio, 2014). Is coding the new literacy? Mother Jones. Recuperado de <https://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education/> [En línea 29/10/19]

Resnick, M. (2002). Rethinking Learning in the Digital Age. En G. Kirkman et al, The Global Information Technology Report 2001-2002. Readiness for the Networked World. Nueva York: Oxford University Press.

Resnick, M. (2007a). Sowing the Seeds for a More Creative Society. Learning and Leading with Technology, December, pp. 18-22.

Resnick, M. (2007b). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. C&C '07 Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition. Washington D. C.

Stager, G. (2014). What's the Maker Movement and Why Should I Care? Scholastic. Recuperado de <http://www.scholastic.com/browse/article.jsp?id=3758336> [En línea 29/10/19]

Sternberg, R.J. (1985). Beyond IQ. A triarchic theory of human intelligence. Cambridge: Cambridge University Press.

Torres, W. (enero-junio, 2013). Incidencia del género en la obtención de metas académicas en estudiantes de grado décimo de dos colegios de la ciudad de Bogotá. El Astrolabio, 12(1). Bogotá, Colombia.

Villafañe, D., Rodríguez, N., Murazzo, M. y Martínez, C. (2013). La importancia de fomentar el pensamiento computacional. Primeras Jornadas Internacionales de Educación con Tecnologías.

WEF (2016). The future of jobs. Employment, skills and workforce strategy for the fourth industrial revolution. World Economic Forum.

Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), pp. 33-35. Recuperado de <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf> [En línea 29/10/19]

Wing, J. (primavera, 2011). Computational thinking - What and why? The Link, (6), pp. 20-23. Recuperado de https://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf

Zapata-Ros, M. (2015). Pensamiento computacional: una nueva alfabetización digital. RED, Revista de Educación a Distancia, 46(4).

EDUCAR S. E. BUENOS AIRES



EDIC.01
ARG.2020



**PROYECTO —
PENSAMIENTO —
COMPUTACIONAL**



www.educar.com.ar

EDUCAR
SOCIEDAD DEL ESTADO

