

Unidad 1: La escritura como medio para comunicar y almacenar la información

Propósito de la unidad

Los estudiantes exploran diferentes formas de representar ideas, comenzando con el lenguaje natural para terminar con lenguaje computacional. Aplicarán la descomposición de la situación, la abstracción y la repetición de patrones para entregar información y obtener respuestas deseadas. El foco de las actividades de esta unidad es el algoritmo, comenzando con las ideas de secuencia ordenada en la entrega de cierta información para ser replicada, y la estructura de paso a paso para llegar a la programación con Scratch. Algunas preguntas orientadoras son: ¿Podría el computador reemplazar la mente humana? ¿Es posible crear programas computacionales para representar cualquier fenómeno o situación, y que nos ayuden a resolver problemas de todo tipo?

Objetivos de Aprendizaje

OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.

OA a. Construir y evaluar estrategias de manera colaborativa al resolver problemas no rutinarios.

OA d. Argumentar, utilizando lenguaje simbólico y diferentes representaciones, para justificar la veracidad o falsedad de una conjetura, y evaluar el alcance y los límites de los argumentos utilizados.

OA g. Elaborar representaciones, tanto en forma manual como digital, y justificar cómo una misma información puede ser utilizada según el tipo de representación.

Actividad 1: Introducción al pensamiento computacional y programación

PROPÓSITO

Se espera que el estudiante explore –sin usar el computador– conceptos básicos vinculados con pensamiento computacional y programación. En la primera parte, se enfrenta a un problema difícil de resolver, en particular por las condiciones entregadas; sin embargo, al usar la descomposición, la abstracción y el descubrir patrones, pueden encontrar la solución. Se pretende que persevere en buscar soluciones, que generalice el problema y que comprenda el concepto de algoritmo, como una secuencia ordenada de pasos para resolver un problema. Se profundiza en la idea de algoritmos y se introduce el concepto de "programar" por medio de dibujos de cuadrículas y con una simbología simple y cercana.

Objetivos de Aprendizaje

OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.

OA a. Construir y evaluar estrategias de manera colaborativa al resolver problemas no rutinarios.

OA g. Elaborar representaciones, tanto en forma manual como digital, y justificar cómo una misma información puede ser utilizada según el tipo de representación.

Actitudes

- Pensar con perseverancia y proactividad para encontrar soluciones innovadoras a los problemas.

Duración: 18 horas pedagógicas

DESARROLLO

¿CÓMO RESOLVEMOS UN PROBLEMA?

1. Trabajen en grupos para encontrar, en 30 segundos, la suma todos los números naturales entre 1 y 200. Uno de los integrantes toma el tiempo y los otros intentan responder, solos o en conjunto.
 - a. ¿Lo consideraron tan complejo que no intentaron hacerlo?
 - b. ¿Lo intentaron y no alcanzaron a hacerlo?, ¿qué trataron de hacer?
 - c. ¿Lo pudieron resolver?, ¿cómo?, ¿cuál es el resultado?
 - d. Si no lo resolvieron, ¿qué es lo que no comprenden aún?
 - e. Prueben ahora con otros números: (de 1 a) 500, 1 000 o 100 000.

2. Ahora quieren que la solución funcione para diferentes números naturales, por ejemplo, un número n : ¿qué secuencia de pasos funciona para cualquier número natural (n)?
 - a. ¿Cómo orientan las preguntas y estrategias a resolver el problema?
 - b. ¿Podrían usar esta estrategia para resolver otros problemas?
3. En sus grupos, comenten lo que hacen cuando se les presentan problemas en sus vidas y reflexionen en torno a las siguientes preguntas: ¿Cómo actúan? ¿Cuál es la secuencia o las preguntas que harían al enfrentar un problema? Anoten en el cuaderno las secuencias que propusieron y compartan con toda la clase.

CONCEPTOS BÁSICOS Y PROGRAMACIÓN EN PAPEL CUADRICULADO

1. ¿Qué entienden por algoritmo? Discutan en el grupo y redacten su mejor aproximación al concepto.
2. ¿Qué significa programa computacional? Discutan en el grupo y completen su mejor aproximación al concepto.
3. Observen la siguiente imagen, comenzando en la parte superior izquierda. ¿Cómo le darían instrucciones a una persona para que replique los cuadrados pintados de negro de la Figura 1 en la Figura 2, sin que dicha persona pueda ver la Figura 1? Prueben dando instrucciones entre los integrantes del grupo.

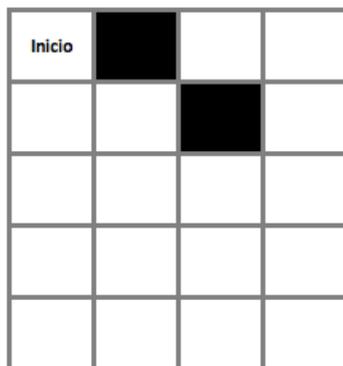


Figura 1: Cuadrilado deseado

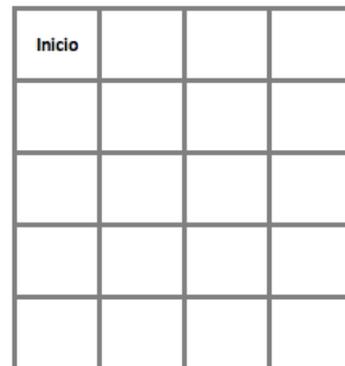


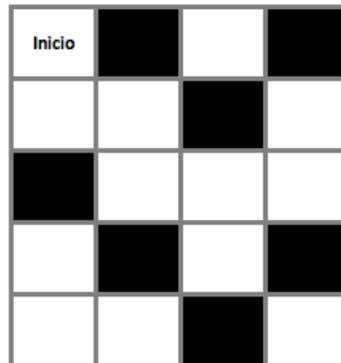
Figura 2: Cuadrilado réplica de Figura 1

4. Ahora solo pueden dar las siguientes instrucciones:

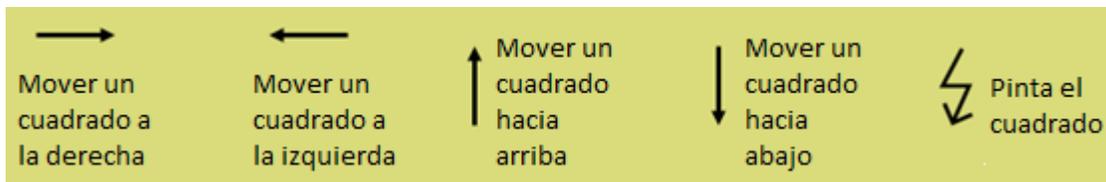
- Mueve un cuadrado a la derecha
- Mueve un cuadrado a la izquierda
- Mueve un cuadrado hacia arriba
- Mueve un cuadrado hacia abajo
- Pinta el cuadrado

Un integrante del grupo da indicaciones a otro para que dibuje la siguiente figura.

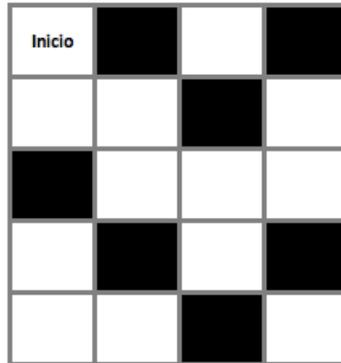
Los demás observan y proponen diferentes alternativas. Repitan hasta obtener la menor cantidad de instrucciones.



5. ¿Qué pasaría si, en lugar de escribir una frase completa para cada instrucción, colocamos un símbolo?, por ejemplo:



6. Usando los símbolos anteriores:
a. Escriban las instrucciones para el siguiente ejemplo:



- b. ¿Cuáles serían sus instrucciones?

CONSTRUYENDO LOS CONCEPTOS DE LENGUAJE COMPUTACIONAL

Si comparan su propuesta para simplificar el texto con la de flechas, ¿qué aspectos están a favor y en contra de cada una?

1. En ciencias de la computación, ¿qué nombre piensan que reciben las Formas 1 y 2?

Forma 1:

“Mover un cuadrado a la derecha”

“Pintar el cuadrado”

“Mover un cuadrado a la derecha”

“Mover un cuadrado hacia abajo”

“Pintar el cuadrado”

Forma 2:



2. ¿Qué nombre piensan que recibe cada expresión de Forma 1 y Forma 2?

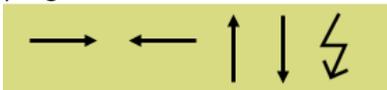
Por ejemplo, expresión de Forma 1:

“Mover un cuadrado a la derecha”

Por ejemplo, expresión de Forma 2:

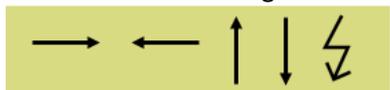


Los grupos definen que uno de sus integrantes asumirá el rol de diseñador de algoritmo y otro será el programador. El diseñador verbaliza cada instrucción y el programador la convierte en símbolos



El programador nunca puede ver la figura original de cuadrículas.

- El diseñador selecciona una figura de las “Fichas de trabajo” proporcionadas por su profesor y le inventa un código de 6 dígitos de letras y números (ejemplo C35AD7). El programador no puede ver esa figura.
- El diseñador da al programador las instrucciones verbalizadas del cuadrículado seleccionado. El diseñador tiene que decir expresiones como: “desde inicio, mueve un cuadrado a la derecha, mueve un cuadrado hacia abajo, pinta el cuadrado...”.
- El programador convierte la secuencia que le dicta el diseñador en un programa, empleando la hoja “Secuencia de Programación”, sin ver la figura seleccionada y utilizando símbolos



- Al terminar, diseñador y programador intercambian roles; cada uno debe diseñar los algoritmos de dos figuras y programar otras dos.
- ¿Qué pueden transferir de este aprendizaje a su vida diaria? ¿Cuándo usan abstracción en situaciones cotidianas? ¿Cuándo usan la secuencia lógica de acciones?

OTRA FORMA DE CODIFICAR, REPRESENTAR Y COMUNICAR

Posiblemente algo sabes del código morse; es un sistema que representa letras y números mediante señales emitidas de forma intermitente (punto: señal corta; raya: señal larga).

A	.-	J	.-.-.-	S	...-	2	..-.-
B	-...-	K	-.--	T	-	3	...--
C	-.-.-	L	.-...-	U	...-	4-
D	...-	M	--	V	...-	5
E	.	N	--.	W	---	6	-----
F	...-	O	---	X	...-	7	-----
G	...-	P	...-	Y	...-	8	-----
H	Q	...-	Z	...-	9	-----
I	..	R	...-	1	0	-----

Figura: Representación de letras y números en código morse

En grupos, respondan si se puede hacer un programa computacional para “traducir” una palabra o frase a código morse y viceversa. Presenten las instrucciones para programar en clave morse; primero intenta dar instrucciones a un compañero –utilizando sonidos o movimientos– para que represente una palabra, y que demás traduzcan la instrucción.

ORIENTACIONES PARA EL DOCENTE

1. Al hacer la puesta en común, plantee las preguntas: ¿Qué les pareció la actividad? ¿Quiénes lo consideraron tan complejo que no intentaron hacerlo? ¿Quiénes lo intentaron y no alcanzaron? ¿Qué trataron de hacer? ¿Quién pudo resolver el problema? Si alguien lo resolvió, pregunte cómo lo hizo y cuál es el resultado. Si dividimos el problema en otros más pequeños, ¿será más fácil resolverlo?
2. Se sugiere guiar a los jóvenes con las siguientes preguntas: ¿Cuánto es $1+100$? ¿Y $2+99$? ¿Y $3+98$? Plantee qué significa generalizar: ¿Cómo lo hacemos? ¿Qué se mantiene? ¿Qué cambia? ¿Puedes observar algún patrón? ¿Cuántos pares de sumas tendríamos? ¿Cuál es el último par que tenemos? ¿Cuál sería el resultado final?
3. Explíqueles que de eso se trata el pensamiento computacional e invítelos a mirar el video <https://www.youtube.com/embed/Vk7fJ9ExITY>. Cuando termine, comente con ellos qué entendieron sobre pensamiento computacional y cómo pueden aplicar esos conceptos y estrategias en su vida.
4. Se sugiere anotar qué creen los estudiantes sobre los términos “algoritmo” y “programa”, además de los respectivos conceptos, sus definiciones y el ejemplo presentado. Después haga lo mismo respecto de “instrucción” o “código”. Escriba las correspondientes definiciones acláreles que, en programación, son similares.
5. Se sugiere el siguiente indicador para evaluar formativamente los aprendizajes:
 - Identifican las partes que componen una situación y que permiten elaborar instrucciones.

RECURSOS Y SITIOS WEB

Sitios web sugeridos para estudiantes y profesores

- Esto es pensamiento computacional, video youtube:
<https://www.youtube.com/embed/Vk7fJ9ExITY>
- Plataforma para iniciar conceptos de programación:
www.code.org

Actividad 2: ¿Cómo abordar un problema y la forma de generar una estrategia de solución?

PROPÓSITO:

Se espera que, desde situaciones cotidianas de la vida, los estudiantes identifiquen algunos conceptos de programación y los relacionen tanto con la programación en Bloques de Scratch, como con diagramas. Se pretende que se interesen por las posibilidades que ofrece la tecnología, y que conozcan y adquieran algunas estrategias y habilidades para resolver problemas.

Objetivos de Aprendizaje

OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.

OA a. Construir y evaluar estrategias de manera colaborativa al resolver problemas no rutinarios.

OA d. Argumentar, utilizando lenguaje simbólico y diferentes representaciones para justificar la veracidad o falsedad de una conjetura, y evaluar el alcance y los límites de los argumentos utilizados.

Actitudes

- Interesarse por las posibilidades que ofrece la tecnología para el desarrollo intelectual, personal y social del individuo.

Duración: 18 horas pedagógicas

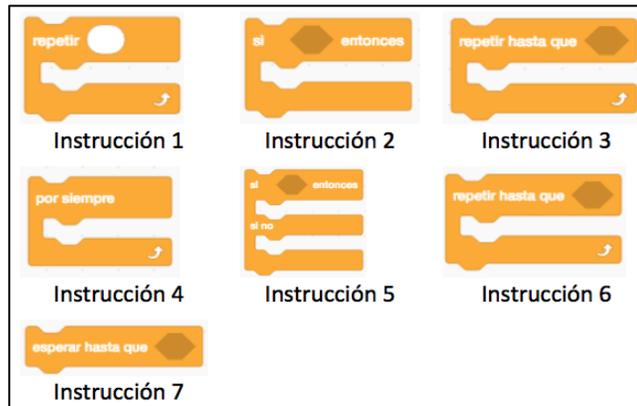
DESARROLLO

ALGUNAS INSTRUCCIONES CON BLOQUES DE SCRATCH

Muchas veces se utiliza el lenguaje Scratch –desarrollado por el Massachusetts Institute of Technology (MIT)– para programar. Sus instrucciones o comandos son bloques, y programar es unir o ensamblar bloques, ¡como en un Lego! Esto disminuye las barreras de entrada para concentrarse en los conceptos de la programación y en resolver problemas, sin tener que aprender términos ni sintaxis especiales.

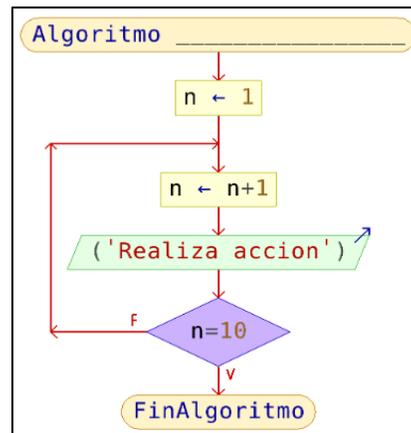
1. Nuestro cuerpo, y en especial nuestro cerebro, actúa en forma automática para hacer algunas acciones, como: “pelaré 10 manzanas para el postre”; “caminaré hasta llegar a la casa de mi amiga”; “el ser humano respira durante toda su vida”.
 - a. ¿Reconoces algunas instrucciones utilizadas en programación en esas acciones?, ¿cuáles?
 - b. De las instrucciones en Scratch que están en la siguiente imagen, ¿cuál(es) usarías para resolver “pelaré 10 manzanas para el postre”? Instrucción _____

- c. ¿Cuál(es) para “caminaré hasta llegar a la casa de mi amiga”? Instrucción _____
 d. ¿Cuál(es) para “el ser humano respira durante toda su vida”? Instrucción _____



2. El siguiente diagrama representa un algoritmo y una de las tres acciones mencionadas anteriormente. En la línea ubicada junto a la palabra “Algoritmo”, escribe el nombre que representa la acción que piensas que se efectúa con ese algoritmo.

- a. ¿Qué representa $n \leftarrow 1$?; y $n \leftarrow n + 1$?
 b. ¿Qué haría esta instrucción ('Realiza accion')?
 c. ¿Qué significan el rombo y las flechas con “F” y “V”?
 d. ¿Da lo mismo poner primero la instrucción $n \leftarrow n + 1$ y después $n \leftarrow 1$?
 e. ¿Qué hace el algoritmo?



3. ¿Usas algoritmos en matemática?, ¿y en la vida cotidiana? Da algunos ejemplos de ambos casos.

LOS FACTORES DE UN NÚMERO

1. Hagan en su cuaderno un algoritmo que permita decir los factores de un número. ¿Han resuelto algo parecido o que se relacione?
 a. ¿Qué otras relaciones pueden hacer entre el problema actual y otro que conozcan?
 b. ¿Cómo sería el programa? ¿Necesita datos de entrada? ¿Cuál es la salida?
 c. Prueben el programa con los siguientes números: 24 y 13

- d. Encuentren los valores faltantes en la siguiente tabla:

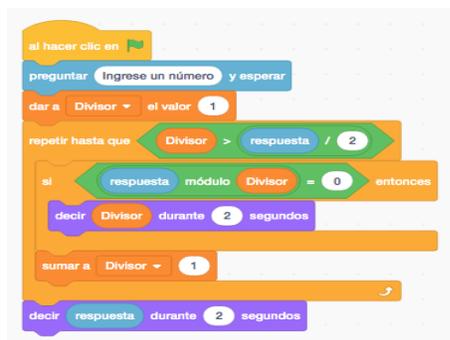
24	12	0
37		5
16		0
	7	3
48	6	
73	8	

- e. ¿Qué tienen en común los números cuyo resto es cero?, ¿y los que tienen resto distinto de cero?
- f. ¿Cuándo un número es factor de otro?
2. Análisis del problema a partir de preguntas.
- Establezcan los pasos: ¿Cuáles son las partes de la solución? ¿Cómo operan? ¿Cómo se relacionan entre sí?
 - Deténganse en las partes clave y respondan: ¿Qué hay que resolver antes de tenerlo todo para desarrollar el programa?
 - Reflexionen en relación a un número dado (por ejemplo, 124): ¿Cómo encontrar el primer número que lo divide? ¿Y el siguiente? ¿Y el siguiente a este? ¿Hasta cuándo hay que repetir este proceso?
 - Unan las partes y organícenlas en un todo; es decir, denle una estructura, una secuencia ordenada que permita resolver el problema. ¿La pueden describir en pocas palabras?
3. Ahora hagan el programa con Scratch. Abran Scratch. ¿Cómo ingresan el número del cual desean encontrar los factores?
- ¿Cómo manejarán los divisores?
 - ¿Qué instrucción les permite implementar un proceso para buscar los divisores?
 - ¿Con qué operación matemática pueden saber si un número es divisor o no del número ingresado?
 - ¿Qué instrucción pueden emplear en Scratch para resolverlo? ¿Qué pasa si es divisor? ¿Y si no lo es?
 - Hagan un programa en Scratch que les permita encontrar los factores de un número entero positivo.
 - Generen un programa.
 - Pruébenlo y depúrenlo:
 - Ingresen valores (números) de los cuales conozcan sus factores; luego busquen valores extremos, ¿qué pasa con 0, con 1, con 10 598, con número negativos?
 - Mejoren su programa y validen los datos: si el usuario ingresa un número negativo o cero, que lo indique y que el sistema pida que se ingrese un número entero positivo, mayor que cero.

4. Reflexionen con sus compañeros respecto de:
 - a. ¿Cómo enfrentas y resuelves un problema para programarlo?
 - b. ¿Qué relación encuentras entre la programación y la matemática, más allá de hacer un programa que trabaje concepto(s) matemático(s)?
 - c. ¿Qué de este conocimiento puedes transferir para resolver problemas cotidianos de tu vida?

ORIENTACIONES PARA EL DOCENTE

1. Se sugiere indicar que un diagrama de flujo es la representación gráfica de un algoritmo. Recuerde la definición de algoritmo como una secuencia de pasos ordenada para resolver un problema; en este caso, pelar manzanas. Pídeles que le expliquen el diagrama (representa la primera acción y se realiza un proceso 10 veces).
2. En la actividad se utiliza el lenguaje de programación por bloques Scratch. Se recomienda que usen tanto el programa en el computador como la plataforma.
3. Una solución en Scratch al problema es:



4. Se sugiere los siguientes indicadores para evaluar formativamente los aprendizajes:
 - Resuelven problemas con estrategias como dividirlo en problemas menores.
 - Usan proposiciones lógicas para controlar el flujo al ejecutar un programa de computación.

RECURSOS Y SITIOS WEB

Sitios web sugeridos para estudiantes y profesores

- Sitio de Scratch. Es un lenguaje de programación visual, desarrollado por el grupo Lifelong Kindergarten, para programar *online*
<https://scratch.mit.edu/>
- Descargar programa de Scratch
<https://scratch.mit.edu/download>

Actividad 3: ¿Cómo la matemática y la programación favorecen el diseño y desarrollo de modelos?

PROPÓSITO

Mediante el modelamiento, se mostrará el potencial de la programación computacional y su ayuda a la humanidad. Se pretende que los estudiantes conozcan qué son los modelos –en especial, al reunir áreas como la programación computacional, la matemática y el modelamiento para implementar una solución a la astronomía– y que entiendan cómo la matemática, las herramientas disponibles y la programación aportan al aprendizaje, la elaboración y el desarrollo de problemas.

Objetivos de Aprendizaje

OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.

OA g. Elaborar representaciones, tanto en forma manual como digital, y justificar cómo una misma información puede ser utilizada según el tipo de representación.

Actitudes

- Aprovechar las herramientas disponibles para aprender y resolver problemas.

Duración: 12 horas pedagógicas

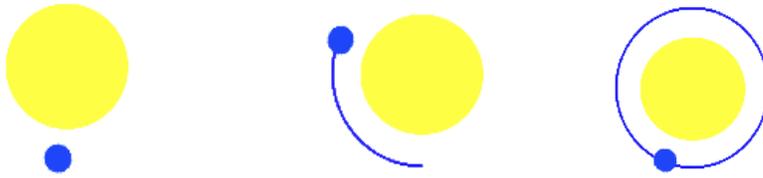
DESARROLLO

EL MOVIMIENTO DE LA TIERRA EN TORNO AL SOL

La imagen que conocemos del sistema solar es un modelo. Su primera versión proponía un sistema heliocéntrico, colocando el Sol al centro del universo conocido; es decir, describía un modelo posicional en que los planetas que giraban en torno al Sol con una trayectoria circular. Busca información con respecto a: quién o quiénes hicieron esta propuesta; en qué periodo; en qué consistió y qué planetas incluía.

Conexión interdisciplinaria:
Ciencias para la ciudadanía.
 OA c, OA d,
 3° y 4° medio

- Basado en el modelo del sistema solar, haz un programa en Scratch que muestre el Sol al centro y la Tierra desplazándose en forma circular a su alrededor.



- Abre Scratch y crea un objeto nuevo con la opción pintar. Haz un círculo amarillo, simulando el Sol, y nómbralo "Sol".
- Crea un nuevo objeto y, de manera similar, haz un círculo azul de menor tamaño que represente la Tierra. Nómbralo "Tierra".
- ¿Cómo es el desplazamiento de la Tierra?, ¿qué forma tiene?, ¿en qué dirección va?
- Ubícate en el objeto Sol y, como programación, pon el "Evento"  y haz que se posicione en .

- Ubícate en el objeto Tierra; para comenzar tu programa, coloca el

"Evento"  y .

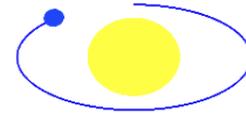


- Según la posición que quedaron la Tierra y el Sol, la Tierra ¿debe desplazarse hacia la izquierda o derecha? Luego de la instrucción  coloca apuntar en dirección según lo que corresponda.
- Para hacer un ciclo que simule una órbita circular de la Tierra en torno al Sol, puedes hacer un ciclo que repita por siempre que la Tierra  y .
- Para dejar un rastro de la órbita de la Tierra, agrega las opciones de lápiz correspondientes para que, cuando comience, borre todas las marcas y luego pongas *lápiz* para que, al desplazarse, deje la marca de la órbita.
- Tu programa ¿puede considerarse un modelo? ¿Por qué sí? ¿Por qué no?

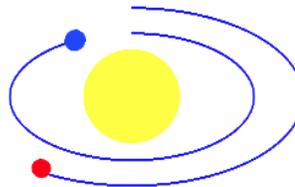


MEJORA DEL PROGRAMA DEL MODELO DEL SISTEMA SOLAR

1. Investiga sobre el sistema solar: qué planetas lo componen, su posición respecto del Sol, la forma de la trayectoria de la órbita de la Tierra.
2. Selecciona uno de los programas que hicieron individualmente y modifiquen el programa del objeto “Tierra”, de manera que su órbita sea elíptica.



3. ¿Existe algún patrón o regularidad en el modelo desarrollado?
4. ¿Se puede transferir ese modelo a otra experiencia?
5. Prueba creando un nuevo objeto: dibuja el planeta “Marte” y ponle ese nombre. Copia el programa del objeto “Tierra” en este nuevo objeto y realiza los cambios para que quede a una distancia mayor del “Sol” respecto de la “Tierra”, y haz la misma órbita elíptica alrededor del Sol.



6. ¿Para qué nos puede servir hacer modelos usando la programación y los computadores? Si les pidieran hacer un modelo para predecir la contaminación ambiental de una ciudad, ¿qué elementos considerarían para construirlo?

ORIENTACIONES PARA EL DOCENTE

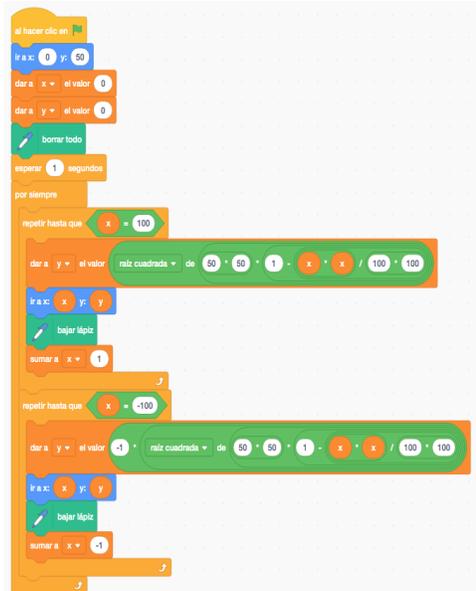
1. Inicialmente podrían resolver el movimiento de la Tierra en sentido del movimiento de las manecillas del reloj; muéstreles algunos videos y pídales que investiguen al respecto.
2. Se propone un problema que se va haciendo más complejo; le sugerimos apoyarlos en este proceso, regulando las variables que se incorporan en el modelo. Discuta con ellos sobre la solución a la ecuación de la elipse, donde, al sacar las soluciones de la raíz cuadrada, se programa tanto la solución positiva como la negativa. Si no implementan la solución negativa, déjelos que vean qué pasa, cómo es la órbita del planeta, y pregúnteles por qué pasa eso.
3. Pruebe variando el modelo. Ya tienen el sistema con una órbita de la Tierra y Marte alrededor del Sol de forma elíptica; ahora, pídales que agreguen la Luna y programen su órbita alrededor de la Tierra. Al período de la órbita que ya tienen, agreguen el período de la rotación de los dos planetas. Recuérdeles considerar las proporciones de los tamaños y las distancias, y que incluyan estos datos. Puede preguntarles sobre las mejoras al modelo.

4. Una solución al problema inicial en Scratch es:

Órbita circular



Órbita elíptica



5. Se sugiere los siguientes indicadores para evaluar formativamente los aprendizajes:

- Usan proposiciones lógicas para el control del flujo en la ejecución de un programa de computación.
- Ejecutan ciclos (*loops*) a partir de un patrón o regularidad que se repite en una secuencia de un programa.

RECURSOS Y SITIOS WEB

Sitios web sugeridos para estudiantes y profesores

- Video del sistema solar
<https://www.youtube.com/watch?v=z8aBZZnv6y8>
- Las órbitas y las leyes de Kepler
<https://solarsystem.nasa.gov/resources/310/orbits-and-keplers-laws/>
- El sistema solar
<https://solarsystem.nasa.gov/solar-system/our-solar-system/overview/>
- Sistema solar
https://es.wikipedia.org/wiki/Sistema_solar
- Espacio de la Nasa
<https://spaceplace.nasa.gov/sp/>

Actividad 4: Ayudar a unos ratones a salir de un laberinto

PROPÓSITO

Se propone a los estudiantes que programen un problema complejo: una primera aproximación a cómo se está implementando el uso de sensores para resolver problemas reales de nuestra vida. Inicialmente, se los invita a que entiendan el problema, lo analicen, se lo imaginen, piensen las posibilidades y dificultades, para luego pasar a la “acción”: hacer un programa en Scratch. Por medio de la programación y, en especial, el uso de sensores, pueden visualizar y comprender cómo se han construido las soluciones que estamos viviendo. Es una buena oportunidad para discutir sobre tecnologías de la inteligencia artificial, donde las personas siguen haciendo la diferencia con su creatividad, con el hecho de pensar y con sus habilidades y sentimientos.

Objetivos de Aprendizaje

OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.

OA a. Construir y evaluar estrategias de manera colaborativa al resolver problemas no rutinarios.

OA d. Argumentar, utilizando lenguaje simbólico y diferentes representaciones para justificar la veracidad o falsedad de una conjetura, y evaluar el alcance y los límites de los argumentos utilizados.

Actitudes

- Aprovechar las herramientas disponibles para aprender y resolver problemas.

Duración: 12 horas pedagógicas

DESARROLLO

UN PROBLEMA COMPLEJO Y SUS IDEAS INICIALES

Se sugiere comenzar proyectando el video: https://youtu.be/SvREIVw5_fY para reflexionar con el curso a partir de las preguntas: ¿Cómo funciona una aspiradora autónoma? ¿Cómo funcionan los autos que se manejan solos? ¿Cómo lo hace un auto para avisar cuando estás cerca de un objeto?

Conexión interdisciplinaria:
Ciencias para la Ciudadanía.
OA f, 3° y 4° medio

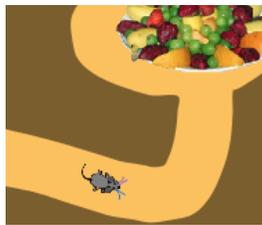
1. Desarrollen un programa en Scratch que simule el desplazamiento de ratones en el interior de un laberinto. Habrá varios ratones que comienzan su trayectoria desde distintos puntos del laberinto y se desplazan en forma autónoma, buscando un plato de frutas. Lo importante es que no se programa la ruta de salida, sino una estrategia para que puedan encontrar la fruta en forma independiente, sin importar el laberinto que tengamos; es decir, al cambiar de laberinto –su forma– el ratón se desplazará sin necesidad de cambiar el programa. Respondan:
 - a. ¿Cómo se agrupan los ratones? ¿Cómo se desplazan?

- b. ¿Qué datos están disponibles? ¿Qué información nos dan? ¿Qué información es importante?
 - c. ¿Cuáles son las condiciones del enunciado del problema? ¿Hay condiciones que no se dan, pero que tendrás que considerar al hacer el programa?
 - d. ¿Cuáles son las restricciones del problema? ¿Hay restricciones que no se dan, pero que tendrás que considerar al hacer el programa?
 - e. ¿Cómo imaginas el programa ejecutado? ¿Cómo funcionaría?
 - f. ¿Debes ingresar información? En caso afirmativo, ¿cuál? En caso negativo, ¿por qué no?
 - g. ¿Cuál es la salida? ¿Cómo se verá tu programa? Mientras mejor imagines tu programa funcionando, más claridad tendrás para resolverlo.
2. ¿En qué se parecen, se diferencian y cómo te pueden ayudar, las estrategias de solución de problemas que implementaste en esta actividad, respecto de las que aplicas en problemas de otras asignaturas o de tu vida diaria?

LA SOLUCIÓN DE UN PROBLEMA COMPLEJO CON PROGRAMACIÓN

1. Hagan grupos de 2 o 3 personas y compartan sus reflexiones, apuntes y dibujos. Compartan el trabajo que acaban de hacer. ¿Qué entiende cada uno sobre el programa para el desplazamiento del ratón?
 - a. Sobre el movimiento de los ratones, ¿qué se dice en el problema y qué investigaste sobre cómo se mueven los ratones?
 - b. ¿Qué quiere decir “desplazamiento autónomo”?
 - c. Se pide que no se programe a los ratones con la ruta de salida del laberinto; es decir, se desea que sean autónomos. Esto se ha aplicado a problemas reales; por ejemplo: hay aspiradoras robóticas y se avanza en la elaboración de autos que se manejan solos. ¿Qué características de la aspiradora y el automóvil los hacen autónomos?
2. Una aspiradora robótica debe adecuarse a diferentes espacios de distintas casas, y los automóviles se deben desplazar en distintos lugares con diferentes condiciones. ¿Cómo funcionan para que se desplacen en cualquier lugar?
3. Observen los objetos ratón y frutas, miren sus disfraces. ¿Hay algo que les llame la atención del ratón? ¿Por qué? En el escenario pueden ver que hay tres fondos de laberintos diferentes. Cuando todos los ratones llegan a las frutas, se cambia el laberinto (siguiente fondo) y los ratones regresan a su posición original.

- a. El siguiente código, ¿puede ser parte de la solución para nuestro problema?
Si el ratón está en esta posición y queremos que este código, ¿le permite hacerlo?
llegue a la fruta...



- b. ¿Qué pasa si cambiamos el laberinto? ¿Qué tienen en común estos laberintos?
- c. Piensen qué condiciones deben tener el ratón y los laberintos para encontrar la solución. Realicen el programa en Scratch para solucionar el problema.
4. Agreguen tres ratones, de manera que recorran el laberinto y lleguen hasta la fruta. Cambien los valores para los bloques de ,  y . Observen qué pasa. ¿Pueden mejorar la simulación de su programa? Profesor y alumnos reflexionan sobre las siguientes preguntas:
- ¿Cómo ha sido el impacto de las tecnologías digitales en los contextos sociales, económicos y culturales?
 - ¿Cómo nos ayuda la lógica matemática a resolver problemas de programación? ¿Y a pensar?
 - ¿En qué pueden ayudar las máquinas a las personas en su vida cotidiana?

ORIENTACIONES PARA EL DOCENTE

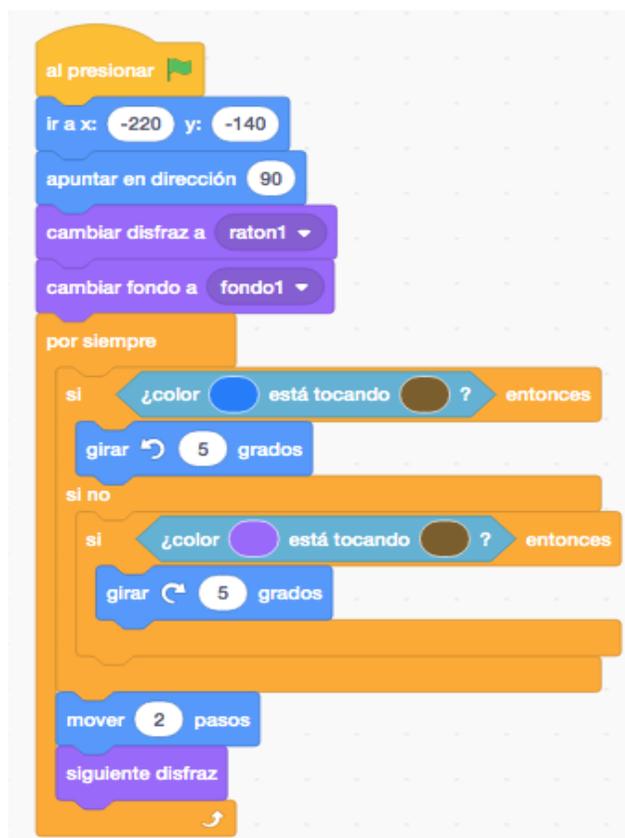
- Se sugiere que los estudiantes programen en su mente, antes de ir directamente al uso de la aplicación. Se los puede invitar a que entiendan y analicen el problema, piensen en la solución y busquen alternativas; es decir, que preparen los elementos para que puedan desarrollar el programa y resolver el problema.
- Es importante que entiendan la diferencia entre decirle al objeto (el ratón) los pasos para salir del laberinto, y programar una estrategia que le sirva para desplazarse autónomamente en el laberinto, sin importar su forma.
- La idea es que piensen cómo se puede desplazar y hacer acciones a un objeto que no pueden "ver"; en este caso, el ratón. Seguramente aparecerán soluciones como usar cámaras de video; discuta con ellos si eso podría servir. Si surgen ideas interesantes, pero complejas de implementar en este caso, invítelos a pensar en soluciones más simples de ejecutar.

Asegúrese de tener acceso a internet o de tener los archivos del video “Ratones_laberinto” y del documento base de Scratch “Actividad_Scratch.sb3”.

4. Si los alumnos no logran usar sensores, pídeles que miren los diferentes grupos de bloques de Scratch. Hágales preguntas acerca de cómo pueden servir para buscar la solución. Puede consultarles por los colores de los bigotes del ratón, porque los del lado derecho son diferentes de los del izquierdo. En este caso, se debe controlar que si el color azul (bigotes del lado derecho) toca el color café (color que está fuera del camino del laberinto), entonces el ratón debe corregir su camino girando algunos grados a la izquierda. Lo mismo sucede con los bigotes del lado izquierdo, pero tendrá que girar hacia la derecha.



5. Una parte de la posible solución para un ratón es:



Para ver la solución completa, ir a <https://scratch.mit.edu/projects/304251756/editor/>

- Para implementar la solución con otros ratones, se debe copiar el objeto y su código. Puede que los estudiantes hagan la copia de objetos, pero comiencen todos desde la misma posición. De ser así, no se verá los diferentes objetos desplazándose por el laberinto. Es interesante discutirlo con ellos; pregúnteles qué puede estar sucediendo. Deberían analizar que, al partir todos del mismo lugar con el mismo código, desplazándose a la misma velocidad y girando igual, los ratones avanzarán superpuestos (uno sobre el otro). Para que no comiencen del mismo lugar, se debe

modificar los valores de 

- Para que los ratones se ubiquen en su posición inicial (original) al cambiar de laberinto, una opción es implementar el siguiente código en los programas de cada objeto ratón:



- Para cambiar los fondos cuando lleguen a la fruta, una solución es colocar el siguiente programa en el objeto fruta:



- Se sugiere los siguientes indicadores para evaluar formativamente los aprendizajes:
 - Resuelven problemas que involucran ejecutar programas mediante lenguaje de bloques.
 - Elaboran representaciones como un pseudocódigo descrito por un diagrama de flujo.
 - Programan en pseudocódigo y lo exportan en un lenguaje simbólico.

RECURSOS Y SITIOS WEB

Sitios web sugeridos para estudiantes y profesores

- Video que muestra un ejemplo de la actividad
https://youtu.be/SvREIVw5_fY
- Plataforma Scratch
<https://scratch.mit.edu>
- Solución del problema
<https://scratch.mit.edu/projects/304251756/editor/>
- Estudio de cien años de inteligencia artificial (Universidad de Stanford)
<https://ai100.stanford.edu/2016-report>
- Otros problemas de ratones en video
https://youtu.be/Pj3KB_SgvDQ

Actividad de Evaluación

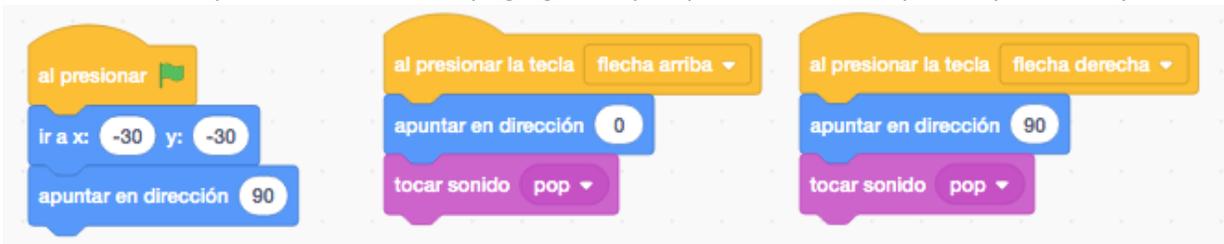
Objetivos de Aprendizaje	Indicadores de evaluación
<p>OA 1. Aplicar conceptos de Ciencias de la Computación –abstracción, organización lógica de datos, análisis de soluciones alternativas y generalización– al crear el código de una solución computacional.</p> <p>OA a. Construir y evaluar estrategias de manera colaborativa al resolver problemas no rutinarios.</p> <p>OA d. Argumentar, utilizando lenguaje simbólico y diferentes representaciones para justificar la veracidad o falsedad de una conjetura, y evaluar el alcance y los límites de los argumentos utilizados.</p> <p>OA g. Elaborar representaciones, tanto en forma manual como digital, y justificar cómo una misma información puede ser utilizada según el tipo de representación.</p> <p>OA k. Analizar y evaluar el impacto de las tecnologías digitales en contextos sociales, económicos y culturales.</p>	<ul style="list-style-type: none"> • Identifican las partes que componen una situación y permiten elaborar instrucciones. • Usan proposiciones lógicas para el control del flujo en la ejecución de un programa de computación. • Ejecutan ciclos (<i>loops</i>) a partir de un patrón o regularidad que se repite en una secuencia de un programa. • Resuelven problemas que involucran ejecutar programas mediante lenguaje de bloques. • Elaboran representaciones como un pseudocódigo descrito por un diagrama de flujo. • Programan en pseudocódigo y lo exportan en un lenguaje simbólico.

Duración: 6 horas pedagógicas

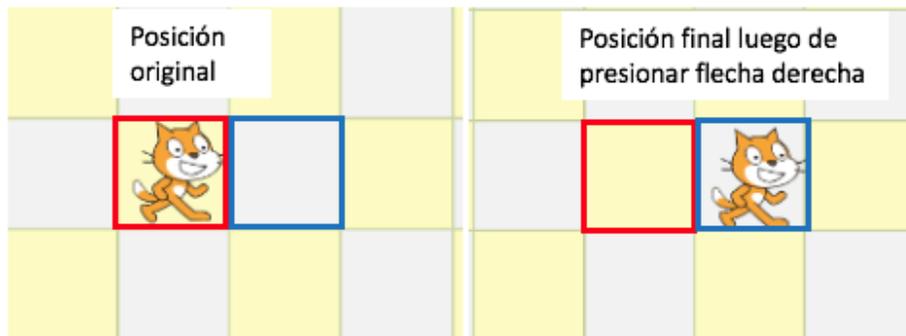
Las siguientes actividades se pueden usar como ejemplos de evaluaciones para la unidad 1, cada una por sí misma o en conjunto. Se sugiere delimitar la evaluación según el contexto y el tiempo disponible.

1. Pida a los estudiantes completar un juego “El gato y el ratón”, en el cual el gato debe perseguir al ratón. Si lo atrapa, emite un sonido; de lo contrario, el ratón se escapa. El gato tiene 3 segundos para atrapar al ratón; de lo contrario, saltará a un nuevo recuadro.
 - a. Describe los movimientos de los objetos gato y ratón.
 - b. Prueba diferentes movimientos para que el ratón se mueva entre los cuadrados en forma aleatoria, y que el gato se mueva al usar las teclas (arriba, abajo, a la izquierda y a la derecha). Nota que el problema es que se deben mover de cuadro en cuadro y estos tienen una distancia de 60 pasos () para el gato y la misma cantidad para el ratón, pero en forma aleatoria, saltando de a uno, dos o más cuadros hacia arriba o hacia abajo.

2. Pídeles que desarrollen un programa en Scratch sobre un juego “El gato y el ratón”, en el cual el gato debe perseguir al ratón. Si lo atrapa, emite un sonido; de lo contrario, el ratón se escapa. El gato tiene 3 segundos para atrapar al ratón; de lo contrario, el ratón saltará a un nuevo recuadro. En https://youtu.be/Pj3KB_SgvDQ hay un video de lo que se espera como resultado final.
- Resuelve los movimientos del gato, que se puede desplazar usando las flechas del teclado hacia arriba, hacia abajo, izquierda y derecha. No es necesario que se mueva entre cuadrados; es decir, no se requiere que avance de un cuadrado a otro.
 - Para esto, carga el archivo <https://scratch.mit.edu/projects/304287041/> y deja una copia en tu computador. Aquí encontrarás el fondo cuadrículado y los objetos gato y ratón. Ahora solo programa el gato. En el objeto gato encontrarás parte del código que deberás completar. Tienes iniciado el código para que el gato se ubique siempre desde un mismo lugar y parte del código para mover flecha arriba y a la derecha. Falta completar este código para que el gato se mueva al presionar las flechas, y agregar bloques para las flechas izquierda y hacia abajo.



- Ahora desarrolla el código para que el gato se desplace entre cuadrados, de uno en uno, al presionar las flechas arriba, abajo, izquierda y derecha. Observa la siguiente figura: muestra un ejemplo del desplazamiento deseado. Recuerda grabar tu archivo.



PAUTA DE EVALUACIÓN

Criterios de evaluación	Niveles de logros		
	Completamente logrado	Se observa aspectos específicos que pueden mejorar	No logrado por ausencia o no se puede entender nada
Resuelven problemas, usando estrategias como dividir un problema en otros menores.			
Usan proposiciones lógicas para el control del flujo en la ejecución de un programa de computación.			
Ejecutan ciclos (<i>loops</i>) a partir de un patrón o regularidad que se repite en una secuencia de un programa.			
Resuelven problemas que involucran ejecutar programas mediante lenguaje de bloques.			
Elaboran representaciones como un pseudocódigo, descrito por un diagrama de flujo.			
Programan en pseudocódigo y lo exportan en un lenguaje simbólico.			