

Pensamiento computacional



Para consensuar los contenidos mínimos fundamentales que se espera que los estudiantes obtengan durante su escolaridad, en septiembre de 2018 se aprobaron los Núcleos de Aprendizaje Prioritarios (NAP) de Educación Digital, Programación y Robótica. Es a partir de esta resolución que la educación digital, la programación y la robótica comenzarán a ser obligatorias en todos los establecimientos del país. Según lo determinado allí, las jurisdicciones llevarán adelante la implementación de los NAP y su inclusión en sus documentos curriculares, adoptando diferentes estrategias y considerando las particularidades de sus contextos, necesidades, realidades y políticas educativas en el lapso de dos años.

Algunas metas de aprendizaje que se proponen son:

- Iniciarse en la resolución de situaciones problemáticas transitando las diferentes etapas del proceso: identificar el problema, formular hipótesis, investigar y elaborar conclusiones.
- Iniciarse en el desarrollo del pensamiento computacional como estrategia para el planteo y la resolución de situaciones problemáticas.
- Intercambiar ideas, realizar diversos registros y analizarlos haciendo uso de diversas herramientas digitales.

Por ello, en *Malabares matemáticos 4*, incluimos la propuesta , con actividades que refuerzan los temas abordados en cada módulo. En ellas se utilizan recursos digitales que potencian el desarrollo del pensamiento computacional, principalmente a través de la programación.

Pero... ¿qué entendemos por pensamiento computacional?

El *pensamiento computacional* es un proceso que permite formular problemas de manera que sus soluciones puedan representarse como secuencias de instrucciones, llamadas algoritmos.

Este proceso de resolución de problemas comprende las siguientes características:

- Organizar y analizar lógicamente la información.
- Representar la información a través de abstracciones (por ejemplo, simulaciones).
- Automatizar estableciendo una serie de pasos ordenados para llegar a la solución, es decir, utilizando algoritmos.
- Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos.

Apunta a generar en los niños una forma de pensar que les permita aprender a plantearse problemas y sus soluciones, cumpliendo una secuencia determinada de pasos en el proceso. El pensamiento computacional ayuda a tomar decisiones de una manera ordenada, secuenciada, lógica y sin ambigüedades. Algo que a veces resulta difícil en el ámbito de las ciencias de corte más social.

Hay muchas formas de desarrollar el pensamiento computacional en la escuela. Aquí aportamos algunas maneras de incluirlo. Lo importante es que una vez que los alumnos logran fluidez en el uso de las herramientas, empiezan a aplicarlo por su cuenta y en un espacio más amplio del propuesto.

Si bien el pensamiento computacional está ligado al razonamiento que se logra programando frente a una computadora, no debe trabajarse necesariamente de esta manera; podemos abordarlo de manera *unplugged* (desconectada/sin PC). Es decir, mediante ejercicios y experiencias de resolución de problemas, realizando trabajos de conceptualización sobre los pasos llevados a cabo en la experiencia.

¿Qué relación hay o en qué medida se diferencian las varias formas de pensamiento computacional de aquellas correspondientes al pensamiento matemático?

Pensemos en un caso. Un alumno desea graficar datos de un experimento y encuentra un patrón común entre estos datos.

La matemática le permite expresar ese patrón mediante una ecuación o una fórmula. De esta manera va a poder predecir resultados posibles.

Cuando incluimos las nuevas tecnologías, los alumnos pueden usar una PC para dar un paso más allá de lo que a primera vista se puede indagar y así lograr hacer análisis con resultados basados en la evidencia.

Es ahí donde aparece el pensamiento computacional, cuando se usan métodos de simulación, redes, recolección automática de datos, razonamiento algorítmico y programación, entre otros.

¿Cómo trabajar con cada una de las propuestas ?

MÓDULO 1. Tramo 1

A tener en cuenta

En la actualidad, para comunicarnos, expresarnos y guardar nuestra información, usamos el sistema de numeración decimal y el alfabeto, según se trate de valores numéricos o de texto. Como las computadoras funcionan con electricidad, reconocen dos clases de mensajes: cuando hay corriente eléctrica, el mensaje es “sí”, y cuando no hay corriente, el mensaje es “no”. Por ello, para representar un valor dentro de una computadora, se usa el sistema de numeración binario, que utiliza solo dos dígitos: el cero (0) y el uno (1).

En el ejercicio se muestra la manera de representar gráficos sin colores, dado que es la forma más sencilla de entender el sistema binario. Sin embargo, se debe tener en cuenta que, tanto para representar caracteres como colores, es necesario trabajar con “grupos” de “0” y “1”.

En computación, al 0 y al 1 se los conoce como bit, que es la contracción de su nombre en inglés (*binary digit*). Cada conjunto de 8 dígitos binarios se denomina byte, que es el que se utiliza cuando es preciso representar un carácter, sea número o letra.

MÓDULO 1. Tramo 2

A tener en cuenta

El programa realizado con Scratch está pensado para que los alumnos practiquen cálculos mentales en sus casas, cambiando el rango de números a utilizar para incrementar la dificultad.

También puede utilizarse en el aula. Se sugiere crear un “campeonato de cálculos mentales”, en el que suma puntos el alumno que adivine primero. Para ello, se empleará una computadora ubicada en un lugar visible para todos, o (en caso de disponer) una computadora y un proyector.

¿Qué significa cada bloque utilizado?

Al presionar “Bandera Verde”: es el evento. Los eventos indican cuándo empieza una estructura de programación, en este caso, al presionar la Bandera Verde.

Fijar (Primer Número) a: es una variable llamada “Primer Número”. Las variables son “cajas contenedoras” que permiten guardar un dato por determinado tiempo. Para crear una variable se debe elegir la opción: “Variable - Crear Variable”.

Número al azar entre 1 y 10: guarda un número aleatorio entre el rango colocado.

Fijar (Segundo Número) a (número al azar entre 1 y 10): guarda el segundo valor en otra variable.

Esperar 2 segundos: es el tiempo que espera antes de dar la respuesta. Este intervalo debería modificarse en función de los números a sumar, para dar más tiempo a los alumnos para pensar la respuesta.

Fijar (Respuesta) a: es una tercera variable que guardará la suma de ambos números.

(Primer Número + Segundo Número): dentro de la categoría “Operadores” encontramos la opción de realizar diferentes cálculos matemáticos. En este caso, utilizamos “Suma” para sumar ambas variables.

Decir (unir la suma de los números es) (Respuesta) por 2 segundos: nos mostrará el resultado de la operación durante 2 segundos.



Para ver las capturas de Scratch en color, escaneá este código.

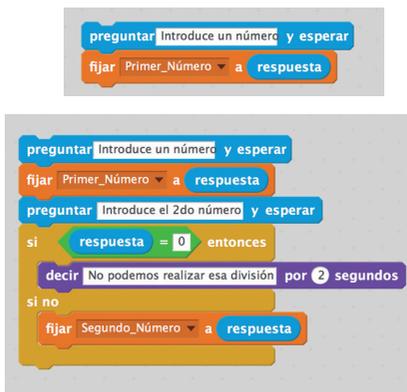
MÓDULO 2. Tramo 2

A tener en cuenta

Se propone realizar un programa que muestre si un determinado número es divisible por otro o no. En este caso no se utilizan variables; no obstante, sugerimos que una vez realizado el programa, como respuesta a la pregunta “¿Se te ocurre otra forma de hacerlo?” se trabaje con dos variables para reemplazar los números (8) y (3).

¿Qué se logra con el uso de variables? Que el programa sea interactivo. Para ello, hay que combinar la variable con el sensor “Pregunta - Respuesta”.

Es interesante que los alumnos analicen qué pasa si se coloca como segundo número el valor: 0. ¿Puede realizarse la división? ¿Cómo podría evitarse? Una vez analizado entre todos, se puede sugerir una manera de resolverlo (como la que se ve en la captura).



MÓDULO 2. Tramo 3

A tener en cuenta

A diferencia de las otras actividades, en este caso se propone analizar y decodificar un programa. ¿Qué hay que tener en cuenta? Primero es preciso entender cada uno de los bloques y, luego, su significado en conjunto.

Se sugiere que sea analizado entre todos, en el pizarrón.

¿Qué significa “ir a x: 0 e y: 0”? (Que se ubica en el centro del escenario).

¿Qué significa “bajar lápiz”? (Que se va a realizar un gráfico en la pantalla).

¿Por qué apuntar en dirección 180? (Se utiliza para que la figura se realice hacia abajo. Es importante tener presente que este bloque no es imprescindible).

Teniendo en cuenta la cantidad de giros y movimientos, ¿qué figura se realiza?

(En este caso, el programa realiza un triángulo rectángulo en el medio del escenario.)

Una vez analizado, se propone que lo realicen en la PC. ¿Todos los bloques son necesarios? ¿Cuáles se podrían evitar? Finalmente, pueden buscar otras maneras de realizar triángulos y otras figuras geométricas con Scratch. Al trabajar a partir del análisis y la descomposición de un programa, se espera que los alumnos puedan entender que existen muchas maneras de realizar un mismo programa.

MÓDULO 3. Tramo 2

A tener en cuenta

En el ejercicio se propone el uso de una planilla de cálculo para conocer funciones que ayudan a realizar cálculos de tiempo. La función `HOY()` permitirá determinar la cantidad de días transcurridos desde una fecha determinada hasta el presente. Es importante reflexionar con los chicos sobre cuál es el cálculo real que está realizando la computadora. ¿Cuál es el dato de origen? ¿A qué dato corresponde la función `HOY()`?

La intención es que los alumnos comprendan que lo que se está realizando es la resta entre la fecha actual y la fecha del primer dato. El resultado son los días transcurridos entre ambas instancias.

Otra característica de este ejercicio es que siempre tomará como valor actual la fecha que figura en la computadora. Si guardamos el archivo y lo abrimos al otro día, actualizará el resultado y mostrará los días correspondientes. Por esta razón, la función no sirve para saber los días transcurridos entre dos hechos históricos.

MÓDULO 4. Tramo 2

A tener en cuenta

En este ejercicio se propone, en un principio, que los alumnos sepan que Scratch reconoce como números decimales aquellos que tienen un “punto”, es decir, no se utiliza la coma. Esto es importante porque los números con coma no serán considerados decimales por el programa.

Luego, se propone realizar una pequeña prueba sobre la forma de mostrar los números decimales comprendidos entre dos números.

Por último, se invita a realizar una calculadora de números decimales.

Previamente se debería recordar:

¿Qué datos permiten interactuar con el usuario? (*Los sensores de preguntas*).

¿Qué bloques permiten guardar información durante un tiempo determinado? (*Las variables*).

¿Qué bloques permiten realizar cálculos? (*Los operadores matemáticos*).

Luego, se propone una etapa de exploración para que los alumnos intenten resolver utilizando los bloques nombrados: “realizar una calculadora de números decimales”.

Una de las posibles respuestas podría ser la que se muestra en la captura que está a la derecha.

