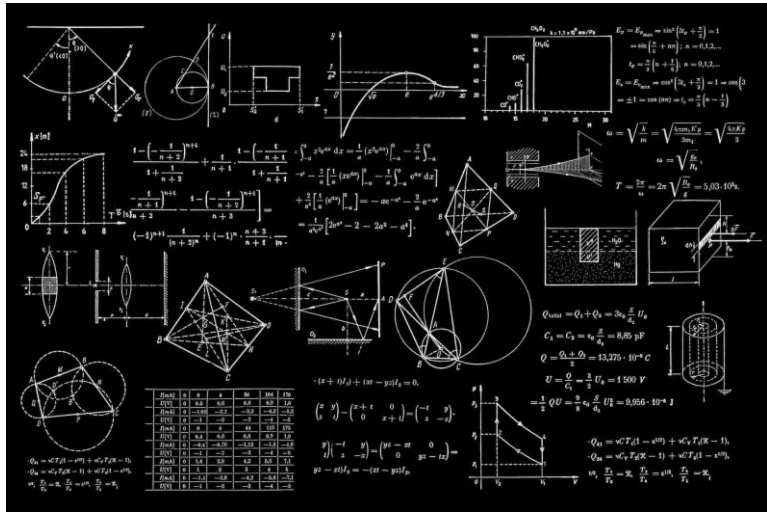
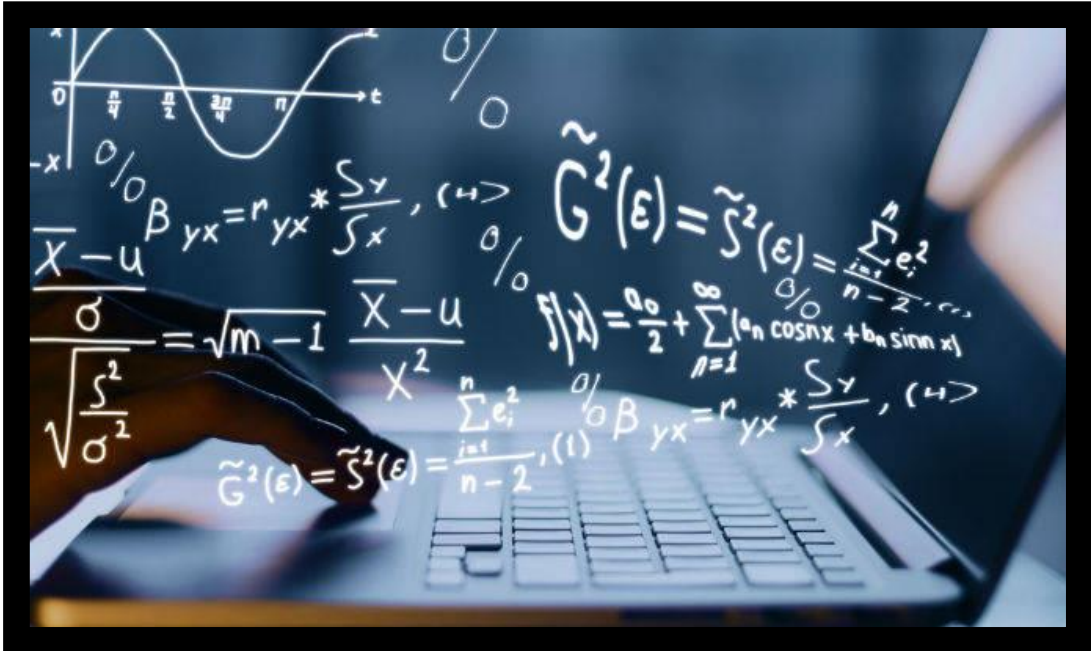


**Formaciones Programa
Código Escuela 4.0**

Matemáticas.



Índice de prácticas



1) Introducción.

2) Álgebra.

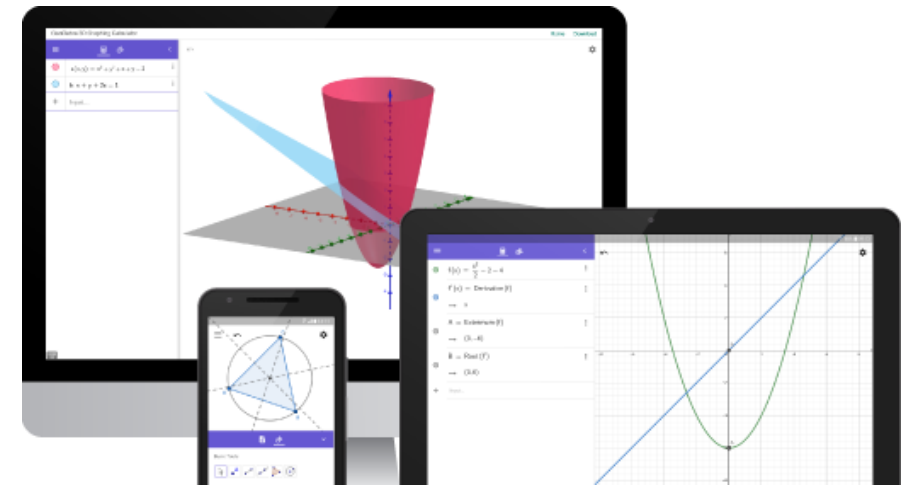
- Álgebra. Raíces cuadradas I.
- Álgebra. Raíces cuadradas II.
- Álgebra. Potencias.
- Álgebra. Representación de funciones I.
- Álgebra. Representación de funciones II.

3) Estadística.

- Estadística. Dado de 6 caras.
- Estadística. Dado de 6 caras. Programa.
- Estadística. Método Monte Carlo. Estadística.
- Método Monte Carlo. Programa

4) Geometría y trigonometría.

- Geometría y trigonometría. Polígonos.
- Geometría y trigonometría. Circunferencia I.
- Geometría y trigonometría. Circunferencia II.
- Geometría y trigonometría. Suma de vectores.



Introducción.



Tanto en la LOMLOE como en el Decreto 65/2022 de 20 de Julio se establece el uso del **pensamiento computacional** como parte de la asignatura de **matemáticas**.

Este documento propone ejemplos para que los alumnos, a través de la creación de programas computacionales, puedan comprender mejor tanto los algoritmos tradicionales para resolver problemas como los conceptos matemáticos que se estudian en clase, fomentando el pensamiento computacional. Los estudiantes deben ser los creadores de estos programas, no solo receptores pasivos del conocimiento.

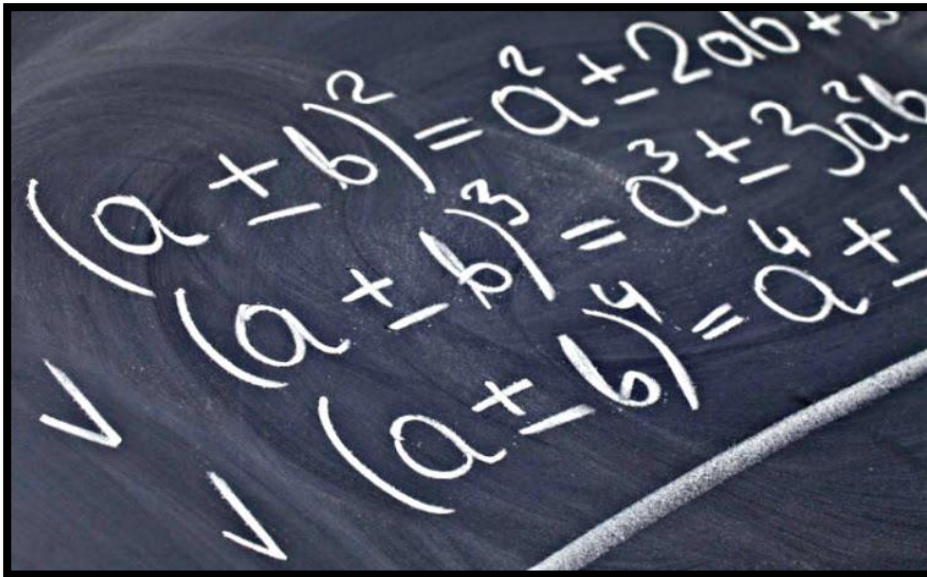
La mayoría de los programas están realizados utilizando programación por bloques, en concreto **MakeCode y Scratch**. MakeCode se utiliza para programar la placa Micro:bit que forma parte de la dotación realizada por la CAM a los centros y Scratch es ampliamente usado en educación desde hace mucho tiempo.

Se han priorizado lenguajes de programación que se suelen usar en las asignaturas del departamento de Tecnología, ya que no se trata de que el alumnado aprenda a programar en clase de matemáticas sino que use la programación como ayuda para la comprensión de las matemáticas. Es fácil trasladar estos programas a los lenguajes de programación (Java, Python,...) que se trabajen en cada centro educativo.

Algebra.



En este caso nos centraremos en programas de algebra sencillos: raíces cuadradas, potencias y representación de funciones lineales. Cómo veremos más adelante la representación de funciones puede ser un poco problemática por como funciona la interfaz gráfica de Scratch.



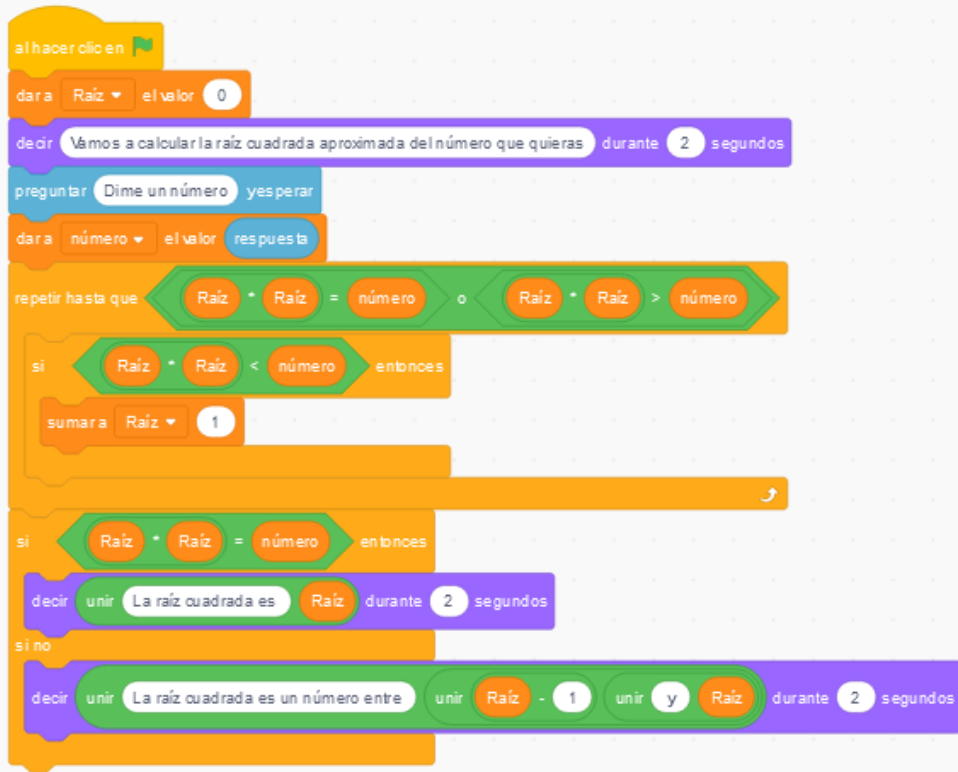
Para realizar estos programas utilizaremos Scratch. Pese a las dificultades mencionadas su amplio uso en educación justifica esta elección. Si en tu centro se trabajan otros lenguajes de programación como Java o Python puedes optar por trabajar con ellos.

El objetivo de estas prácticas es que el alumnado piense en los algoritmos de resolución de problemas que utiliza y los adecúe a como trabajan los ordenadores.

Algebra. Raíces cuadradas I.



Vamos a utilizar la velocidad de computación de los ordenadores modernos para aproximar las raíces cuadradas. Primero aproximaremos a un número entero y posteriormente añadiremos decimales. Usaremos las variables numero y raíz.

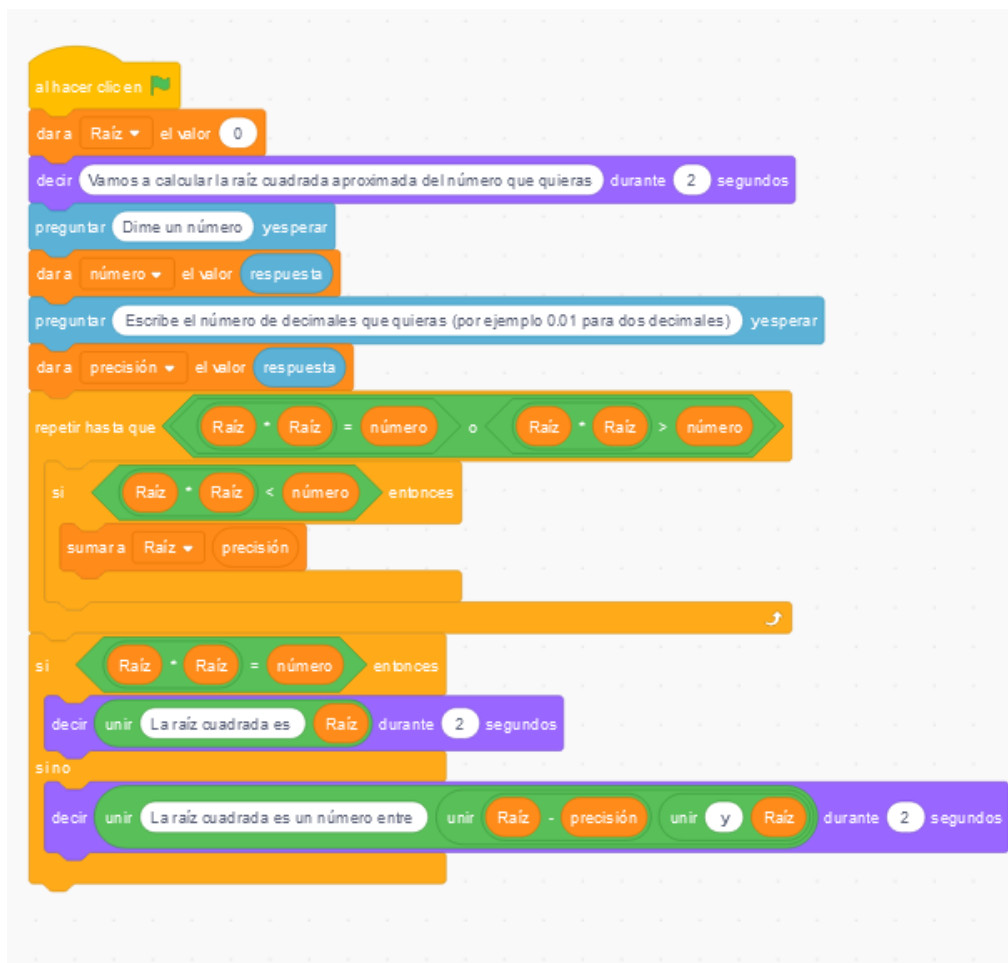


El concepto es muy sencillo. Empezamos dando a la variable raíz el valor 1 y multiplicándola por si misma. Si el resultado es menor que el número del cual queremos hacer la raíz cuadrada sumamos 1 a la variable y pasamos a 2, 3,... hasta que lleguemos a un número mayor o igual al deseado.

Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Algebra. Raíces cuadradas II.



Para poder controlar el número de decimales basta con crear una nueva variable (precisión).

Con esta variable controlamos que valor sumamos al contador (si ponemos 0.1 obtendremos una precisión de una décima, 0.01 centésima y así sucesivamente)

Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Algebra. Potencias.



Calcular las potencias es uno de los programas más sencillos de hacer. Usaremos las variables: base y potencia, contador y resultado.



El programa nos pide el valor de la base y el número al que lo vamos a elevar.

Posteriormente multiplicara la base por si misma. Cada vez que haga está multiplicación sumará uno al contador hasta que este sea igual a la potencia deseada.

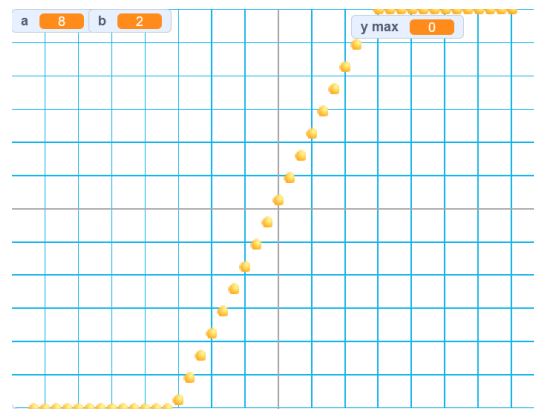
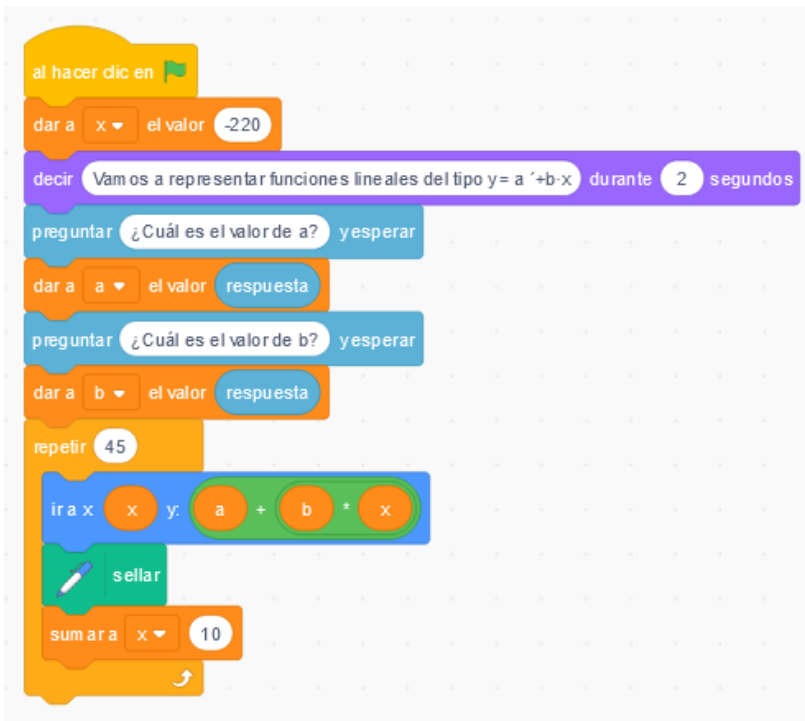
Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Algebra. Representación de funciones I.



Representar funciones con Scratch es sencillo pero conlleva algunos problemas. El principal es que el lienzo está limitado a valores de 240 en el eje x y 170 en el eje y (aproximadamente). Esto hace que sea fácil que muchos de los puntos no se puedan visibilizar correctamente. Aunque podemos prorratear los valores al hacerlo se deformará la pendiente. Para poder dibujar la gráfica necesitaremos cargar la extensión lápiz.



Podemos hacer pequeñas modificaciones al programa para representar todo tipo de funciones.

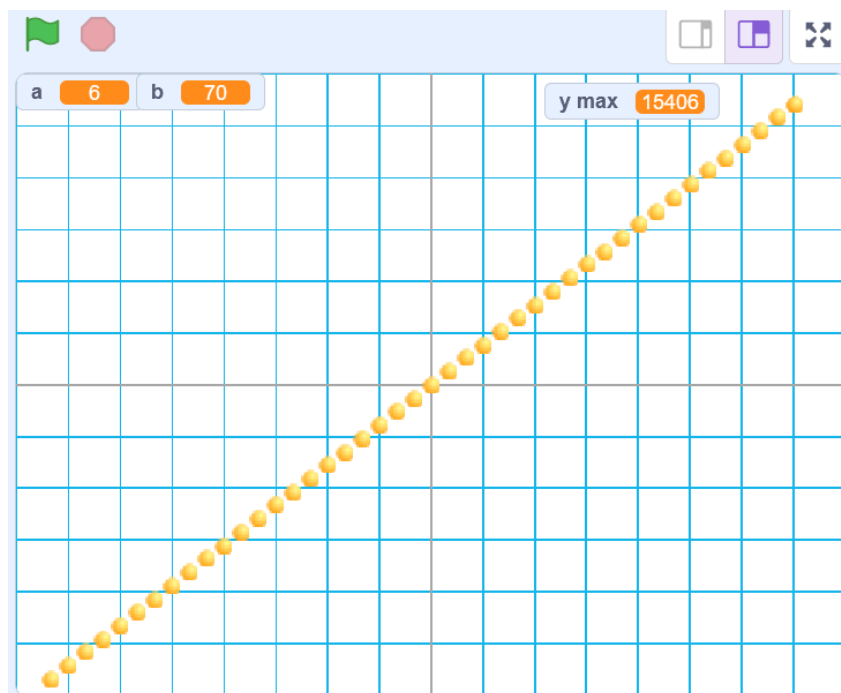
Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Algebra. Representación de funciones II.



Para poder representar todos los puntos de la gráfica hay que crear listas dónde almacenar los valores de x e y . Posteriormente se calcula el valor máximo (en valor absoluto) de Y para luego prorratar los valores de manera que el punto más alto esté en 170. La complejidad del programa aumenta considerablemente, por lo que se han añadido bloques de funciones propias por si se quiere dar al alumnado el archivo con la función de prorratio ya hecha.



Para poder hacernos una idea de la escala se ha dejado visible el valor de Y máxima.

Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Estadística.



Una de las principales dificultades al introducir la estadística es que los estudiantes comprendan que las predicciones probabilísticas **solo se manifiestan de forma consistente con un volumen de datos o un número de repeticiones suficientemente grande.**



En nuestro caso utilizaremos la plataforma MakeCode para realizar dos programaciones:

- Un **dado de 6 caras** en el que podamos simular tiradas y observar como el resultado se acerca al esperado y se vuelve estable al lanzarlo muchas veces.
- Obtener el valor del **número π** usando el método de **Monte Carlo**.

Estadística. Dado de 6 caras.



En este sencillo ejemplo programaremos la placa Micro:bit utilizando el lenguaje de programación **MakeCode** de manera que al pulsar el botón A se genere un número al azar. Si el número es igual a 5 lo almacenaremos en la **variable Veces 5**. Las tiradas se almacenarán en **Nº Tiradas** y en la **variable Estadística** iremos calculando el porcentaje de veces que se obtiene 5.

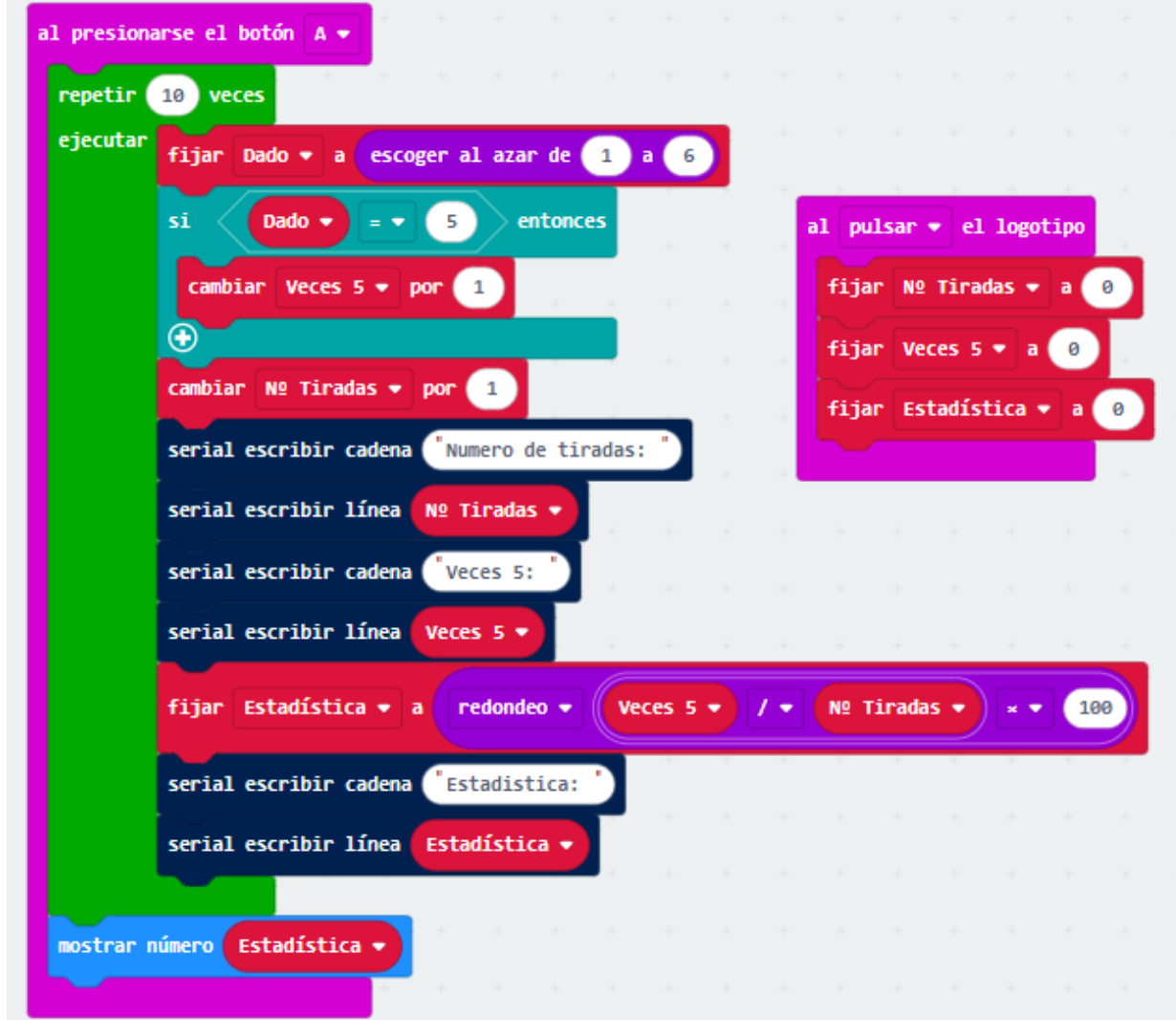


Podemos ejecutar el programa conectando la placa Micro:bit o utilizando el simulador del navegador.

En código se ha incluido la transmisión de datos en serie que permite observar la evolución al incrementar el número de tiradas y descargar un CSV con los datos obtenidos.

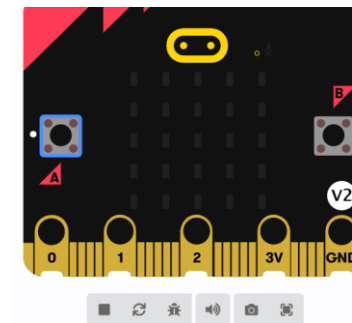


Estadística. Dado de 6 caras. Programa.



En este programa se lanzan 10 tiradas seguidas (Repetir 10 veces). Se puede cambiar fácilmente para hacer que haga 100, 500 o el número que deseemos y acercarnos más rápidamente al valor predicho por las matemáticas.

El código **serial escribir cadena** se utiliza para nombrar la variable de entrada de datos mientras que **serial escribir línea** mostrará el valor de la variable en ese momento.

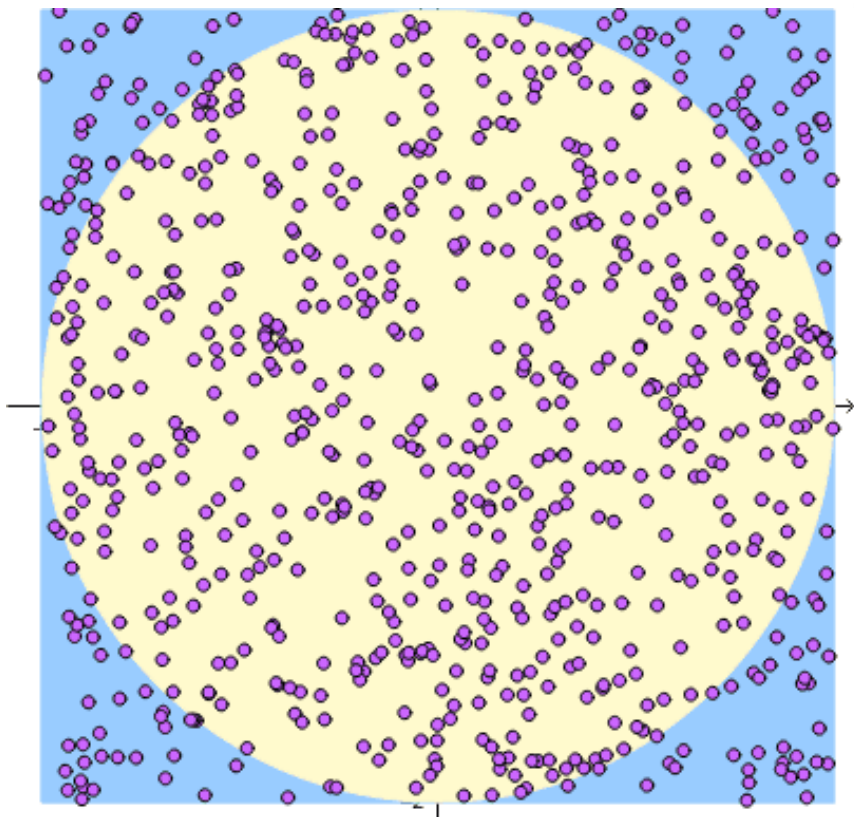


Al presionar por primera vez el botón A nos aparecerá la opción **Mostrar datos simulador**. Si conectamos la placa también tendremos la opción **Mostrar datos dispositivo**.

Estadística. Método Monte Carlo.



El método Monte Carlo consiste en lanzar puntos al azar dentro de un cuadrado de lado L . Dentro del cuadrado hemos inscrito un círculo de radio $L/2$. La probabilidad de que el punto esté dentro del círculo multiplicada por 4 nos dará el valor de π .

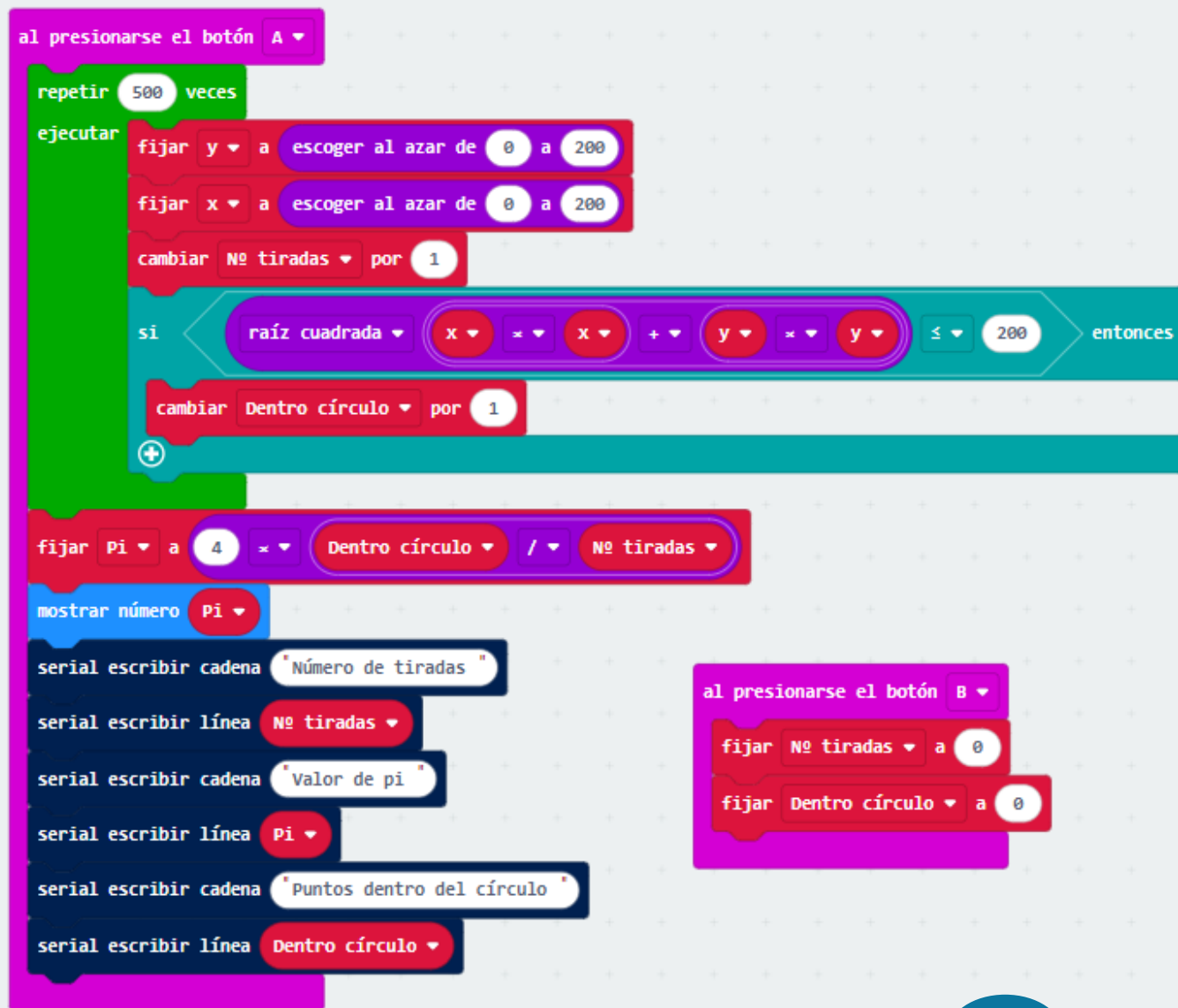


Utilizaremos las variables x , y (coordenadas del punto), N° tiradas y Dentro del círculo. Dado que no podemos generar números aleatorios decimales usaremos un lado de 200.

La Micro:bit viene incorporada con varios sensores, de manera que se puede recoger datos de temperatura, luminosidad,... para posteriormente hacer tratamiento de datos. Puede resultar especialmente interesante para realizar proyectos STEM en colaboración con otros departamentos.



Estadística. Método Monte Carlo. Programa.

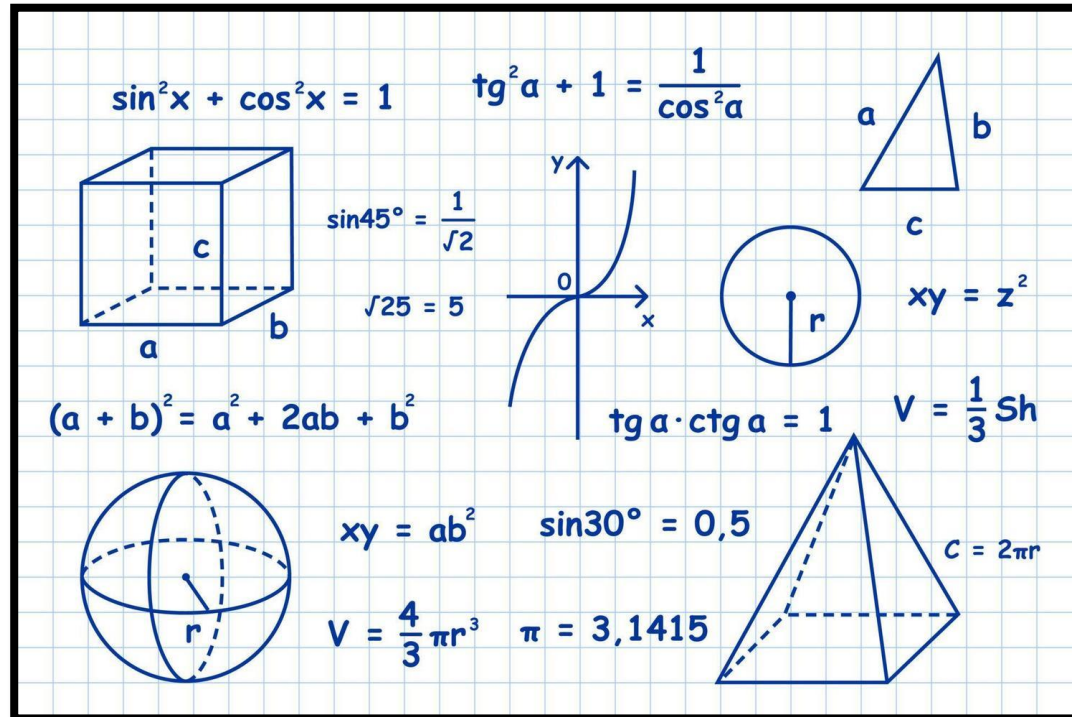


Si el punto está dentro del círculo su módulo debe ser menor o igual al radio.

Geometría y trigonometría.



Utilizaremos la programación para visualizar conceptos de geometría y trigonometría. En primer lugar dibujaremos un polígono regular de lado L y número de lados N. El alumnado podrá comprobar como al aumentar el número de lados el polígono se irá asemejando a una circunferencia. Posteriormente utilizaremos la ecuación paramétrica para dibujar una circunferencia de dos maneras distintas. Por último haremos un programa para sumar vectores.

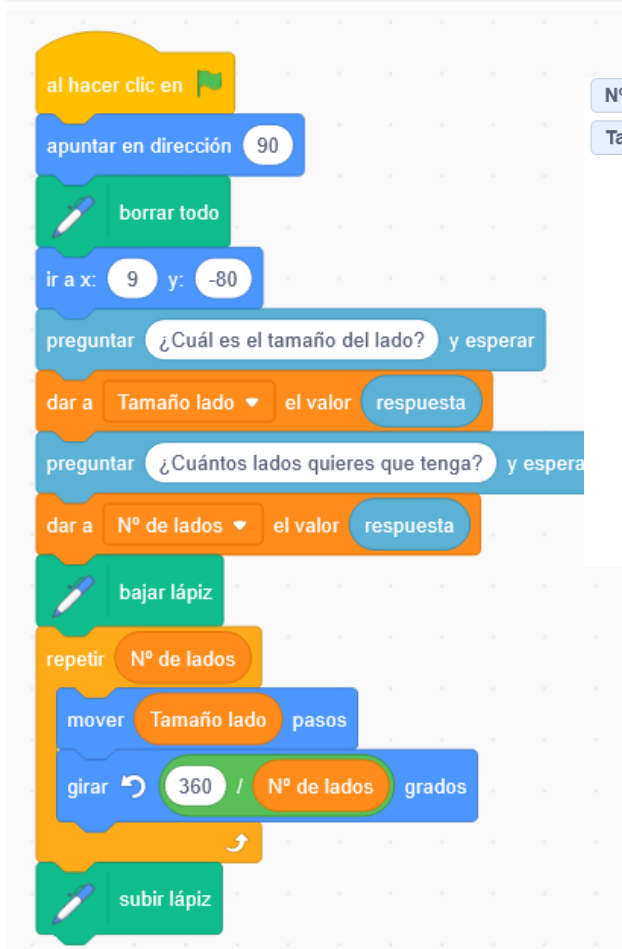


Para realizar estos programas utilizaremos Scratch. Aunque su interfaz gráfica dificultada alguno de los programas su amplio uso en educación justifica esta elección. Si en tu centro se trabajan otros lenguajes de programación como Java o Python puede ser conveniente trabajar con ellos.

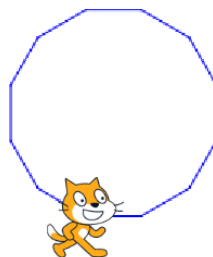
Geometría y trigonometría. Polígonos.



Dibujar polígonos con Scratch es muy sencillo. Primero creamos las variables Tamaño lado y N° de lados. Posteriormente le decimos que dibuje un lado y gire un ángulo de $360^\circ/\text{N}^\circ$ de lados tantas veces como lados tenga el polígono. Ten en cuenta que para dibujar **necesitarás cargar la extensión lápiz**.



N° de lados 12
Tamaño lado 34



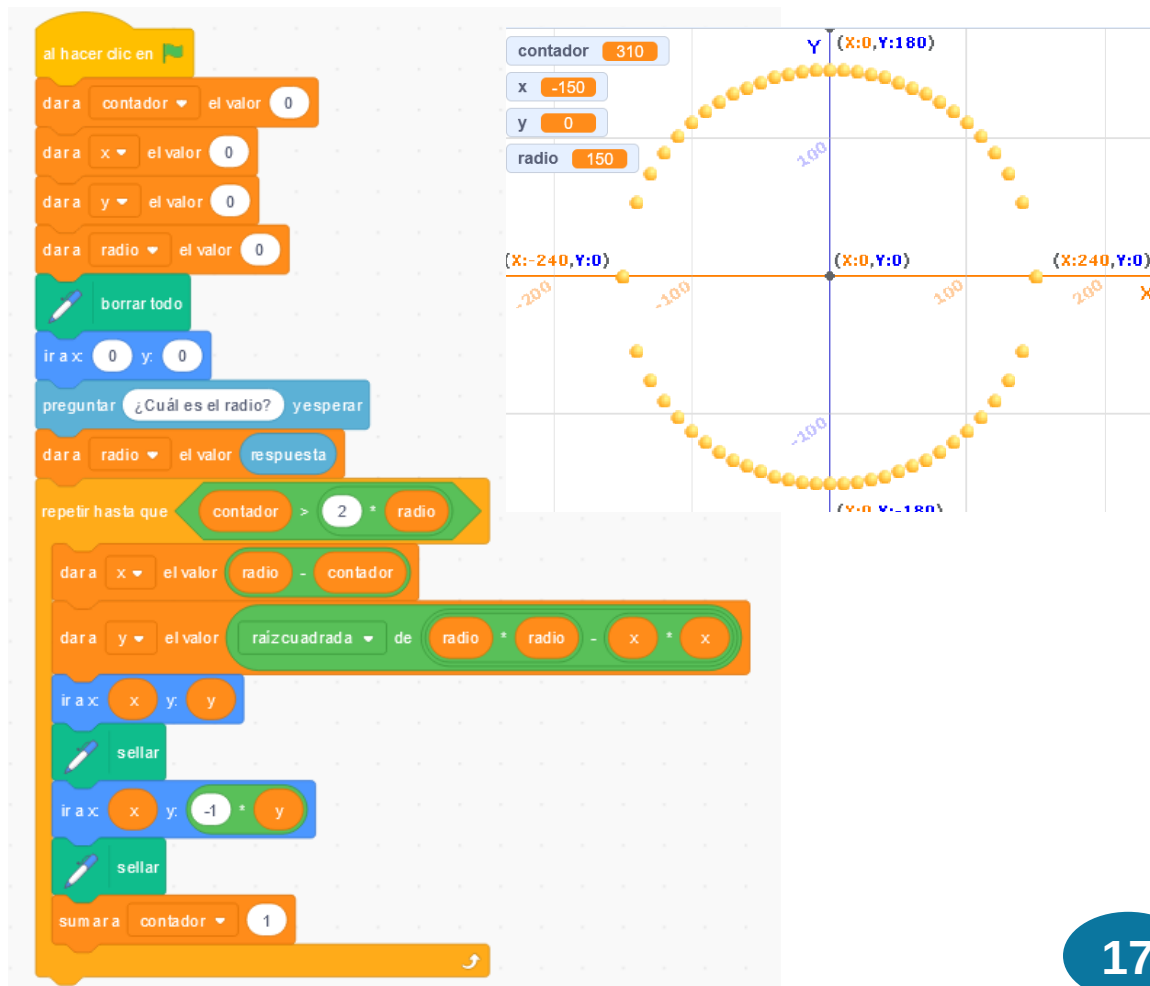
También se puede hacer el programa para que dibuje varios polígonos con distintos colores, tamaños,... hay que tener en cuenta que si el lado es muy grande se saldrá del lienzo de Scratch y no podrá dibujar el polígono.



Geometría y trigonometría. Circunferencia I.



En cualquier punto de la circunferencia se debe cumplir que $R^2 = X^2 + Y^2$. Despejando la Y obtenemos la ecuación que nos permitirá dibujarla. Necesitaremos crear las variables radio, contador, x e y . Para dibujar **necesitarás cargar la extensión lápiz**.



Podemos cambiar el número de puntos que dibujan el círculo sumando a contador un número mayor, por ejemplo si sumamos 10 en lugar de 1 obtendríamos una imagen como la que tienes a la izquierda.

Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.

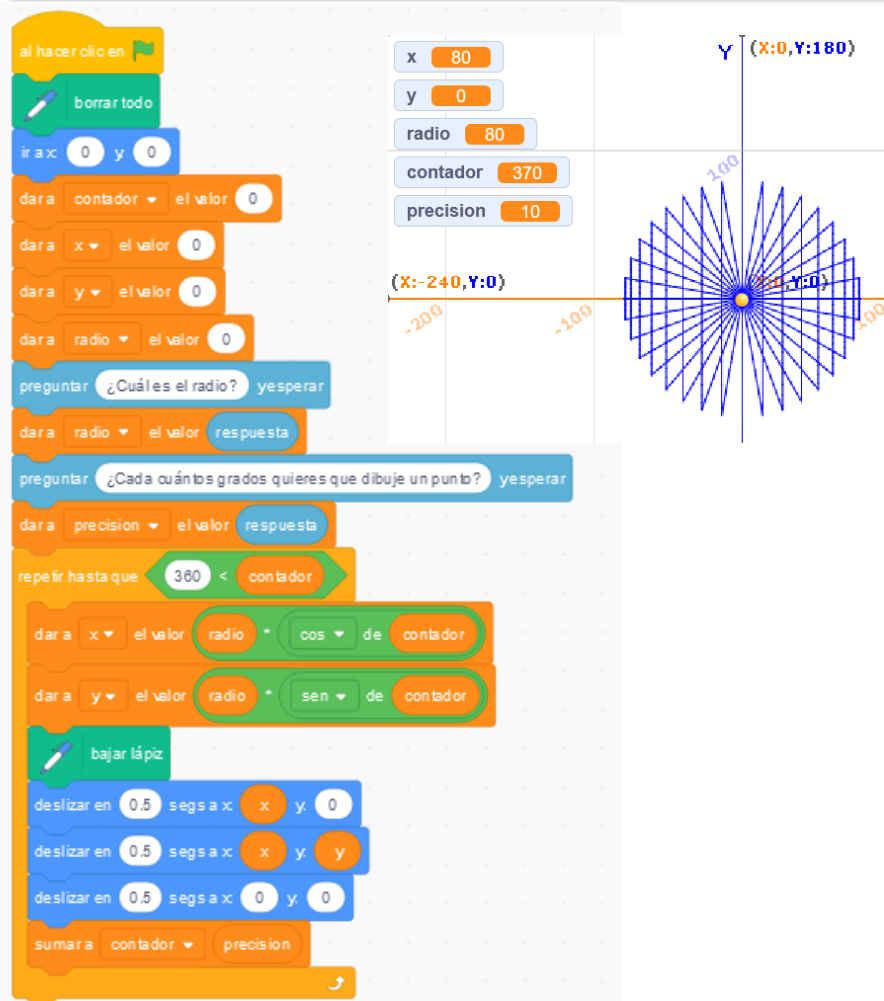


[Enlace al programa Programa en html.](#)

Geometría y trigonometría. Circunferencia II.



Ahora vamos a dibujar la circunferencia utilizando senos y cosenos. Utilizaremos las variables radio, contador, precisión, x e y. El programa irá dibujando triángulos rectángulos cuyos vértices formarán la circunferencia.



Para dibujar el contorno se podrían almacenar los valores de x e y en un lista y hacer que pase por todos ellos.

Puedes hacer que dibuje los triángulos más rápidamente cambiando el valor del tiempo en el bloque deslizar en.

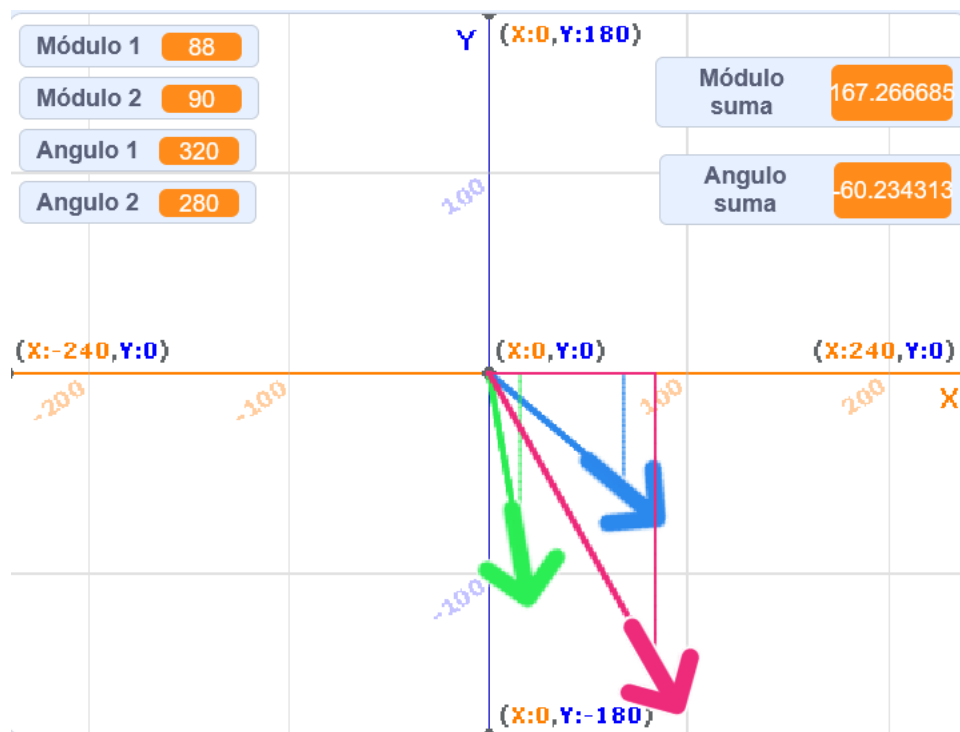
Para probar el programa puedes descargar el archivo HTML y hacer doble clic sobre él para abrirlo en el navegador.



Geometría y trigonometría. Suma de vectores.



Utilizaremos senos y cosenos para descomponer los vectores en sus componentes, sumarlas y después dibujar el vector resultante. Utilizaremos las variables Módulo, ángulo, x e y para cada uno de los vectores (12 variables en total).



Al hacer el arcseno para calcular el ángulo del vector suma no se distinguirán el primer y cuarto cuadrante del segundo y tercero respectivamente. Es necesario utilizar el valor de la X (mayor o menor que cero) para corregir el resultado.

Para dibujar la flecha que indique el sentido del vector debes tener en cuenta que en Scratch la orientación va en la dirección de las agujas del reloj y tiene el 0° en lo que normalmente consideramos 90° , por lo que la orientación deber ser $90^\circ - \text{ángulo}$.





¡Muchas gracias!