#SABERES DIGITALES

nuevas formas para comprender y hacer en educación

Estrategias para desarrollar el pensamiento computacional

Colección “Desconectados”

Material destinado a docentes

Alejandro Adrián Iglesias y Fernando Bordignon

alejandro.adrian.iglesias@unipe.edu.ar, fernando.bordignon@unipe.edu.ar

Abril, 2019

unipe Universidad Pedagógica Nacional

www.unipe.edu.ar

Material destinado a docentes

1. Introducción

Este documento es un resumen que parte de un libro que está siendo desarrollado por la Universidad Pedagógica Nacional (UNIPE) en el contexto del programa de investigación “Más allá de las pantallas”. El objetivo de este texto es introducir los temas trabajados en el libro a los docentes interesado en participar de la construcción didácticas para el desarrollo del pensamiento computacional. Se ofrece por lo tanto una rápida introducción a la problemática de la enseñanza del pensamiento computacional a través de actividades desconectadas que no requieran del uso de tecnologías digitales.

En la primera parte de este texto se presentan, en pocas palabras, el concepto de pensamiento computacional y de actividades desconectadas. Es necesario expandir y profundizar estos conceptos, por lo cual se recomienda seguir las lecturas y actividades que se encuentran en los apartados “Para saber más”. Finalmente, en la segunda parte se presentan las actividades desarrolladas por la UNIPE así también como los criterios que se han tenido en cuenta para desarrollarlas.

2. Pensamiento computacional un saber de época.

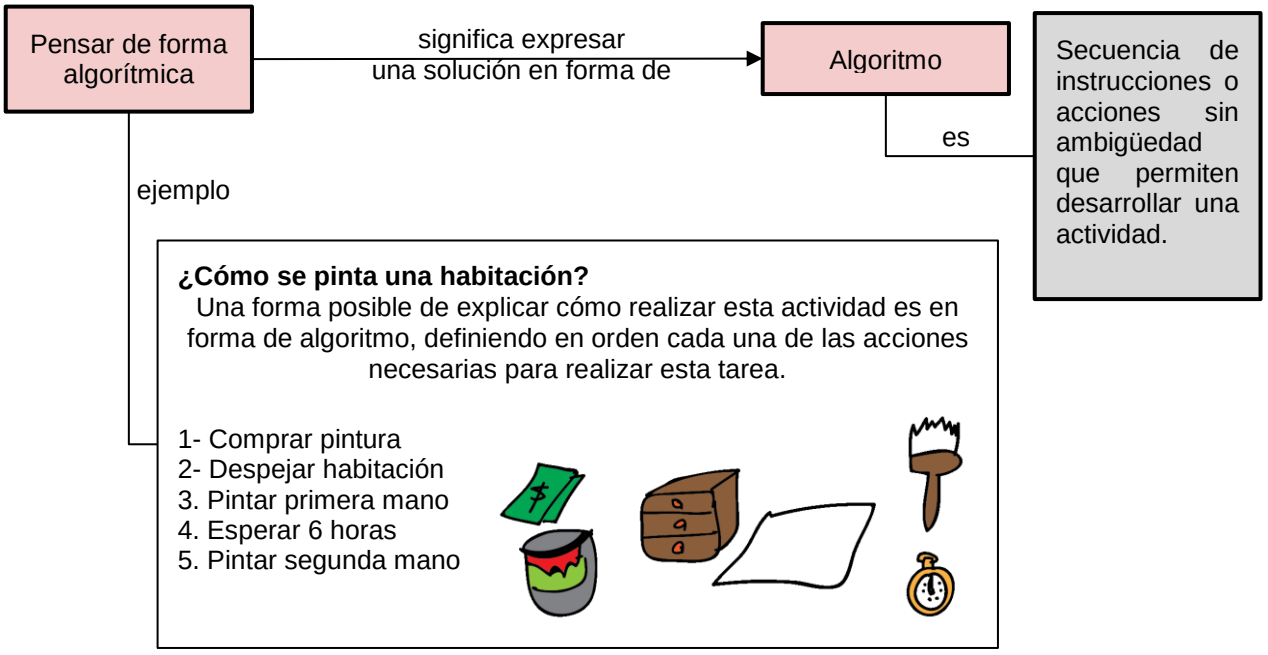
En los últimos tiempos el término “pensamiento computacional” (PC) ha tomado cierta popularidad tanto nacional como internacionalmente. Se lo utiliza para hacer referencia a técnicas y metodologías de resolución de problemas donde intervienen saberes relacionados con la programación de computadoras. Sin embargo, este pensamiento no solamente se lo restringe a problemas informáticos, sino que se lo puede tomar de una manera más amplia, para razonar y trabajar sobre otros tipos de situaciones y áreas de conocimiento.

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:

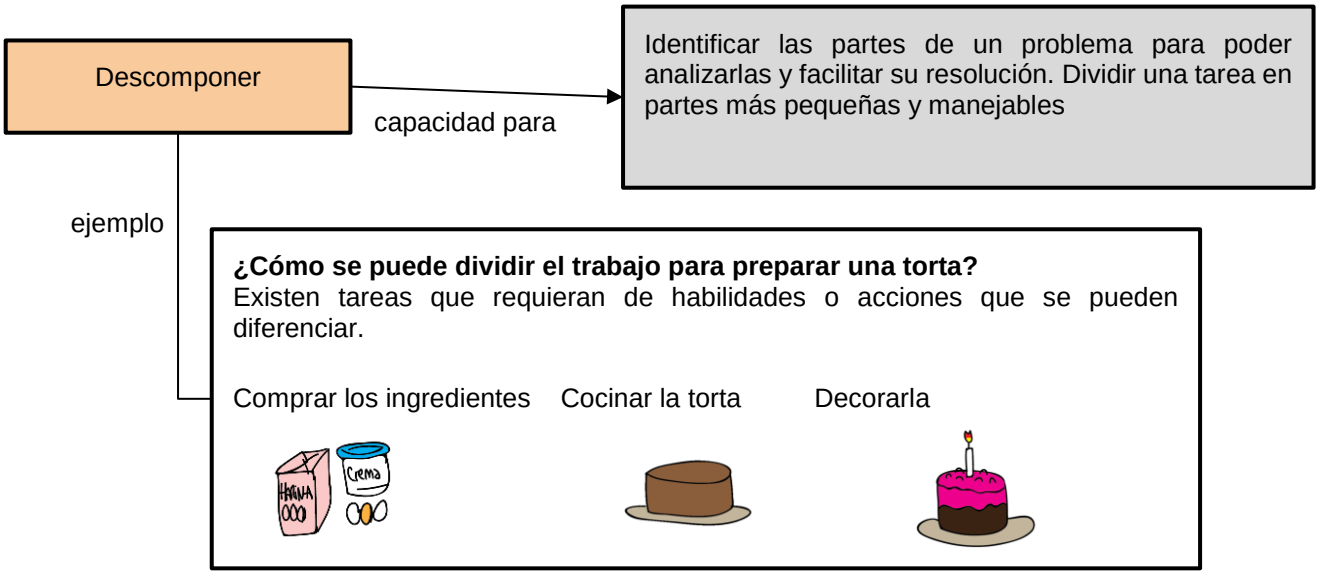
> *"A la lectura, escritura y aritmética, debemos agregar el pensamiento computacional en la habilidad analítica de cada niño. Así como la imprenta facilitó la difusión de la lectoescritura y el conocimiento matemático, […] la computación y las computadoras facilitan la difusión del pensamiento computacional. El pensamiento computacional implica resolver problemas, diseñar sistemas y comprender el comportamiento humano, basándose en los conceptos fundamentales de la informática (Wing, 2006)."*

Las capacidades vinculadas con el pensamiento computacional, entendido como un proceso cognitivo que implica un razonamiento lógico aplicado a la resolución de problemas, se puede resumir en los siguientes puntos (CAS, 2015):

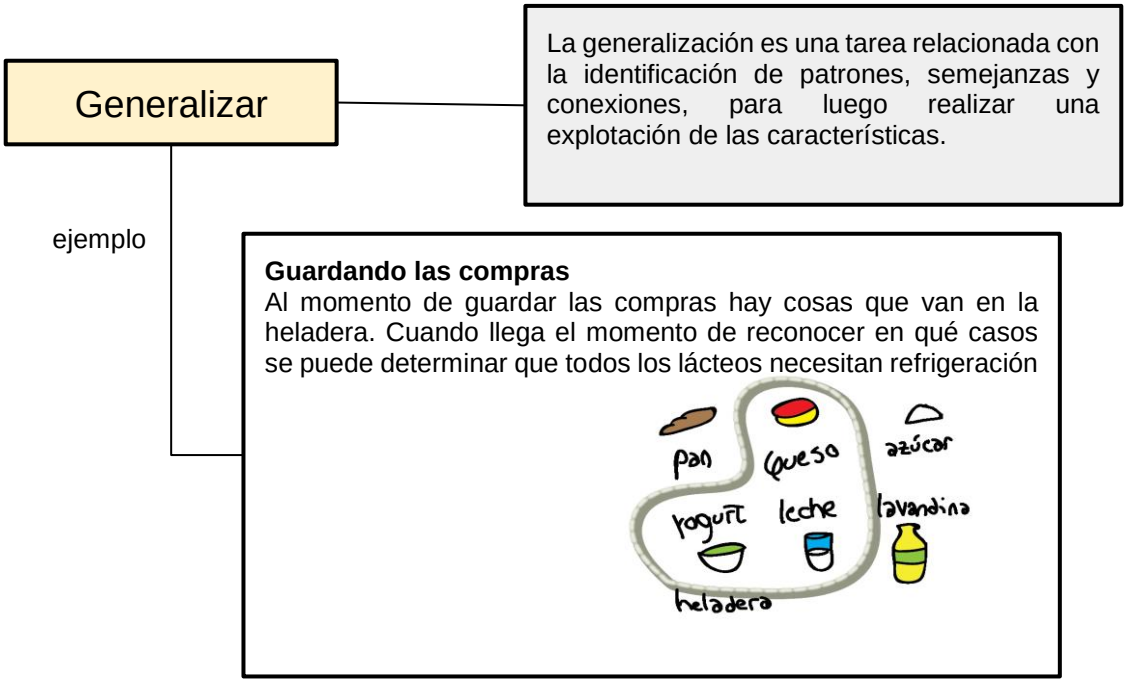
Pensar de forma algorítmica: se refiere a la capacidad para expresar soluciones de forma tal que se componga de una serie de pasos no ambiguos que permita que una persona o un procesador de información pueda ponerlo en práctica.



Descomponer: se refiere a la capacidad de poder dividir e identificar las partes más pequeñas que componen a un problema, haciéndolo más fácil de analizar y solucionar.



Generalizar: se refiere a la capacidad de descubrir patrones en los elementos que componen un problema o en las soluciones que son aplicables a ellas. Esta habilidad guarda una estrecha relación con la abstracción.



Abstraer: se refiere a la capacidad de elegir representaciones que permitan simplificar un problema y así poder realizar conceptualizaciones sobre los mismos.

Abstraer — capacidad de — Decidir qué detalles de algo hay que resaltar y cuáles ignorar

ejemplo

¿Cómo preparar la merienda?

Existe otra forma de representar estas oraciones para preparar una infusión, para ello podemos abstraer el nombre de la bebida como una "X".
Preparar té: Hervir agua y colocar saquito de té.
Preparar café: Hervir agua y colocar saquito de café.
Preparar mate cocido: Hervir agua y colocar saquito de mate cocido.

Preparar X: Hervir agua y colocar saquito de X.

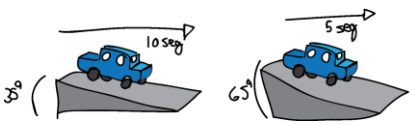
Evaluar: Se refiere a la capacidad de poder analizar críticamente las soluciones creadas para detectar y corregir errores, como así también verificar si una solución es eficiente en término de uso de recursos.

Evaluar — significa poder — Poner a prueba las soluciones para descubrir si son eficaces y eficientes

ejemplo

¿La pendiende diseñada es optima?

Una vez que se da la solución a un problema se debe evaluar si es útil para los casos para los que la va a utilizar teniendo en cuenta las posibles variantes.
¿La pendiente dada es suficiente para el tiempo que se quiere lograr? Que pasa si el auto va más o menos cargado?



En pocas palabras, el pensamiento computacional se concibe entonces como una metodología para la resolución de problemas que, si bien está vinculada a las tecnologías digitales y a la automatización, implica el desarrollo de capacidades, estrategias y formas de análisis que es posible aplicar para otras áreas y problemas.

Para conocer más.

Se recomienda la lectura del documento "Pensamiento Computacional. Guía para profesores" para enriquecer este pequeño resumen.

Disponible en: http://www.codemas.org/wp-content/uploads/2016/04/Pensamiento-computacional-Gu%C3%ADa-para-profesores.pdf

3. Recursos educativos para fomentar el desarrollo del pensamiento computacional en los jóvenes.

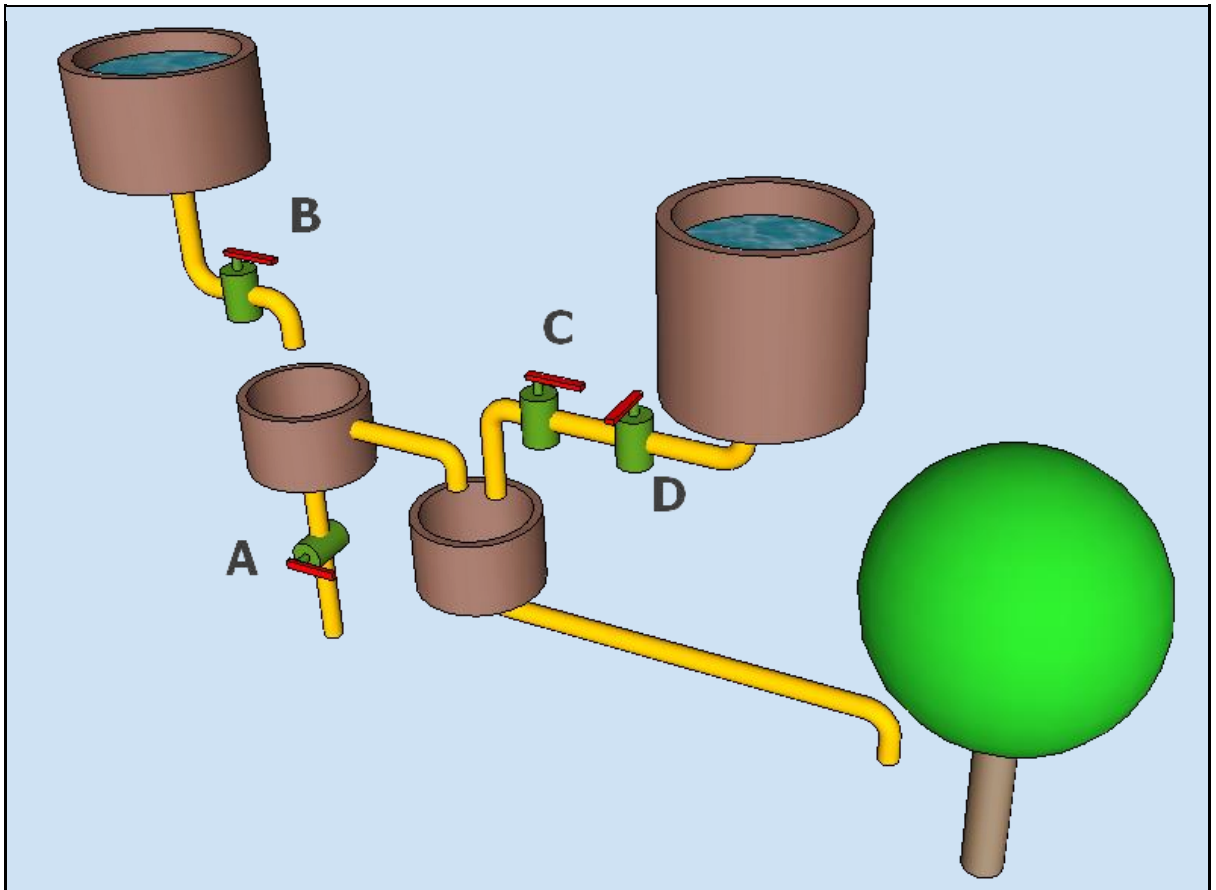
Si bien el desarrollo del pensamiento computacional es un tema relativamente nuevo, es posible hallar una gran variedad de opciones a la hora de elegir una estrategia didáctica para ello. Los enfoques más populares se basan en resolver **problemas de programación** utilizando plataformas en línea o de ejecución local. Estas actividades suelen formar parte de programas educativos de organizaciones sin fines de lucro que buscan promover la disciplina (Code.ORG internacionalmente y la Fundación Sadosky en Argentina). Los entornos de programación más utilizados suelen presentarse a través de herramientas visuales que utilizan bloques encastrables para crear programas usando una interfaz de "arrastrar y soltar". La ventaja de utilizar este tipo de actividades es que existe una gran oferta internacional y nacional probadas y acompañadas de guías

didácticas, que permiten abordar con cierta profundidad una gran variedad de los problemas del pensamiento computacional.

Ejemplo de actividad "Hora de Código" de code.org. *A través de elementos propios del popular juego "Minecraft" se plantea un problema a los jóvenes que debe ser resuelto utilizando la interfaz de programación de bloques gráficos. Se puede ver en sector etiquetado como "bloques" donde se encuentran las instrucciones disponibles y un sector llamado "Espacio de trabajo" donde se coloca la solución que puede ser puesta a prueba con el botón ejecutar.*

Otro tipo de estrategia posible son las denominadas **actividades desconectadas**, las cuales consisten en una gran variedad de ejercicios, juegos y experiencias que se desarrollan sin el uso de computadoras. En general

este tipo de estrategias didácticas abordan de manera separada los temas y los problemas del pensamiento computacional. Así, por ejemplo, existen actividades que ejercitan la capacidad de abstracción, otras que se basan el reconocimiento de patrones y otras que pueden enfocarse en la explicación de cómo funciona un algoritmo en particular.



Ejemplo de actividad desconectada del test Bebras https://www.bebras.org *En esta actividad los jóvenes deben identificar en qué casos el árbol recibiría agua de los tanques.*

Para resolverlos deben especificar los estados de cada una de las válvulas "A,B,C,D" y si el árbol recibe o no agua, y así confeccionar la tabla siguiente:

A	*B*	*C*	*D*	*Llega agua?*
Abierto	*Abierto*	*Abierto*	*Abierto*	*Si.*
Abierto	*Abierto*	*Abierto*	*Cerrado*	*No.*
Abierto	*Abierto*	*Cerrado*	*Cerrado*	*No.*
Etc….				

Estas estrategias didácticas no son mutuamente excluyentes, sino que, por el contrario, resultan más efectivas al usarse en complemento una de la otra. Una forma usual de utilizar ambos enfoques, es utilizar las actividades desconectadas como una introducción al tema a trabajar, para luego avanzar en prácticas que involucren el uso de tecnologías digitales. Este punto será desarrollado en el libro que se encuentra en producción por la UNIPE.

Para conocer más.

Tal como han sido mencionado Code.org, propone una serie de ejercicios que tiene gran popularidad

Disponible en:

https://hourofcode.com/es/learn

Así mismo, la fundación Sadosky propone ejercicios en la misma línea junto con manuales y actividades de acompañamiento.

Disponible en:

http://program.ar/manual-para-docentes/

4. Actividades desconectadas

Las experiencias *unplugged o* desconectadas, como han sido presentadas, comprenden actividades que se realizan con ejercicios y problemas que no requieran de una computadora para su resolución. El concepto de actividades *unplugged* para el desarrollo del pensamiento computacional surgió originalmente a partir de un programa de educación que buscaba involucrar a estudiantes de la escuela primaria con contenidos propios de la disciplina de la computación y la programación. Éstas actividades, que se encuentran disponibles en el sitio web csunplugged.org, se han complejizado desde su creación y abarcan un gran abanico de actividades. Conforme el pensamiento computacional se ha introducido en la currícula obligatoria educativa en distintos países del mundo, (especialmente en la Argentina con los nuevos diseños curriculares que abarcan desde los niveles iniciales hasta los de las escuelas técnicas) han dejado de ser actividades que solamente se aplicaban en cursos extracurriculares de corta duración para ser elementos que pueden ser utilizados dentro del desarrollo habitual de los contenidos escolares. Por este motivo, ahora también es usual encontrarlos acompañados de material para los docentes. En este sentido puede resultar interesante analizar el material que proponen tanto la fundación Sadosky, CS Unplugged, como "Pensamiento computacional Brasil".

Estas actividades desconectadas comparten características independientemente de a qué iniciativa o programa les diera su origen:

- ✓ No utilizan computadoras
- ✓ Dan un sentido de juego o desafío al estudiante para que esté explore
- ✓ Suelen incorporar elementos de trabajo manual o corporal
- ✓ Tienen un enfoque constructivista.
- ✓ Suelen ser cortas y simples de explicar
- ✓ A veces, están acompañadas de elementos de fantasía que le dan cohesión y ayudan a desarrollar metáforas.

La naturaleza de estas actividades les confiere una serie de ventajas por sobre los acercamientos tradicionales a la computación que se basan en software y hardware:

- ✓ **El protagonismo está en el problema a trabajar y no en la tecnología para necesaria resolverlo:** al no tener que enfrentarse con problemas propios de una implementación de software, el problema que se plantea a resolver es el núcleo de la actividad y permite presentar a las ciencias de la computación como algo más que solo saber programar.

- ✓ **Menor nivel de abstracción:** gracias a que estos elementos se pueden ver y tocar y que, en general, se representan a través de metáforas se logra quitar niveles de abstracción a los problemas de computación más esenciales. Usualmente los docentes de tecnología y programación suelen valerse de estas herramientas para explicar los conceptos, pero no son utilizados luego para la experimentación de los estudiantes, para permitirles a ellos madurar los conceptos.

- ✓ **Menos tiempo:** se pueden desarrollar actividades que se desarrollan en periodos de tiempos cortos, haciendo que sea más fácil de incorporar en ambientes educativos y dejando espacio para la reflexión sobre lo trabajado. Cuando se usa un enfoque educativo que incorpore algún software de programación, la principal

barrera que se encuentra es que primero se debe dedicar mucho tiempo a entender las interfaces y las instrucciones del software, en consecuencia, los problemas de programación que se trabajan quedan opacados por esta dificultad.

- ✓ **No requieren de conocimientos técnicos:** no se necesita de personal capacitado en tecnologías digitales para mantener o utilizar las actividades.

- ✓ **Bajo costo:** En general las actividades desconectadas requieren de una inversión muy baja de dinero e incluso la mayoría de ellas se pueden realizar utilizando solo lápiz y papel.

- ✓ **No requieren de infraestructura tecnológica:** no se necesita Internet ni computadoras y por lo tanto no tienen costos de mantenimiento ni de actualización.

Estas características hacen que las actividades desconectadas sean particularmente interesantes para desarrollar en ciertos escenarios:

- ✓ *Como primera actividad para introducir un tema de ciencias de la computación,* ya que permite disminuir el nivel de abstracción y construir metáforas para luego analizar y entender mejor los conceptos que subyacen a las actividades.
- ✓ Como actividad motivadora que despierte interés en los jóvenes sobre la disciplina, sin necesidad de abordar problemas complejos ni dedicarle extensas jornadas de trabajo.
- ✓ Cuando se deseen incorporar prácticas de pensamiento computacional a un curso y no se disponga del tiempo o los recursos económicos necesarios para implementarlo a través del uso de hardware y software.
- ✓ *Cuando se quiera desarrollar una estrategia educativa a gran escala* que pretenda introducir los temas de ciencias de la computación a través de la currícula en distintos niveles educativos en formaciones que no sean de carácter específico.

Es en este último punto es por el cual las actividades desconectadas han crecido en popularidad y oferta debido al poco tiempo y recursos que requieren para su implementación. Como se ha mencionado anteriormente csunplugged.org ha surgido específicamente con ese objetivo y luego se ha trasladado a otros programas educativos, como por ejemplos las iniciativas de la fundación Sadosky donde se las puede ver entremezcladas con otras actividades más tradicionales que utilizan la plataforma Scratch de programación.

Es posible realizar una taxonomía dentro de las actividades disponibles ya que poseen características que pueden hacer que resulten más o menos efectivas dependiendo del ámbito en que se utilicen. Es por ello que se necesita diferenciarlas para mejorar su análisis y su implementación.

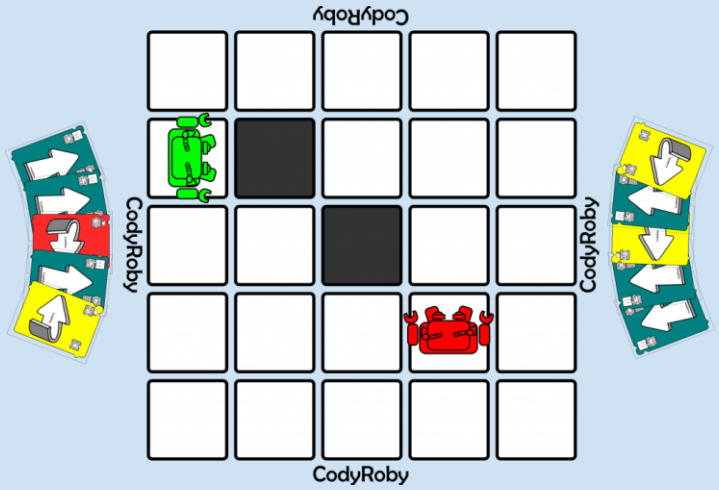
Actividades kinestésicas: *Los estudiantes desarrollan actividades donde deben seguir instrucciones para recorrer un laberinto. En esta caso ellos toman el papel de autómatas que interpretan programas para desplazarse (Ejemplo tomada de la web csunplugged.com)*

Actividades kinestésicas: Son aquellas actividades que involucran movimientos físicos de parte de los estudiantes. Suelen ser apropiados para el nivel inicial y los primeros años de primaria, sin embargo, para los últimos años o para niveles superiores no resulta apropiado ya que no es atractivo para los estudiantes y puede llegar incluso a desmotivarlos. Ejemplos de estas actividades es construir con tarjetas u otro recurso

un conjunto de instrucciones “aplaudir” “saltar” “girar” y diseñar “bailes” que los estudiantes deban ejecutar conforme a cómo se utilizan las tarjetas. Así también otra opción común que un estudiante o docente se desplace físicamente en un laberinto y que deba ejecutar las instrucciones recibidas.

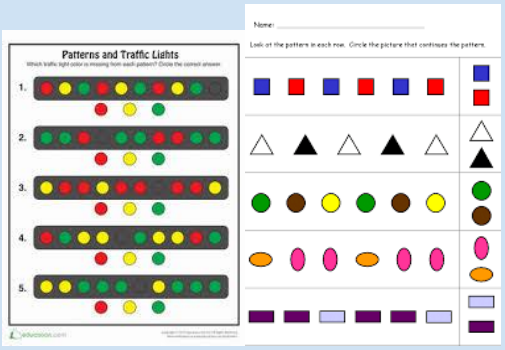
Actividades con recursos didácticos tangibles. Los tangram son ejemplos de actividades en los que se requieren recursos físicos para desarrollarse. A la derecha, se puede ver un laberinto diseñado para ser recorrido por una bolita. Este puede ser utilizado para diseñar “circuitos lógicos” sencillos donde intervienen compuertas que permitan o no el paso de la bolilla.

Actividades con recursos didácticos tangibles: Son actividades que involucran recursos físicos para lograr presentar el problema y sus soluciones. Ejemplo de este tipo de actividad son los Tangram y otro tipo de rompecabezas.



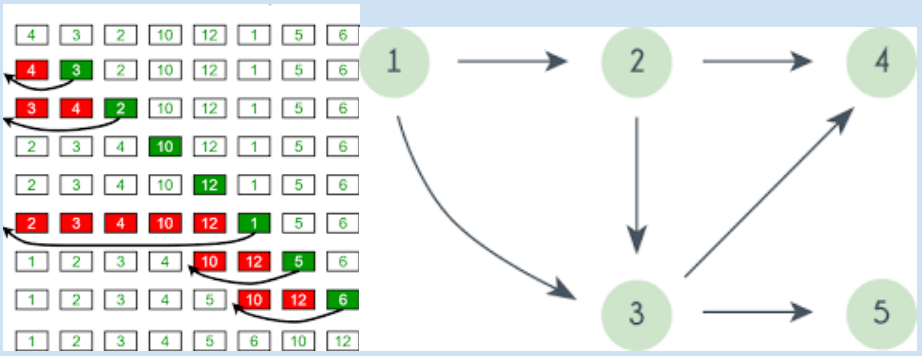
Juegos de mesa. *Cody & Roby es un juego de mesa gratuito y abierto disponible para descargar desde internet (http://codeweek.it/cody-roby-en/diy-starter-kit/). En él se ofrecen tarjetas que deben utilizar los jugadores para crear programas que ejecutará el robot de cada participante.*

Juegos de mesa: Tal como indica el nombre son juegos que se valen de diferentes elementos que planteen problemas de ciencias de la computación en donde los estudiantes participan en calidad de jugadores. Dentro del ambiente del juego, desarrollar los conceptos que se quieren abordar pedagógicamente debería implicar que los jugadores mejorarán en el resultado de sus partidas.



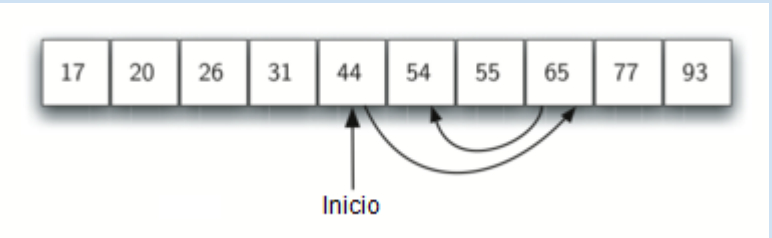
Actividades de búsqueda e identificación de patrones. Estas actividades suelen apoyarse en recursos visuales, y suelen pedir que los jóvenes descubran el patrón por el cual los objetos presentados están organizados.

Actividades de búsqueda e identificación de patrones: En general estas actividades buscan despertar las habilidades relacionadas con el descubrimiento de patrones y la capacidad de generar abstracciones. Suelen ser sencillas y basarse tanto en elementos gráficos como en las características de una serie de elementos a los que hay que aplicarles alguna categoría o etiqueta. Pueden comprender actividades para descubrir las reglas que subyacen a la división de dos o más conjuntos de elementos, por ejemplo podría ser buscar que diferencia al conjunto “{manzana, cerezas, frutilla}” de “{limón, banana, melón}” (en este caso el color).



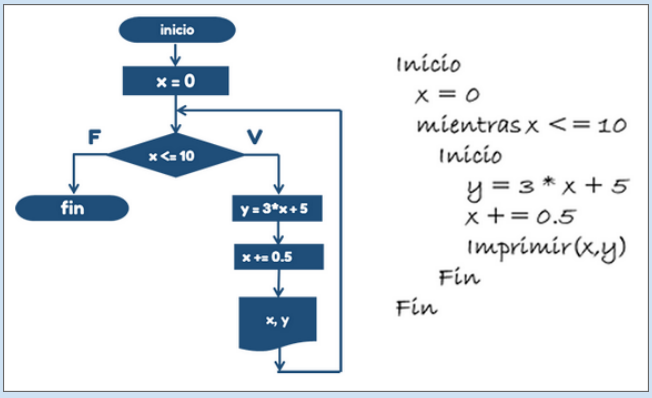
Actividades representación y ejecución de algoritmos.*Muchas de estas actividades se basan en algoritmos de búsqueda o de ordenamiento. En el primer caso se ve la ejecución de un algoritmo de ordenamiento donde se analiza cada elemento y se lo coloca en el lugar que corresponde según orden ascendente.*

Actividades representación y ejecución de algoritmos: En este caso son actividades donde a partir de la explicación del funcionamiento de un algoritmo los estudiantes deben ejecutarlo sobre un conjunto de datos. Ejemplos de este tipo de actividad pueden ser construir imágenes en grillas a partir de reglas, o aplicar algoritmos de ordenamiento a un conjunto de cartas numeradas.



Actividades de descubrimiento de algoritmos: *en este caso se pretende que el estudiante descubra un algoritmo de búsqueda binaria, al analizar una lista de elementos ordenada.*

Actividades de descubrimiento de algoritmos: Son actividades que buscan que los estudiantes descubran cuál es el algoritmo detrás de alguna actividad que permita llegar a una solución. Un ejemplo de este tipo puede ser ofrecerles a los estudiantes un conjunto de elementos y que ellos deban generar las reglas que permitan realizar una clasificación.



Actividades de creación de algoritmos. *Los diagrama de flujo o el pseudocódigo son alguna herramienta útiles que permiten crear "programas" sin utilizar computadoras*

Actividades creación de algoritmos: Son actividades similares a las anteriores, pero corresponden a problemas que tienen soluciones más amplias y donde la creatividad pueda tener un papel más importante. Este tipo de actividades son cercanas a la programación sin el uso de computadoras, por lo que requiere de una persona que ayude a interpretar las instrucciones para buscar errores. Son actividades ideales para realizar en grupos y funcionan en diferentes niveles educativo dependiendo de la complejidad que se quiera llegar. Usualmente utilizar herramientas para generar un prototipo de un algoritmo puede ser un primer paso antes de pasar a traducirlo a un lenguaje de programación tradicional.

5. Consideraciones finales

La enseñanza del pensamiento computacional es una actividad relativamente nueva para niveles de secundario, primaria e inicial. Por tal motivo estas constituyen una didáctica en construcción. Esta característica crea la necesidad de analizar de forma crìtica el material y las estrategias didácticas que se quieran utilizar. No obstante, esto no significa que el docente deba aferrarse a un enfoque tradicional basado en la memorización y en los ejercicios de respuesta única, por el contrario, el docente debe desarrollar capacidad de análisis y evaluación para trabajar con enfoques y herramientas didácticas innovadoras sin perder su mirada crítica.

En su artículo “CS Unplugged-- “How is it Used, and Does It Work?” Tim Bell y Jan Vahrenhold realizan una caracterización de las didácticas desconectadas utilizadas para promover el pensamiento computacional y reúnen allí una serie de críticas provenientes de pares académicos. Es interesante sintetizar algunas conclusiones de este trabajo:

- ✓ Las actividades desconectadas pueden resultar poco motivadoras para cursos de estudiantes en los últimos años de secundaria, especialmente para aquellas actividades que tienen componentes kinestésicos. Esto significa que se deben aplicar con criterio el tipo de actividad que se desea usar teniendo en cuenta no solo el nivel de conocimientos previos, sino también la edad de los estudiantes y los intereses y gustos de los mismos.
- ✓ La aplicación de actividades de PC desconectadas puede ser poco significativas o incluso tener un impacto negativo si no son acompañadas de un marco pedagógico que las vinculen con actividades que incorporen efectivamente tecnologías digitales.

Las actividades desconectadas son difíciles de comparar a través de los diferentes trabajos académicos debido a que existe una gran variedad tanto en tipos como en implementaciones posibles. Esto, sin

embargo, no invalida las críticas recogidas, sino que pone en evidencia la necesidad de una sistematización tanto en la clasificación como en el análisis de la efectividad de cada una de las actividades para los diferentes niveles educativos y estrategias didácticas complementarias. Un plan de trabajo completo para desarrollar el pensamiento computacional en los jóvenes por lo tanto, no debe limitarse solamente a las actividades desconectadas, sino que éstas deben estar acompañadas de un marco pedagógico más amplio que busque vincularlas con las prácticas tradicionales que utilizan tecnologías digitales.

Para conocer más.

Se detallan a continuación las actividades y sitios mencionados que contienen actividades e información sobre actividades desconectadas para el desarrollo del PC.

La propuesta original, con actividades en español https://csunplugged.org/es/

Propuesta de PC en Brasil con actividades en español http://www.computacional.com.br/index.html#atividades

6. Referencias

CAS (2015) Pensamiento Computacional. Guía para profesores. Computing At School.

Wing, J. (2011). Research notebook: Computational thinking—What and why? The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. Disponible en http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

CS Unplugged—How Is It Used, and Does It Work? Tim Bell and Jan Vahrenhold 2018

Disponible en https://link.springer.com/content/pdf/10.1007%2F978-3-319-98355-4_29.pdf

Fundación Sadosky (2013) CC – 2016. Una propuesta para refundar la enseñanza de la computación en las escuelas argentinas. http://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf

Actividades desconectadas para el desarrollo del pensamiento computacional.

unipe Universidad Pedagógica Nacional PÚBLICA Y GRATUITA

En complemento a las actividades sugeridas en los sitios mencionados, hemos desarrollado una serie de ejercicios y juegos para realizar, con muy pocos recursos, experiencias desconectadas. Consideramos que las capacidades que componen el pensamiento computacional están en gran parte coordinadas por el pensamiento algorítmico, ya que es una característica que las engloba y coacciona. Por tal motivo, es necesario que la ejercitación de tales capacidades se realice de forma conjunta y no como elementos separados que deban ser tratados individualmente. Esta motivación es parte de los criterios de diseño de las actividades, por lo que los ejercicios y los problemas que se proponen están enfocados en el pensamiento algorítmico como elemento central de coordinación.

Debido a que es imposible escribir programas que se ejecuten por sí solos (si no se cuenta con un procesador) serán los mismos estudiantes y los docentes quienes actuarán interpretando dichas instrucciones en las actividades planteadas. Esto confiere algunas desventajas en la comodidad pero algunas ventajas en lo que concierne al aprendizaje y al trabajo en equipo. Tener que ejecutar el código con la ayuda de un par, facilita la interacción entre los estudiantes y promueve un aprendizaje en colaboración al mismo tiempo que requiere un mayor análisis de los programas creados.

Las actividades propuestas están pensadas como una alternativa para poder trabajar con estos temas en un aula en la que no se cuente con los elementos informáticos necesarios para poder desarrollar trabajos de

programación. Sin embargo, no es éste el único entorno en el que creemos que puede resultar provechoso aplicarlas, ya que pueden llegar a ser útiles de igual forma como un primer paso en el mundo de la programación más allá de las restricciones de recursos.

La mayor parte de estas actividades se desarrollan, al inicio, de una forma más guiada que requerirá de una mayor intervención docente, proporcionando algunos conceptos de andamiaje para ir avanzando luego en actividades que le den mayor libertad a los estudiantes conforme se complejizan. Se intentará por lo tanto brindar oportunidades donde haya más de una solución al mismo problema y que los jóvenes puedan expresarse creativamente resolviendo los mismos.

Las actividades están cuidadosamente diseñadas para hacer que los estudiantes experimenten con distintos aspectos de la programación y de la resolución de problemas, en particular:

- ✓ Análisis de datos y de problemas.
- ✓ Búsqueda de patrones en los datos y soluciones.
- ✓ Abstracción de las soluciones creadas para aplicarlas en más de un problema.
- ✓ Construcción de soluciones algorítmicas.
- ✓ Estructuras básicas de programación: programa, instrucción, estructura repetitiva, estructura condicional.

Cada una de las actividades se presentan primero en su intención pedagógica, definiendo cada uno de los elementos que lo componen, así como su porqué, y finalmente se pone a disposición en formato de anexo las actividades preparadas para ser entregadas a los estudiantes.

Las actividades que componen la propuesta, en orden de complejidad son:

a) Hamburguesería Don Simón:

Actividad dirigida al segundo ciclo de la escuela primaria, donde se presentan una serie de problemas en donde se plantea la automatización de la confección de los sandwiches a través de una máquina hipotética que recibe ciertas instrucciones. La mayoría de los ejercicios contempla más de una solución y permite que haya margen para optimizar y comparar soluciones. Las actividades se van complejizando y plantea en última instancia el diseño de una máquina similar a la utilizada que prepare bebidas. Esta propuesta hace foco en el uso de instrucciones simple y en el concepto de programa y de sintaxis.

b) Carrera de ratones:

Actividad orientada, de manera central, a la escuela primaria aunque también se puede adaptarse a otros niveles. Se basa en un juego de mesa sencillo que hace hincapié en el uso de estructuras repetitivas. Se plantea para dos o más jugadores por tablero, donde cada jugador debe crear un programa (con las cartas que le toca que contienen las instrucciones) que le permita a su ratón comer la mayor cantidad de quesos. Gana el jugador que mayor cantidad de quesos haya reunido. Al ser un juego donde las instrucciones se reparten al azar, se favorece el uso de soluciones creativas para aprovechar al máximo las cartas. Esta actividad avanza sobre el uso de estructuras repetitivas simples para optimizar el uso de las cartas que obtiene cada jugador.

c) Desafíos del ratón:

Recurso orientado a la escuela primaria y primeros años de secundaria. Se proponen escenarios que, aprovechando el modelo juego de mesa, incorpora nuevos elementos de programación. En particular en las estructuras repetitivas con condición y un trabajo un poco mayor con los condicionales. Se puede pensar como una actividad cercana a la tortuga de logo, propuesta en 1967 por Seymour Papert.

Las actividades, si bien tienen un principio y un final, no se consideran como un paquete cerrado ni mucho menos como un kit. Se espera que al hacer público los criterios expuestos y la justificación pedagógica de cada decisión en las actividades creadas, le de las herramientas necesarias al docente para expandir estas actividades y adaptarlas a sus necesidades educativas.

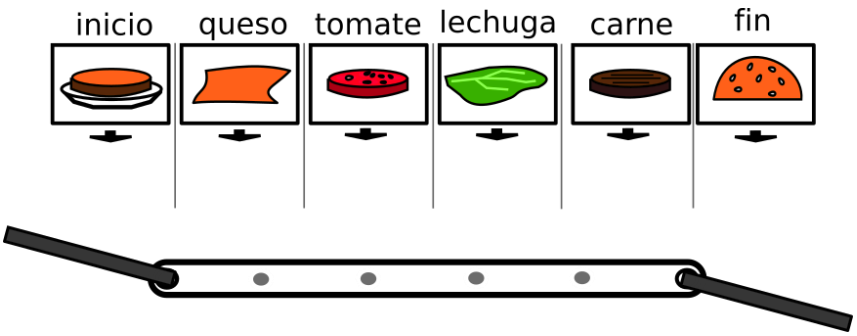
Si bien se presentan en un orden, estas tres actividades pueden utilizarse de forma separada. En particular "Carrera de ratones" es la más apropiada para los más jóvenes (aunque también funciona para los más grandes), mientras que la "Hamburguesería de Simón" y los "Desafíos del ratón" más apropiados para los últimos años de primaria.

En cuanto a la utilización en el aula, se sugiere acompañar el desarrollo de las actividades con momentos de reflexión que permitan sacar provecho de la actividad práctica. Muchas veces suele suceder que debido a la novedad de la propuesta estos momentos se pierden. Sin embargo, el verdadero aporte educativo no está en las actividades en sí mismas, sino en la capacidad del docente de relacionar estas actividades con los temas que desea trabajar.

Tal como ha sido mencionado anteriormente, resulta interesante continuar este trabajo con actividades que requieren de software y hardware. Una propuesta sencilla y atractiva para los estudiantes, pueden resultan los recursos que pone a disposición Code.org en la "hora de código". Estas actividades se pueden desarrollar en línea, con ejercicios de programación que utilizan interfaces visuales y que están acompañadas de elementos fantásticos de personajes de películas y juegos famosos. El docente podrá evaluar cuál de ellos puede resultar más provechoso y atractivo para los estudiantes.

Actividad: Hamburguesería de “Don Simon”

Esta actividad plantea una máquina sencilla que cuenta con una cinta transportadora, una serie de dispensadores de alimentos y un intérprete de instrucciones. Se puede ver la máquina hipotética a continuación.



Esta actividad está centrada en trabajar el **concepto de instrucción** y de **programa** y plantea además conceptos que tienen que ver con optimizar las soluciones planteadas.

Cada instrucción en los programas de esta máquina requiere un tiempo de ejecución, y por tal motivo ordenar bien la secuencia de instrucciones termina por crear programas más o menos complejos. Por otro lado la actividad se plantea dividida en distintas etapas, donde se van agregando instrucciones y desafíos.

En el día 1 se presenta el problema principal, que es construir recetas de hamburguesas y traducir estas recetas a órdenes que pueda entender la máquina. Las instrucciones principales son: inicio, ingrediente X (ejemplo lechuga), flecha →, flecha ← y fin. En estas recetas no existe espacio para la optimización, ya que en general son programas muy sencillos. Sin embargo si se va planteando la idea de que cada instrucción tiene un costo de tiempo y que hay que tenerlo en cuenta al momento de crear los programas.

En el día 2 se incorporan nuevas instrucciones avanza X, retrocede X y timbre. Las primeras dos permite escribir programas más cortos, a la vez que introduce un costo en tiempo diferente, que hace que a veces sea oportuno utilizarlas y otras veces no. Esto da lugar a optimizaciones. La última instrucción nueva, timbre, servirá en las próximas actividades para ayudar a seguir el programa.

En el día 3 se incorpora un nuevo elemento, el botón. De esta manera se introduce una pequeña noción de condicional, al marcar ciertas secciones del programa que puede o no ejecutarse. Probablemente requerirá ayuda del docente en esta primera instancia para entenderlo mejor. Para simplificar, se eligieron 3 combos de hamburguesas a confeccionar, y se provee un esqueleto de programa para que completen los estudiantes.

En el día 4 se incorpora un botón nuevo, pero no se provee un esqueleto de programa. Se espera que gracias a lo aprendido con la actividad anterior, ellos sepan incorporar al código en el lugar indicado, las nuevas instrucciones.

En el día 5 se plantea un problema de optimización: acomodar el orden de los ingredientes en la máquina para reducir los tiempos de confección. Por tal motivo los estudiantes deberán analizar los combos, el programa construido y la máquina misma. Esta actividad sirve para fomentar un pensamiento lateral, en el sentido de que si no podemos optimizar el programa tal vez podemos optimizar la forma en que la máquina los ataca, además plantea la idea a los estudiantes de que las máquinas se pueden modificar, algo que puede ser útil si en un futuro se continúa trabajando en la línea de la computación física (trabajando por ejemplo con plataformas como Arduino).

En el día 6 se propone en otro nivel crear directamente una máquina, similar a la utilizada hasta el momento, pero que sirva para preparar las bebidas. Se detallan las recetas, los botones y las instrucciones que tendrá. Los estudiantes deberán confeccionar el programa y el orden de los ingredientes en la cinta para dar solución al problema.

Se espera que a partir de la resolución de estas actividades los estudiantes hayan trabajado con su capacidad de abstracción, entiendan que significa una instrucción, qué es un programa y cómo una máquina funciona a partir de interpretar las instrucciones que una persona introduce en un programa. Además se da pie a trabajar sobre el concepto de procesador y del conjunto de operaciones del mismo.

Más allá de los "días" de los desafíos propuestos, esta actividad se puede expandir creando nuevas máquinas imaginarias que sirvan para resolver otros problemas. La imaginación es el límite, ya que estas máquinas no cuestan dinero.

Actividad: Carrera de ratones.

Se presenta en esta ocasión un juego de mesa donde los estudiantes deberán crear programas para ganar puntos. Si bien se puede jugar de a uno o más jugadores, lo ideal es jugarlo una pareja por tablero. Se anexan al final las reglas y las fichas y cartas necesarias para poder jugar.

En este juego cada estudiante cuenta con un ratón propio que recibe órdenes a partir de programas. El objetivo del juego es capturar la mayor cantidad de quesos posible y este termina cuando ya no quedan más quesos en el tablero.

Se les plantea a los jugadores un objetivo claro y simple que se debe conseguir a partir de utilizar, con creatividad y criterio, las cartas de instrucciones que les son dadas por azar. Por tal motivo, interiorizar el significado de las instrucciones y cómo se combinan le da mayores posibilidades al jugador de ganar. El azar contenido en la distribución de las cartas garantiza que se presenten nuevas situaciones y se mantenga un interés en seguir jugando.

El juego además de seguir trabajando el concepto de programa e instrucción, avanza en la capacidad de utilizar estructuras repetitivas simples que requieren cierto nivel de abstracción en cuanto a los movimientos del ratón.

La dicotomía que se plantea entre ser el programador y el intérprete de las instrucciones, hace que los estudiantes se pongan en cada uno de los roles logrando así un entendimiento más profundo de cómo las instrucciones y la disposición de las mismas afectan al programa como un todo.

El cruce que se produce entre los oponentes del juego habilita además un trabajo colaborativo en la interpretación y ejecución de los programas así como también funciona como un espacio de diálogo y esparcimiento.

Actividad: Desafíos del ratón.

En esta actividad se plantean una serie de ejercicios que tiene que ver con avanzar en complejidad con los temas trabajados en el juego de mesa, atendiendo a problemas de programación y de eficiencia. La diferencia principal de esta actividad con la anteriores, es que están diseñados de manera tal que puedan ser trabajados tanto de manera individual como grupal y que cada nivel está construido para desafiar al estudiante en un aspecto particular de la programación, enfrentándolo a situaciones con las que puede no haberse encontrado antes en el juego de mesa.

Se plantea como un videojuego, en el sentido de que conserva el formato de "niveles" que se deben resolver utilizando cierta cantidad de tarjetas para construir programas (siguiendo la línea del juego de mesa). Reutiliza en tal sentido las fichas y las cartas del juego de mesa y agrega algunas más.

Cada nivel está compuesto por un escenario y una serie de cartas que se pueden utilizar. Opcionalmente en algunos niveles también proponen desafíos que buscan que el estudiante quiera conseguir una mayor eficiencia en su programa (utilizando menos instrucciones por ejemplo) o encarar el problema de una forma diferente. Es importante respetar las cartas sugeridas para lograr que se le presenten los problemas que fueron pensados para cada nivel.

El objetivo del programa para todas las actividades es que el ratón coma todos los quesos que existen en el escenario sin que este se salga del camino o caiga en una trampa.

A lo largo de los niveles se introducen nuevas cartas (que no se encuentran presente en el juego de mesa) y que sirven para profundizar los conceptos adquiridos y trabajados, haciendo especial hincapié a los ciclos y a las estructuras condicionales.

El primer nivel tiene por objetivo establecer los elementos básicos de este tipo de actividades para entender cómo es que funcionan. Se les debe explicar a los estudiantes las reglas del mismo y determinar el espacio de trabajo y la forma en que van a resolver, probar y compartir las soluciones. Nuevamente, es importante explicar con qué cartas cuentan para resolver cada nivel.

Se sugiere el trabajo por grupos de no más de dos o tres estudiantes y establecer pautas para que, en determinados momentos de la clase, se compartan las soluciones creadas, para probarlas y evaluarlas junto con los estudiantes.

Los jóvenes deben examinar el escenario y diseñar el programa que resuelva el nivel utilizando la menor cantidad de tarjetas posibles, de esta manera se busca que los estudiantes traten de optimizar las soluciones. Sin embargo no se deben tomar como inválidas a aquellas soluciones que no sean óptimas.

El momento de compartir las soluciones creadas debe aprovecharse para realizar una reflexión grupal sobre lo trabajado, destacando las soluciones ingeniosas y la forma en que éstas funcionan.

A continuación se presentan cada uno de los niveles desde un punto de vista docente y luego en el anexo se presentan en forma de ficha imprimible cada uno de ellos para que estén listos para ser utilizados por los estudiantes.

Objetivos de los niveles.

Nivel 1.

El objetivo del primer nivel es presentar el formato de las actividades para resolver un ejercicio sencillo. Aunque simple, el problema admite varias soluciones y este hecho sirve de ejemplo para que los estudiantes sepan que los ejercicios planteados no necesariamente tienen una sola solución.

Para el desafío las soluciones posibles son:

Solución 1	**Solución 2**	**Solución 3**

Avanzar
Avanzar
Avanzar
Comer
	Repetir x 3
	  Avanzar
	Comer
		Repetir x 2
		  Avanzar
		Avanzar
		Comer

Nivel 2.

El segundo nivel es muy similar al primero, la solución más probable es la de usar la repetición por 3, de avanzar y comer. Esta actividad sirve para experimentar cómo afecta la cantidad de repeticiones a la ejecución del programa para el nivel. Hay que recordarles a los estudiantes que “repetir” solo se termina cuando haya concluido la cantidad de repeticiones asignadas en la carta. Es decir que si repiten por 4 la acción de avanzar, la solución es incorrecta porque se sale del nivel.

Nivel 3.

En este nivel se introducen los giros. El objetivo del nivel es que los estudiantes descubran que hay un patrón que se repite y cómo este patrón se puede resolver usando una estructura repetitiva. Además en el desafío se propone la utilización de un ciclo anidado, es posible que muchos lleguen a la solución, sin embargo, es

importante destacar cómo funciona este tipo de estructuras ya que solo la instrucción “Avanzar” es la que se va a repetir dos veces por cada ejecución del “repetir x3” principal.

Las soluciones en este nivel son:

Solución trivial	**Solución probable**	**Solución del desafío**
Avanzar Avanzar Comer Girar al cuadrado Avanzar Avanzar Comer Girar al cuadrado Avanzar Avanzar Comer Girar al cuadrado	Repetir x3 Avanzar Avanzar Comer Girar al cuadrado	Repetir x3 Repetir x2 Avanzar Comer Girar al cuadrado.

Nivel 4

El cuarto nivel presenta una estructura similar al nivel anterior y constituye una forma de afianzar lo aprendido. No alcanzan las cartas de avanzar para crear una solución sin utilizar una estructura repetitiva. El desafío, por otro lado, busca que el estudiante anide dos estructuras de repetir por dos, para comprobar que es equivalente a repetir por cuatro.

Solución 1	Solución 2	Solución del desafio
repetir x4 Avanzar Comer Girar al cuadrado Girar al cuadrado Avanzar Girar al cuadrado	repetir x4 Avanzar Comer Girar al círculo Girar al círculo Avanzar Girar al círculo	repetir x2 repetir x2 Avanzar Comer Girar al círculo Girar al círculo Avanzar Girar al círculo

Nivel 5

Este nivel tiene también un patrón, ligeramente más complejo, que se debe resolver con una estructura repetitiva. Se plantean también dos desafíos: el primero para optimizar usando una estructura repetitiva y el segundo se busca un desafió más cercano al pensamiento creativo buscando resolver el problema haciendo que el ratón gire siempre para un mismo lado (combinando la estructura repetitiva con un giro).

Nivel 6

En esta ocasión se plantean dos patrones a resolver combinando múltiples estructuras repetitivas, requiere mayor experimentación que los niveles anteriores ya que se usan estructuras iguales para resolver problemas ligeramente diferentes.

Nivel 7

Este nivel es similar al anterior, sirve para afianzar lo trabajado antes de seguir avanzando en prácticas más complejas.

Nivel 8.

Se introduce un concepto nuevo que es la repetición bajo condición. En este caso la condición es que se debe repetir las instrucciones siempre y cuando haya quesos. Esta instrucción servirá luego par introducir los condicionales.

Nivel 9.

Se introduce un obstáculo nuevo. Que si bien no modifica lo trabajado hasta el momento, al igual que la instrucción nueva tomará sentido al introducir finalmente el condicional.

Nivel 10.

Se busca que el estudiante separe del ciclo algunas instrucciones antes de usar las repeticiones, para aprender a manipular el problema en pos de adaptarlo a cosas ya conocidas. Es un código más complejo que el utilizado hasta el momento. En este caso una solución posible sería:

Solución
avanzar girar al círculo

```
avanzar
comer
repetir x2
    girar al círculo
mientras haya quesos
    avanzar
    comer
```

Nivel 11.

Se plantea un nivel sencillo, que dará pie al nivel siguiente donde se vuelve a introducir el obstáculo de la trampa.

Nivel 12.

Se introduce finalmente la carta condicional. Esta carta chequea si delante del ratón se encuentra o no una trampa y en consecuencia se ejecuta o no ciertas instrucciones.

Nivel 13.

Se propone un escenario donde se utilizar la carta “hay trampa” como una especie de sensor que permite descubrir además donde hay quesos.

Solución posible:
mientras haya quesos avanzar girar al círculo ¿hay trampa adelante? comer girar al cuadrado

Nivel 14

Se afianza lo aprendido en el nivel anterior utilizando las trampas como sensores un sensores.

Nivel 15

En esta ocasión se propone un escenario con múltiples soluciones pero con una respuesta muy sencilla, pero que requiere anidar 3 estructuras.

Solución posible
mientras haya queso avanzar ¿hay trampa adelante? comer repetir x 2

```
        girar al círculo
```

Nivel 16

En este escenario se predispone un patrón donde solo hay quesos en el lugar en el que hay trampas debajo.

Nivel 17

Se presenta un escenario con un patrón simple que requiere de una ejecución más compleja que en el escenario anterior, para afianzar lo aprendido y anidar ciclos. Es uno de los programas más complejos hasta el momento en lo que respecta a abstracción y anidación de estructuras repetitivas.

Solución posible

```
mientras haya queso
  avanzar
  girar al círculo
  ¿hay trampa adelante?
      repetir x2
          comer
          repetir x2
              girar al círculo
          repetir x3
              avanzar
```

girar al cuadrado

Nivel 18

Si bien el escenario se plantea como complejo, en realidad, la solución que conlleva es muy sencilla. Es más que nada mostrarle a los estudiantes que problemas complejos no necesariamente requieren soluciones complejas.

Nivel 19.

Se presenta un patrón en el que solo hay quesos para comer cuando hay por debajo trampas. Detectar esta situación requiere que el estudiante anide dos estructuras condicionales.

Nivel 20.

El último nivel del desafío es similar al anterior en el sentido de que requiere del descubrimiento de un patrón para actuar en consecuencia, la diferencia está en la complejidad del mismo.

Más niveles.

Se invita al docente a extender los niveles ya sea por su parte o junto a los estudiantes, diseñando nuevos desafíos y cartas de instrucciones.