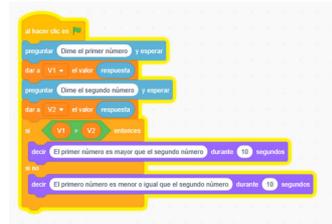


Sentido algebraico y pensamiento computacional



Mónica Arnal Palacián
marnalp@unizar.es



Introducción

Definición y consideración **NO ÚNICA**.

El **pensamiento computacional** es la manera de razonar que nos permite enfrentarnos a un problema sobre unos datos con el objetivo de que un ordenador lo resuelva (Palop y Díaz, 2024).



Introducción

La resolución de problemas entronca con los demás ejes de la competencia matemática: el razonamiento y el **pensamiento computacional**, la representación de objetos matemáticos, el manejo y la comunicación empleando lenguaje matemático y las destrezas socioafectivas.

El **pensamiento computacional** ofrece un acercamiento distinto a la resolución de problemas que encuentra puntos en común con el proceso clásico dentro de las matemáticas.

Introducción

CE.M.4. Utilizar el **pensamiento computacional** organizando datos, descomponiendo en partes, reconociendo patrones, generalizando e interpretando, modificando y creando algoritmos, en situaciones de aprendizaje con el andamiaje adecuado, para modelizar y automatizar situaciones cercanas y significativas para el alumnado.

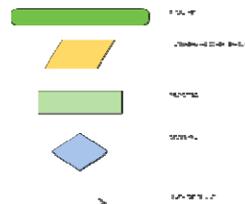
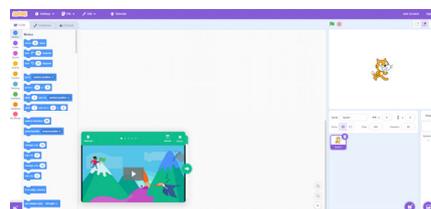
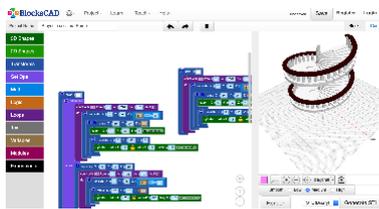
<i>Primer Ciclo</i>	<i>Segundo Ciclo</i>	<i>Tercer ciclo</i>
4.1. Describir rutinas y actividades sencillas de la vida cotidiana que se realicen paso a paso, utilizando principios básicos del pensamiento computacional en situaciones de aprendizaje con el andamiaje adecuado. 4.2. Modificar algoritmos sencillos, así como crear algoritmos en situaciones cercanas y significativas para el alumnado.	4.1. Automatizar situaciones sencillas de la vida cotidiana que se realicen paso a paso o sigan una rutina utilizando principios básicos del pensamiento computacional en situaciones de aprendizaje con el andamiaje adecuado. 4.2. Modificar algoritmos dados de antemano, propios o creados por otros, así como diseñar nuevos algoritmos.	4.1. Modelizar situaciones de la vida cotidiana utilizando principios básicos del pensamiento computacional en situaciones de aprendizaje con el andamiaje adecuado. 4.2. Modificar algoritmos dados de antemano, propios o creados por otros, así como diseñar nuevos algoritmos.
<i>Matemáticas (1º - 3º ESO)</i>	<i>Matemáticas A (4º ESO)</i>	<i>Matemáticas B (4º ESO)</i>
4.1. Reconocer patrones, organizar datos y descomponer un problema en partes más simples facilitando su interpretación computacional. 4.2. Modelizar situaciones y resolver problemas de forma eficaz interpretando y modificando algoritmos.	4.1. Reconocer e investigar patrones, organizar datos y descomponer un problema en partes más simples facilitando su interpretación y su tratamiento computacional. 4.2. Modelizar situaciones y resolver problemas de forma eficaz interpretando, modificando y creando algoritmos sencillos.	4.1. Generalizar patrones y proporcionar una representación computacional de situaciones problematizadas. 4.2. Modelizar situaciones y resolver problemas de forma eficaz interpretando, modificando, generalizando y creando algoritmos.

Introducción

También tiende a confundirse la programación con el pensamiento computacional. El **pensamiento computacional** puede desarrollarse sin necesidad de utilizar ordenadores.

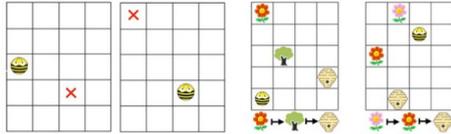
El análisis, adaptación y creación de algoritmos es un ejemplo muy claro de **pensamiento computacional** en matemáticas.

Introducción

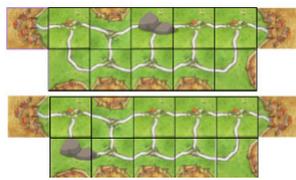




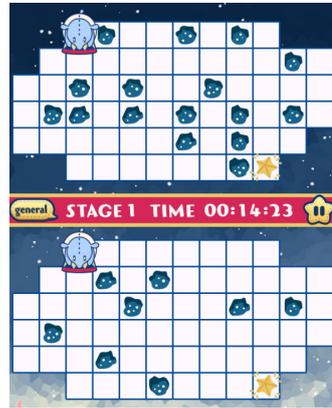
Algunos ejemplos



Diago y Arnau (2017)



TFG Javier Ciria (2023)



<https://www.cokitos.com/juego-de-atencion-dividida/play/>



<https://www.bebas.org/>

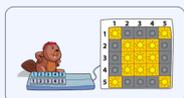
Algunos ejemplos

Bebras es una iniciativa internacional cuyo objetivo es promover la informática (ciencias de la computación o informática) y el pensamiento computacional entre los estudiantes de todas las edades. Los participantes suelen estar supervisados por profesores, que pueden integrar el desafío Bebras en sus actividades docentes. El desafío se lleva a cabo en las escuelas utilizando ordenadores o dispositivos móviles.

Pre-Primary Task Example
Ages 5-8



Primary Task Examples
Ages 8-10



Cadets
Ages 13-14



Seniors Task Examples
Ages 17-18



Benjamins Task Examples
Ages 11-12



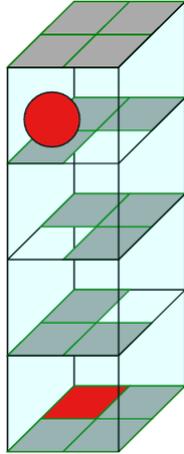
Juniors Task Examples
Ages 15-16





Introducción

Robot esférico (11-12 años)



Si la BeaverBall se mueve a una casilla blanca, baja un nivel. La BeaverBall ignora los comandos que la hacen moverse fuera de los límites.

- A: E, W, N, W, W
- B: E, W, N, E, S, W
- C: E, W, E, N, S, W
- D: E, N, W, S, N, E,



A - The BeaverBall does not reach the bottom level (it cannot move in direction W and so ignores the final two W commands).



B - The BeaverBall reaches the bottom level but stops on a square that is not the GOAL square.



C - The BeaverBall reaches the bottom level but stops on a square that is not the GOAL square.



D - The BeaverBall reaches the GOAL. So, this is the correct answer.



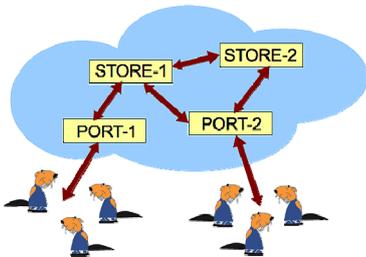
<https://www.bebas.org/task-examples>



Introducción

Nube de castores (13-14 años)

Los castores almacenan sus datos en una nube de cuatro servidores interconectados. Para una mayor seguridad, todos los datos se almacenan tanto en el servidor STORE-1 como en el STORE-2. Para una mayor probabilidad de acceso, se puede acceder a todos los datos a través de los servidores de puerto PORT-1 y PORT-2. Los datos se almacenan únicamente en los servidores de almacenamiento y los servidores de puerto no almacenan ningún dato. Se puede acceder a STORE-1 a través de STORE-2 y viceversa. ¿Qué afirmación es FALSA?



Opciones:

- A) Si STORE-1 y PORT-2 se destruyen, no se podrá acceder a ningún dato de los castores.
- B) Si STORE-2 y STORE-1 se destruyen, se perderán todos los datos de la nube de los castores.
- C) Si PORT-1 y PORT-2 se destruyen, no se podrá acceder a ningún dato de los castores.
- D) Si PORT-2 y PORT-1 se destruyen, se perderán todos los datos de la nube de los castores.

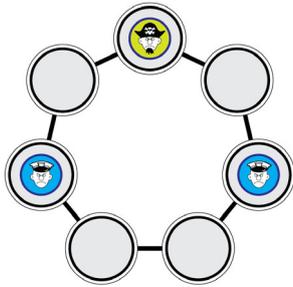
Solución: La respuesta D es falsa. Si PORT-2 y PORT-1 se destruyen, todos los datos de la nube de Castorian quedan inaccesibles, pero no se pierden.

<https://www.bebas.org/task-examples>



Introducción

Piratas (15-16 años)



Jane y Jill juegan al juego de mesa Pirate Hunters. En cada turno, uno de los policías (pero no ambos) se mueve a una casilla vecina. En el siguiente turno, el pirata, que es más rápido, siempre salta dos casillas. Los policías siempre se mueven a un lugar desocupado, no pueden moverse a un lugar ocupado por el pirata o su compañero policía. El juego termina cuando el pirata se ve obligado a saltar sobre uno de los policías lo que ocurriría ahora (imagen), excepto que actualmente es el turno de los policías. Por tanto, para ganar, los policías deben obligar al pirata a colocarse en esta posición cuando sea el turno del pirata.

Jane, es el pirata. Si ayudas a Jill a jugar una partida perfecta, ¿cuántos movimientos hará antes de que la pirata sea capturada?

- A. 2
- B. 3
- C. 5
- D. Jill no podrá ganar

Solución:

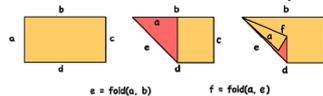
Jill no puede ganar la partida si Jane juega perfectamente.



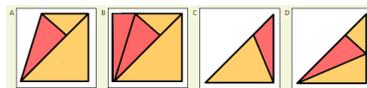
Introducción

Plegado de papel (17-18 años)

Beaver ha desarrollado un lenguaje de programación para el plegado de papel. Este lenguaje se puede utilizar para explicar cómo plegar cualquier hoja de papel con lados rectos. Uno de los comandos de este lenguaje es «fold». $e = \text{fold}(a, b)$ significa: plegar la hoja de papel de manera que el lado a quede completamente sobre el lado b . De esta forma se crea un nuevo lado, el pliegue. Esta línea se denomina e . Ejemplo:



El papel permanece sobre la mesa durante el plegado y la longitud del lado b es el doble de la longitud del lado a . ¿Cómo queda el rectángulo de papel (a, b, c, d) tras ejecutar estos tres comandos? $e = \text{fold}(c, a)$; $f = \text{fold}(c, d)$; $g = \text{fold}(a, f)$



Solución: La respuesta correcta es la A.

Algoritmos. Suma, resta, multiplicación y división

Algoritmos de las operaciones

Algoritmos de la suma <ul style="list-style-type: none">- Tradicional- Suma sin llevadas	Algoritmos de la resta <ul style="list-style-type: none">- Tradicional- Anglosajón- Alumna zaragozana- Resta sin llevadas
Algoritmos de la multiplicación <ul style="list-style-type: none">- Algoritmo escrito usual- Algoritmo extendido- Duplicación- Campesino ruso	Algoritmos de la división <ul style="list-style-type: none">- Algoritmo escrito usual- Duplicación

Algoritmos de las operaciones

Algoritmos de la resta: tradicional vs anglosajón

Aprende

EN LA REGATA PARTICIPAN 43 BARCOS. YA HAN LLEGADO A LA META 15. ¿CUÁNTOS BARCOS FALTAN POR LLEGAR?

Hay que restar $43 - 28$.

1.º Observa las unidades. Como $8 > 3$, añade 10 unidades a 3: $3 + 10 = 13$. Ahora resta $13 - 8 = 5$.

2.º Como has sumado 10 unidades al minuendo, tienes que añadir una decena al sustraendo: $2 + 1 = 3$. Ahora resta $4 - 3 = 1$.

D	U
4	03
-	28
5	

D	U
4	03
-	28
15	

$43 - 28 =$ Faltan por llegar barcos.

Aprende

LA ENTRADA AL ACUARIO CUESTA 22 EUROS. TENGO 15 EUROS. ¿CUÁNTO DINERO ME FALTA?

Hay que restar $22 - 15$.

1.º Observa las unidades. Como $5 > 2$, transforma 1 decena en unidades.

2.º Resta las unidades. $12 - 5 = 7$

3.º Resta las decenas que han quedado. $1 - 1 = 0$

D	U
2	2
-	15
7	

D	U
1	12
-	15
07	

Algoritmos de las operaciones

Algoritmo alumna zaragozana

1º Cuando la cifra del minuendo es menor que la del sustraendo, mentalmente invierten las cifras de modo que “al grande se le quita el pequeño”.

2º Posteriormente, después de la inversión de cifras, el resultado se le resta a 10. Se anota este segundo resultado.

3º Si se ha realizado esta inversión “me llevo una”.

4º Continuación del algoritmo.



<https://colegiomontecanal.es/experiencias-a-pie-de-aula-algoritmo-de-resta-alternativo-desarrollado-por-una-alumna/>

Algoritmos de las operaciones

Algoritmo de la resta sin llevadas

Ejemplo: $347 - 258$

1º Realizamos la siguiente resta: $999 - 258 = 741$

2º Realizamos la siguiente suma: $741 + 347 = 1088$

3º "Quitamos" el 1 y lo añadimos en las unidades $\rightarrow 89$

Tarea: ¿Es cierto para cualquier resta "con llevadas"?

Algoritmos de las operaciones

Algoritmo de la multiplicación tradicional vs extendido

$$195 \times 4$$

	UM	C	D	U
		1	9	5
×				4
<hr/>				
			2	0
		3	6	
+		4		
<hr/>				
		7	8	0

$$195 \times 10$$

	UM	C	D	U
		1	9	5
×			1	0
<hr/>				
			5	0
		9	0	
+	1	0		
<hr/>				
	1	9	5	0

$$195 \times 25$$

		UM	C	D	U
			1	9	5
		×		2	5
<hr/>					
				2	5
			4	5	
+			5		
<hr/>					
			9	7	5
			1	0	
+		1	8		
		2			
<hr/>					
		3	9	0	0
		4	8	7	5

Algoritmos de las operaciones

Algoritmo ¿tradicional? de la división

Japan:
$$\begin{array}{r} 5 \\ 3 \overline{)17} \\ \underline{15} \\ 2 \end{array}$$

Korea:
$$\begin{array}{r} 5 \\ 3 \overline{)17} \\ \underline{15} \\ 2 \end{array}$$

India:
$$\begin{array}{r} 3 \overline{)17} \text{ (5)} \\ \underline{15} \\ 2 \end{array}$$

Brazil:
$$\begin{array}{r} 17 \overline{)3} \\ \underline{-15} \\ 5 \\ 2 \end{array}$$

Argentina:
$$\begin{array}{r} 17 \overline{)3} \\ \underline{-15} \\ 5 \\ 2 \end{array}$$

Israel:
$$\begin{array}{r} 5 \\ 17 \overline{)3} \\ \underline{15} \\ 2 \end{array}$$

Turkey:
$$\begin{array}{r} 17 \overline{)3} \\ \underline{-15} \\ 5 \\ 2 \end{array}$$

Portugal:
$$\begin{array}{r} 15 \\ 3 \overline{)17} \\ \underline{5} \\ 2 \end{array}$$

France:
$$\begin{array}{r} 17 \overline{)3} \\ \underline{15} \\ 5 \\ 2 \end{array}$$

the Netherlands:
$$\begin{array}{r} 3 \overline{)17} \text{ (5)} \\ \underline{15} \\ 2 \end{array}$$

a) 1472 : 5

b) 1402 : 5

Algoritmos de las operaciones

Algoritmo de la multiplicación por duplicación

27 x 14

	27	1
•	54	2
•	108	4
•	216	8
	378	

Algoritmos de las operaciones

Algoritmo de la división por duplicación

405: 14

14	1
28	2
• 56	4
• 112	8
• 224	16
<hr/>	
392	28 Cociente

Cociente 28

RESTO: 13
405 – 392

Algoritmos de las operaciones

Algoritmo de la multiplicación por doble y mitad ("campesino ruso")

27x 14

27	14
54	7
108	3
216	1
<hr/>	
378	

Máximo común divisor. Algoritmo de Euclides con Scratch

Algoritmos de Euclides

Sea M el máximo común divisor de A y B . Como $A=B \cdot C+R$ y M divide a A y a B , entonces divide también a R .

Calculemos el MCD de 150 y 90:

Comenzamos dividiendo 150 entre 90. Una vez hecho esto, se puede observar que el cociente es 1 y el resto 60.

$$150 = 1 \cdot 90 + 60$$

Por tanto, podemos escribir:

$$90 = 1 \cdot 60 + 30$$

Dividimos ahora, por tanto, 90 entre 60. Una vez hecho esto se puede observar que el cociente es 1 y el resto 30. Por tanto, podemos escribir:

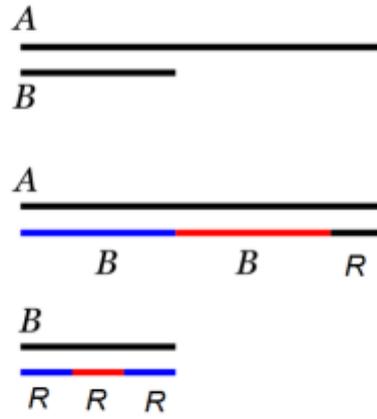
$$60 = 2 \cdot 30 + 0$$

Dividimos ahora, por tanto, 60 entre 30. Una vez hecho esto se puede observar que el cociente es 2 y el resto 0. Por tanto, podemos escribir:

Como el resto ha salido 0, el proceso ha acabado y podemos asegurar que el último divisor es el MCD. Así:

$$MCD(150,90) = MCD(90,60) = MCD(60,30) = 30$$

Algoritmos de Euclides



Algoritmos de Euclides

MCD (60,102)

MCD (25, 125)

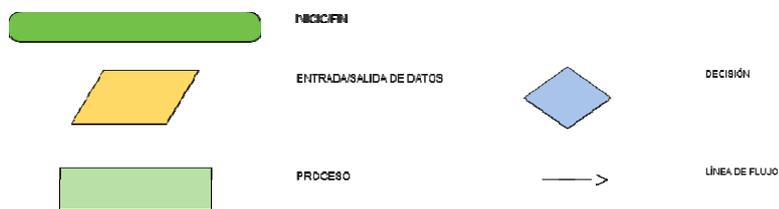
MCD (32, 135)

MCD (36, 36)

Diagramas de flujo

Cuando nos decidimos a programar un algoritmo en un lenguaje de programación (en nuestro caso Scratch), es necesario crear un esquema antes de implementar dicho algoritmo en el ordenador. Una de las maneras de representar dicho esquema es a través de un diagrama de flujo.

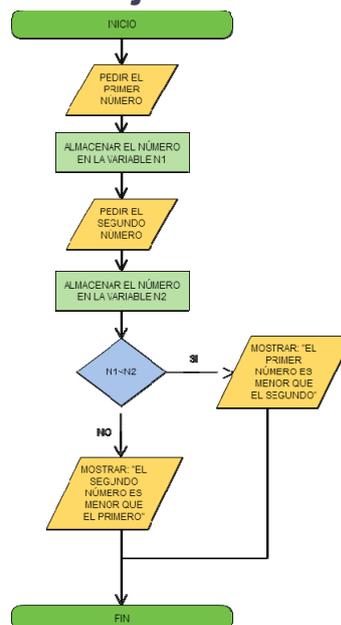
Las representaciones básicas son las siguientes:



Diagramas de flujo

Ejemplo:

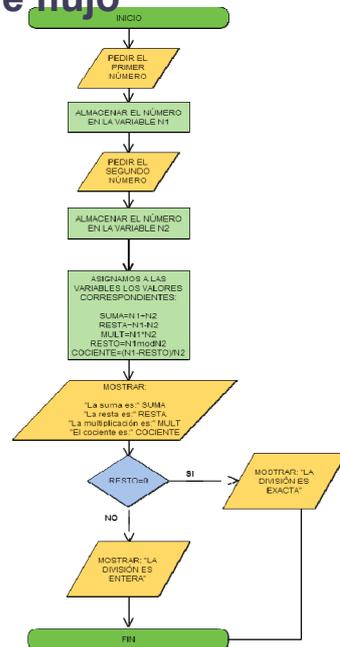
Ordenación de dos números



Diagramas de flujo

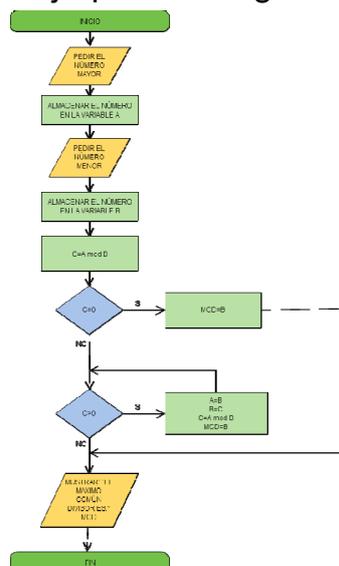
Ejemplo:

Calcular la suma, resta, multiplicación y división de dos números. Además, dicho algoritmo debe decir si la división tiene o no resto 0.



Diagramas de flujo

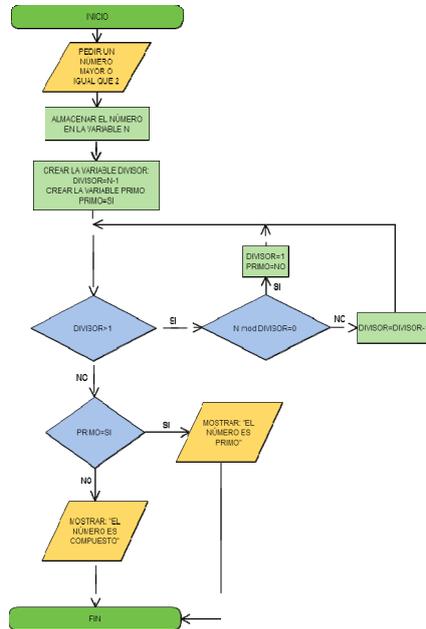
Tarea: Diseño el diagrama de flujo para el "Algoritmo de Euclides".



Diagramas de flujo

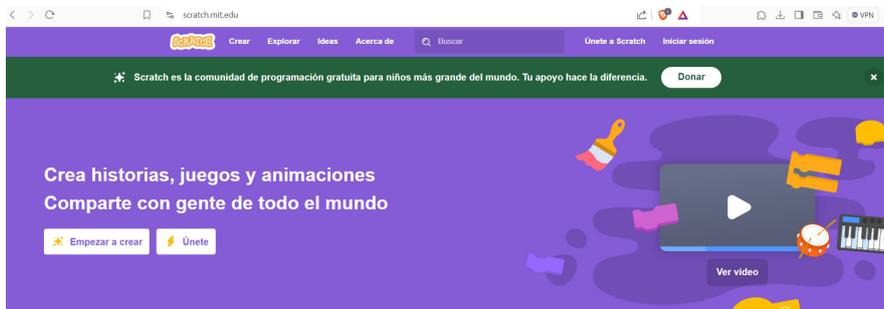
Ejemplo:

Dado un número por pantalla, determinar si el número es primo o no

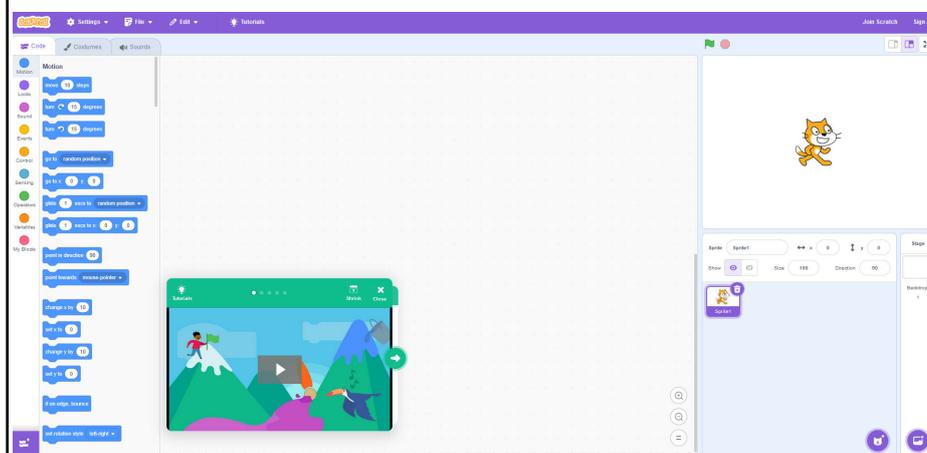


Scratch

<https://scratch.mit.edu/>



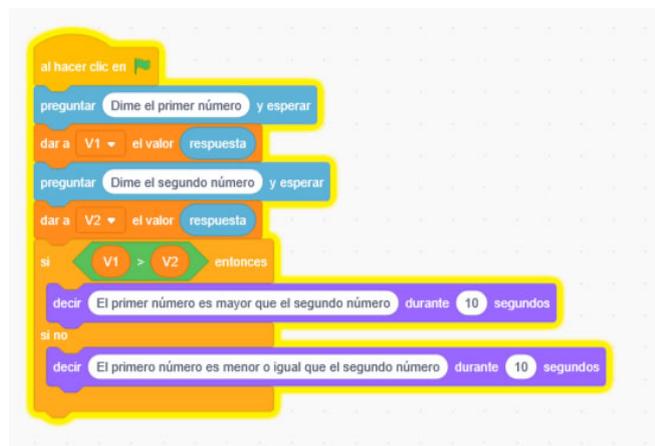
Scratch



<https://scratch.mit.edu/>

Scratch

Tarea: Con Scratch, introduce dos números e indica si el primero es mayor o menor que el segundo.



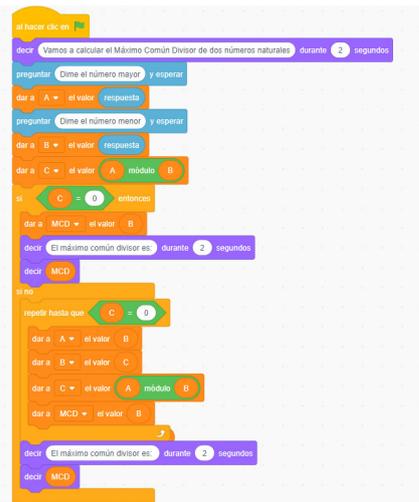
Scratch

Tarea: Con Scratch, introduce dos números y realiza la suma, resta, multiplicación y división entre ellos.



Scratch

Tarea: Con Scratch, introduce dos números y determina el Máximo Común Divisor entre ellos utilizando el Algoritmo de Euclides.



Más información en:

Baeza, M. Á., Claros, F. J., Sánchez-Compañía, M. T., & Arnal-Palacián, M. (2017). Pensamiento matemático avanzado y scratch: El caso del máximo común divisor. *Pensamiento Matemático*, 7(2), 43-64.

Baeza, M. Á., Arnal-Palacián, M., Claros, F. J., & Sánchez-Campaña, M. T. (2018). Comprensión del máximo común divisor. Uno: Revista de didáctica de las matemáticas, 80, 43-47.

BlocksCAD 3D

<https://www.blockscad3d.com>



Gallery

CREATE NOW

Looking to use BlocksCAD in the classroom?

Register with us to learn about our curriculum and school license options

Register

BlocksCAD for Education

A new dashboard for teachers with standards-aligned lesson plans

CLICK HERE >

BlocksCAD & SOLIDWORKS' Apps for Kids!
SolidWorks | Apps For Kids Logo

CLICK HERE >

Upcoming

BlocksCAD 3D

