

# 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 eso, en *Malabares matemáticos 6*, 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 manera ordenada, secuenciada, lógica y sin ambigüedades, algo que a veces resulta difícil en el ámbito de las Ciencias Sociales.

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 forma; 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 2

#### A tener en cuenta

El programa *Scratch* es un lenguaje de programación creado por el MIT, diseñado para que cualquier persona pueda realizar diferentes programas de manera muy intuitiva, utilizando bloques agrupados en categorías.

Las categorías que se utilizan en este ejemplo son: EVENTO (al presionar la bandera verde) - SENSORES (Preguntas) - VARIABLES - CONTROL (repetir) - OPERADORES (multiplicación).

En esta placa se propone realizar un programa para que los alumnos puedan practicar potencias.

Se sugiere analizar, entre todos, qué operaciones realizamos con las potencias. El exponente es el número que indica las veces que la base aparece como factor. Luego, los alumnos deberán pensar una manera de realizarlas utilizando *Scratch*.

Una opción posible es esta, pero es importante tener en cuenta que hay varias maneras de realizar lo mismo. Se puede invitar a pensar qué cambios podrían hacer para mejorarlo.

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



### MÓDULO 1. Tramo 3

#### A tener en cuenta

Con *Scratch* podemos hacer todo tipo de operaciones matemáticas. En este caso, proponemos realizar una calculadora que determine si un número es primo o no, pero, en lugar de que los alumnos piensen qué bloques utilizar, se les muestra el código dividido en pequeñas partes, y se propone analizar qué función cumple cada una de ellas en la calculadora final. Es importante que puedan reconocer la función de cada una de las variables y su relación, con ejemplos prácticos. Podrían realizarla en *Scratch* y probar con diferentes valores para luego, entre todos, analizar y realizar la conceptualización.

## MÓDULO 2. Tramo 1

### A tener en cuenta

Usamos los metros para medir las longitudes; litros, para medir capacidades, y el tiempo, lo medimos en horas, minutos y segundos. Para medir la capacidad de almacenamiento de información, utilizamos los *bytes*. La unidad más utilizada actualmente es GB (*gigabyte* o *giga*) que equivale a 1.000 millones de *bytes*.

Para calcular la información que se utiliza en las diferentes unidades, dividimos o multiplicamos por 1.024:

- 1 *petabyte* = 1.024 TB
- 1 *terabyte* = 1.024 GB
- 1 *gigabyte* = 1.024 MB
- 1 *megabyte* = 1.024 KB
- 1 *kilobyte* = 1.024 B
- 1 *byte* = 8 bits
- 1 bit = 1 (uno) o 0 (cero)

Si un *byte*, son 8 bits, la base es 8, a diferencia del sistema decimal, cuya base es 10.

- 1.024 es un múltiplo de 8.

$$1.024 : 8 = 128$$

$$128 : 8 = 16$$

$$16 : 8 = 2$$

Este último "2", representa las dos únicas posibilidades: el uno (1) o el cero (0).

Para convertir algo que tenemos en *Gigas* a *Megas*, tendremos que multiplicar por 1.024.

Si de *megabytes* lo queremos pasar a una unidad inferior como, por ejemplo, los *bytes*, primero haremos una multiplicación por 1.024, y después volveremos a multiplicar el resultado por el mismo número.

## MÓDULO 3. Tramo 3

### A tener en cuenta

En esta placa es importante tener en cuenta que Scratch reconoce como números decimales aquellos que tienen un punto, es decir, no utiliza la coma.

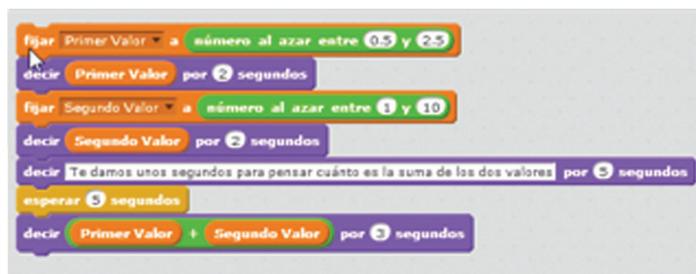
Se propone realizar una calculadora de sumas con números decimales. Estos se generarán aleatoriamente entre el rango que se proponga. Una manera posible es, combinar los bloques de la siguiente manera.

A la variable llamada "Primer Valor", se le adjudica un número al azar (aleatorio) entre el rango comprendido entre 0,5 y 2,5. Luego, lo muestra en la pantalla durante dos segundos.



Una vez establecido el primer valor, se repite lo mismo con un segundo valor. Finalmente, se les da un tiempo para que los alumnos puedan pensar la respuesta y, utilizando la categoría "Operadores", mostrará el resultado.

En el ejemplo, se utiliza un valor decimal y uno, entero. También podrían utilizarse dos valores decimales.



## MÓDULO 4. Tramo 1

### A tener en cuenta

A diferencia de la placa anterior, aquí se propone que el usuario del programa, introduzca el valor con el que desea operar.

Previamente se debería recordar: ¿qué datos nos permiten interactuar con el usuario? Los sensores de preguntas.

¿Qué bloques nos permiten guardar información durante un determinado tiempo? Las variables.

¿Qué bloques nos permiten realizar cálculos? Los operadores matemáticos.

Luego, se propone una etapa de exploración para que los alumnos intenten resolver utilizando los bloques nombrados. ¿El desafío? Realizar una calculadora de números decimales. Esta podría ser una de las posibles respuestas.



## MÓDULO 4. Tramo 4

### A tener en cuenta

BlocksCAD es una página web basada en bloques tipo Scratch que nos permite, entre otras tantas cosas, visualizar e imprimir, con una impresora 3D, diferentes cuerpos geométricos. Trabaja a partir de algunas formas que considera “básicas” como cilindros, esferas y cubos. A partir de estas, combinándola con otros bloques, crea diferentes geometrías.

El entorno lo podemos dividir en tres partes:

1. Área de programa: conjunto de bloques que representan las instrucciones a ejecutar para “renderizar” (hacer) el modelo 3D. Se arrastran desde la barra de bloques y se van encajando unos con otros para determinar la lógica de ejecución-construcción.
2. Barra de bloques: paleta que contiene los bloques que se pueden utilizar en el área de programa. Los bloques se arrastran de una zona a otra.
3. Área de dibujo o renderizado: hacer clic en el botón “Hacer” (o “Render”), el programa ejecuta el modelo 3D a partir de los bloques que aparezcan en el área de programa.

Además, tenemos la típica barra de herramientas para manejar los archivos, determinar las preferencias del entorno o acceder a la ayuda y ejemplos del programa.