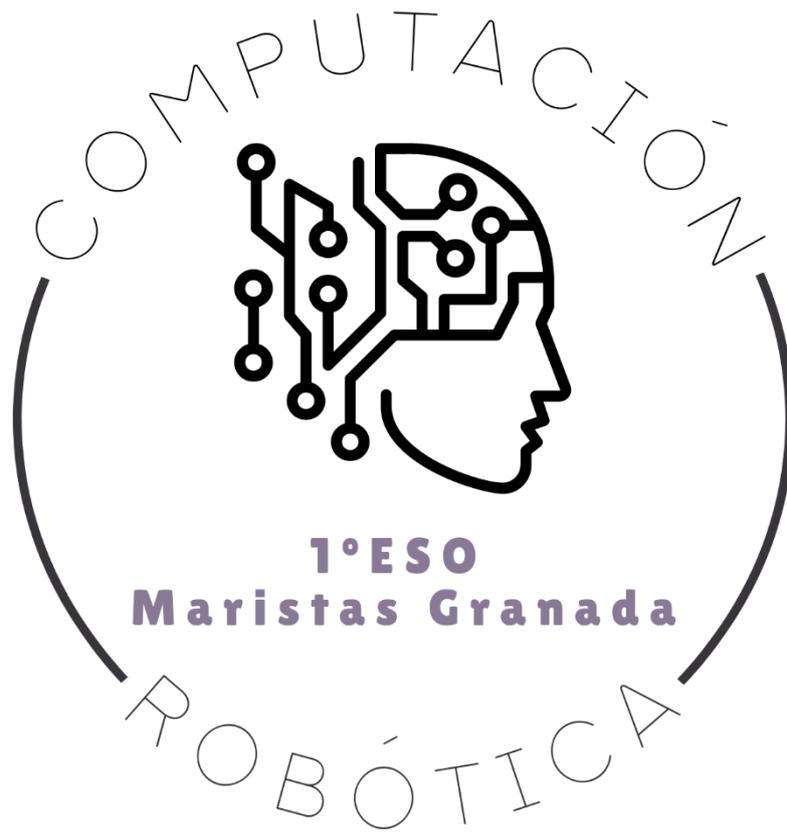


CURSO 2023-2024



RETO 6:
PRIMER CÓDIGO DE
BLOQUES CON SCRATCH

COMPUTACIÓN Y ROBÓTICA 1ºESO

COLEGIO MARISTA LA INMACULADA
CALLE SÓCRATES, 8
18002 - GRANADA

Índice

0. Ubicación en la programación.....	2
1. Descripción del producto final a entregar.....	3
2. Aprendemos practicando.....	4
2.1 Primeros pasos en la programación por bloques: Movimiento, Repetir (Bucles) y Condicional.....	4
MOVIMIENTO: Nivel 1 al 5.....	5
BUCLES (REPETIR): Nivel 6 al 13.....	5
CONDICIONAL SI y SI NO: Nivel 14 al 20.....	5
2.2 Programación en Scratch.....	6
MOVIMIENTO.....	7
BUCLES (REPETIR).....	8
CONDICIONAL SI y SI NO.....	8
SENSORES.....	8
EVENTOS.....	9
3. Por si quieres seguir ampliando en casa.....	10

0. Ubicación en la programación

Título: Reto 6. Primer código de bloques con Scratch

Evaluación: Primera

Temporalidad: 1 sesión (2 horas consecutivas)

Índice de contenidos: Introducción en el entorno de programación por bloques. Interfaz de diferentes espacios de programación. Creación de códigos de programación. Movimiento de objetos por un espacio cartesiano. Manejo de bucles de repetición. Uso de los condicionales. Iniciación en los sensores.

Breve resumen de la situación: Se comenzará a programar bloques en dos entornos diferentes. La primera parte de la sesión será guiada por un pequeño tutorial donde nos iremos familiarizando con este tipo de código y como hay que ir ordenando las órdenes para programar correctamente.

Una vez terminado el tutorial pasaremos al uso de Scratch, que nos abrirá un abanico de posibilidades en la programación. Será una toma de contacto donde aprenderemos como movernos sobre la pantalla, el uso de la herramienta de repetición y los condicionales. Con esto asimilaremos los conceptos básicos de la interfaz de Scratch.

Terminaremos realizando un programa con Scratch, donde se hará uso de todas las herramientas aprendidas.

1. Descripción del producto final a entregar

Como adelantamos anteriormente en la asignatura, vamos a aprender diferentes lenguajes de programación. Es decir, diferentes formas de comunicarnos con los ordenadores. Esta es la primera sesión donde vamos a programar y es importante tener presente todo lo aprendido en temas pasados: algoritmos, diagramas de flujos, eficiencia de algoritmos, etc.

Nuestro código permite al ordenador entender las órdenes a ejecutar y mostrarlas en pantalla. A veces el ordenador no hará lo que deseamos que haga. Pero esto no es culpa del ordenador, sino que nosotros no estaremos desarrollando bien las órdenes a ejecutar. El ordenador lee e interpreta las ordenes que le damos de manera literal. Por eso es fundamental ensamblar bien las diferentes piezas de código.

Compaginaremos explicaciones del profesor con la puesta en práctica por parte del alumno.

Como ya hemos comentado, los problemas reales de la vida no suelen tener solución única. En la programación pasa lo mismo. Se puede llegar a una solución por varios caminos distintos. Sin embargo, habrá planteamientos que serán de mayor o menor complejidad, o que nos permitirán acortar el número de bloques a emplear en el código.

Cada alumno debe completar la ficha de trabajo de la sesión, que en esta ocasión estará subida en Teams en formato editable. Trabajaremos en esta sesión de manera individual, para garantizar que todos los alumnos manipulen Scratch. En algún momento, cuando el profesor lo indique, se podrá consultar con el compañero alguna duda o intercambiar planteamientos.

La ficha de trabajo individual debe entregarse completada, a la finalización de las dos horas de la sesión, por Teams. El formato de entrega de la ficha será en Word, añadiendo impresiones de pantalla donde aparezca claramente el usuario. Además se añadirá el enlace del programa creado.

La calificación final de la ficha de trabajo es la media de la calificación obtenida en cada uno de los retos.

Los retos se califican según los siguientes criterios:

- Están completos todos los retos y bien explicados.
- Cada reto tiene asociada su impresión de pantalla y una explicación.
- Aparece con claridad en la impresión de pantalla el usuario que ha realizado el programa.
- Se puede acceder al programa creado.
- Se ejecuta correctamente el programa final creado.

2. Aprendemos practicando

Vamos a empezar a aprender a programar. Este año nos introduciremos en la programación por bloques. Es el inicio de la programación, ya que, en la programación por bloques, las ordenes que damos al programa nos vienen creadas y definidas previamente. Nosotros, de una manera dinámica, vamos a ir manipulándolas para crear pequeños programas útiles en nuestro día a día o algún minijuego.

2.1 Primeros pasos en la programación por bloques: Movimiento, Repetir (Bucles) y Condicional

Para llegar a estos objetivos vamos a aprender a usar Scratch. Pero antes vamos a empezar con otro programa por bloques que nos dará una visión inicial, para después dar el salto a Scratch.

Lo primero que haremos es crear un usuario en la web code.org. Ten en cuenta los siguientes puntos:

1. Ve al siguiente enlace: https://studio.code.org/users/sign_in.
2. Regístrate usando tu cuenta de @maristasmediterranea.com.
3. Usa tu nombre real como nombre de usuario.

Una vez registrado, veremos el siguiente vídeo que nos ubicará en el entorno de trabajo por bloques:

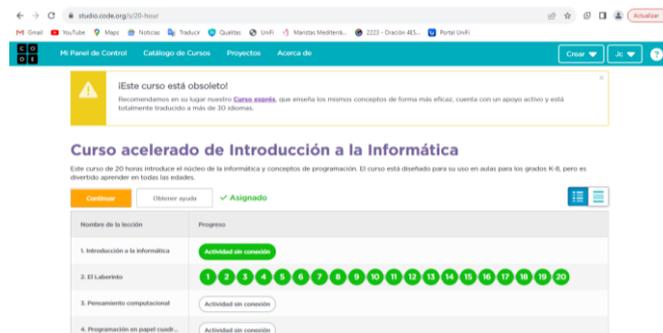
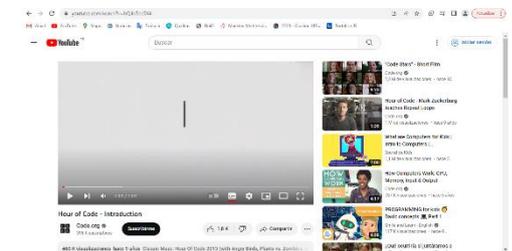
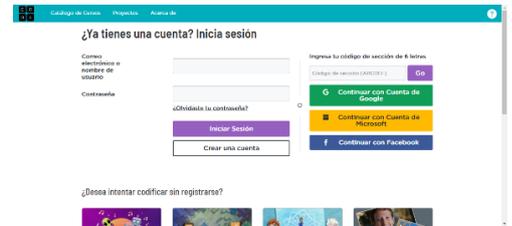
<https://youtu.be/bQilo5ecSX4>

A continuación, vamos a empezar a programar.

El funcionamiento de un programa consiste en dar una serie de comandos que el ordenador sigue de forma secuencial (uno detrás de otro).

Esta es la misma idea que vais a utilizar para resolver los retos que os propone la página code.org.

Para ello vamos a desarrollar la "Lección: 2. The Maze" del siguiente enlace: <https://studio.code.org/s/20-hour>



Iremos explicando lo que vamos a hacer y justo después cada alumno, de manera individual, irá desarrollando los niveles hasta llegar al Nivel 20, que será comprobado por el profesor. Esta validación del profesor formará parte de la nota de este reto.

MOVIMIENTO: Nivel 1 al 5

Tras empezar la lección, vamos a seguir las indicaciones y superando niveles. Lo primero que vamos a aprender es a movernos. Para ello, podemos desplazar los objetos en pantalla usando los bloques **“avanzar”** o **“girar”**. Iremos ensamblando unos bloques con otro como piezas de puzzle, y se irán ejecutando de arriba hacia abajo de manera ordenada.

Avanzar: desplaza el objeto una casilla.

Girar: rotará el objeto en el sentido que digamos, pero sin desplazarse de la casilla en la que se encuentre.



BUCLES (REPETIR): Nivel 6 al 13

Vamos a aprender a usar los Bucles. Hay diferentes tipos, pero para empezar vamos a aprender dos.

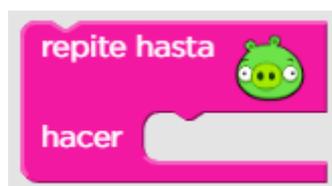
¿Para qué sirve un bucle? Básicamente para recortar líneas de código y ahorrarnos repetir los mismos comandos.

Algunos comandos se repiten muchas veces en los códigos de programación. Para no tener que incluirlos repetidamente se puede utilizar el bloque **“repetir”** (indicando el número de veces que queremos que repita algo) e incluir dentro de él los comandos que deseamos que se repitan.

Todo lo que está dentro del bloque **“repetir”** se repetirá tantas veces como indiquemos. Hasta que no se termine de repetir ese número de veces, el ordenador no continuará leyendo el siguiente código que se encuentre por debajo del bloque repetir.

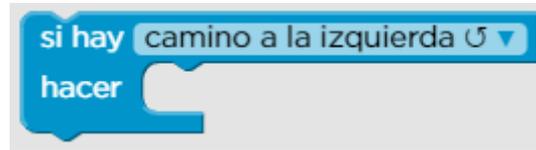


Otro tipo de bucle es el **“repetir hasta...”**. Este bloque nos pone una condición, es decir, nos dice que se va a repetir lo que esté dentro del bucle hasta que ocurra una cosa concreta. En lugar de incluir un número de veces a repetir, este bloque nos permite repetir algo hasta que se cumpla la condición (en la siguiente imagen de ejemplo, el bloque se repite hasta que se alcance el cerdo).



CONDICIONAL SI y SI NO: Nivel 14 al 20

En ocasiones un programa necesita ejecutar unos comandos si sucede una cosa y otros comandos si sucede la contraria. Para esto se utilizan los comandos **“Si hay ... hacer”** o **“Si hay hacer y si no”**. Estos comandos nos darán la opción de que nuestro objeto esté condicionado a hacer una cosa u otra según nos interese. Es decir, le damos una condición y, si se cumple, pasa lo que a continuación este introducido en el bloque.



En el condicional “Si hay ... hacer y si no” le damos una condición al programa. Si se cumple la condición, pasa algo en nuestro programa. Pero si no se cumple, pasa otra cosa. Es el mismo algoritmo con condicionales visto en clase en el Reto 1, donde poníamos un ejemplo para ir al cine o a la piscina según si llovía o no.

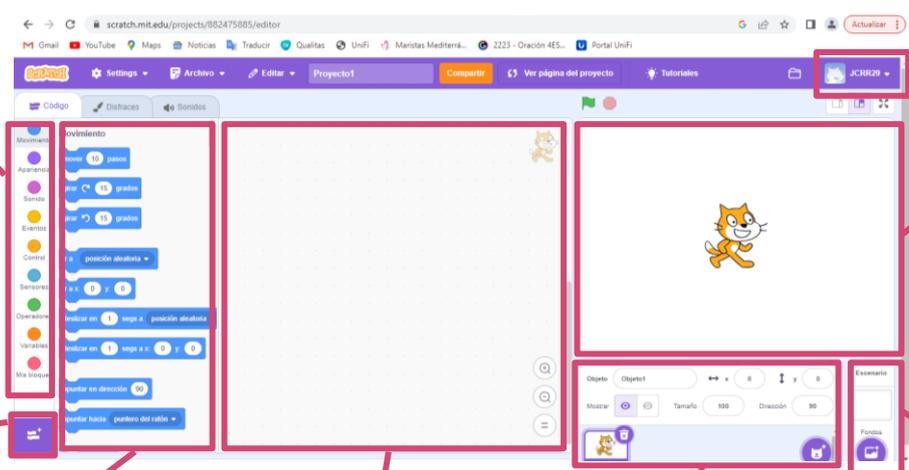


2.2 Programación en Scratch

Una vez terminada la toma de contacto con los bloques vamos a trabajar con **Scratch**. Es una aplicación web gratuita de programación por bloques (muy similar al usado en el apartado anterior). Este programa se puede descargar y usar offline. Sin embargo, nosotros trabajaremos online desde el siguiente enlace: <https://scratch.mit.edu/> .

Lo primero que haremos será registrarnos con nuestro correo de @maristasmediterranea.com. Usaremos nuestro nombre + apellidos (reales) para cuando, creamos futuros proyectos, poder identificarlos con facilidad.

Pulsaremos en “**crear un proyecto**” y nos aparecerá la siguiente interfaz, con la cual nos vamos a ir familiarizando poco a poco.



Los diferentes tipos de bloques de código que podemos usar.

Botón para añadir algún bloque que no esté en los predefinidos.

Aparecen los bloques del tipo que tenemos seleccionado y que podremos ir arrastrando para ir ensamblándolos.

Aquí unimos los diferentes bloques que hemos seleccionado en la parte de la izquierda y que conforman nuestro código.

Nos da información relativa a los objetos (tamaño, posición, etc.). Aparecen todos los objetos creados. Desde aquí podemos añadir más objetos.

Zona donde aparecen los diferentes fondos de pantalla y desde donde podemos añadir más.

Usuario que crea el programa.

Este es el espacio donde veremos el resultado final que hemos creado. Será la pantalla que ejecutará el código redactado.

MOVIMIENTO

A continuación se nos presenta una tabla con varios bloques de la sección de Movimiento. No están explicados todos los bloques que hay en dicha sección. Solo mostramos los que vamos a usar inicialmente y con más frecuencia. A la izquierda de la tabla nos aparece el bloque tal cual lo encontramos en Scratch y a la derecha una breve explicación de su función.



En Scratch debemos tener en cuenta que el movimiento de los objetos se realiza como si nos movieramos en un espacio de ejes cartesianos (esto lo estudiaréis en profundidad en Matemáticas).

En el siguiente enlace puedes hacerte una idea sobre los ejes cartesianos, aplicado a la programación:

<http://www.matematica7.com/plano-cartesiano.html>

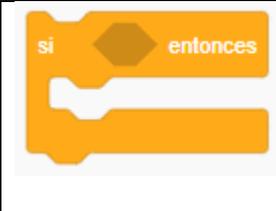


	<p>Con este bloque conseguiremos que el objeto seleccionado vaya al valor de "x" (coordenada horizontal) y de "y" (coordenada vertical) que escribamos. Con esto podremos mover el objeto al punto exacto que decidamos.</p>
	<p>Con estos dos bloques conseguiremos movimientos horizontales para el caso de la "x" y movimientos verticales para el caso de la "y". Es decir, cuando usemos sumar a "x" el valor de 10, lo que lograremos será mover el objeto a la derecha 10 píxeles. Lo mismo pasará con la "y", solamente que el desplazamiento será vertical. El valor de 10 podremos modificarlo a nuestra conveniencia.</p>
	<p>Con estos bloques obtendremos la posibilidad de crear un movimiento de rotación del objeto seleccionado. El valor de 15 nos marca los grados que va a rotar. Este valor se puede modificar a nuestro gusto. Por otro lado, las flechas nos indican el sentido en el cual se va a realizar la rotación.</p>

BUCLES (REPETIR)

	<p>Se repite lo que está dentro de este bloque tantas veces como aparece indicado. En el ejemplo de la imagen es 10, pero puedes modificarlo.</p>
	<p>Se repite lo que está dentro del bloque infinitas veces.</p>
	<p>Se repite lo que está dentro del bloque, pero hasta que se cumple la condición que está dentro del hexágono. En la imagen de la izquierda no aparece la condición (tendremos que añadirla al hexágono añadirla con los bloques de sensores u operaciones).</p>

CONDICIONAL SI y SI NO

	<p>Este bloque de condición se ejecutará solamente si se cumple la condición que nos aparece dentro del hexágono. En la imagen no aparece condición alguna, (tendríamos que añadirla con un sensor u operación).</p>
	<p>Este bloque nos presenta dos posibilidades (dos caminos). Si se cumple la condición del hexágono, se ejecutará lo que esta justo debajo de la condición “si entonces”. Sin embargo, si no se cumple lo que esté dentro del hexágono, entonces se ejecutará inmediatamente lo que esta debajo del “si no”.</p>

SENSORES

Vamos a aprender de momento solo uno de los sensores. Con este bloque podemos introducir condiciones dentro de los hexágonos anteriormente mencionados.

	<p>Al introducir este bloque le estaremos preguntado al objeto: ¿Estás tocando algo que tenga el color que seleccionemos?</p>
--	---

EVENTOS

Con estos bloques conseguiremos iniciar un conjunto de bloques. Un evento será el primer bloque que se usará para iniciar el programa (bandera verde, en la imagen inferior) o bien el que controle qué hacer cuando se pulse una tecla concreta (espacio, en la imagen inferior).

	<p>Siempre usaremos este bloque para iniciar cualquier conjunto de bloques de código. Nos dice que al pulsar el botón de la bandera verde se ejecutará el programa.</p>
	<p>Este bloque nos da la posibilidad de elegir una tecla del teclado. Al pulsar dicha tecla se ejecutará todo lo que este enlazado con este bloque.</p>

3. Por si quieres seguir ampliando en casa

Te proponemos que mejores el programa del laberinto de la ficha de trabajo que tienes subida a Teams, añadiendo dos funciones más:

1. Cuando llegues a la META, debe aparecer un mensaje diciendo: "HAS GANADO".
2. Al pulsar el espacio, se reinicia el programa.

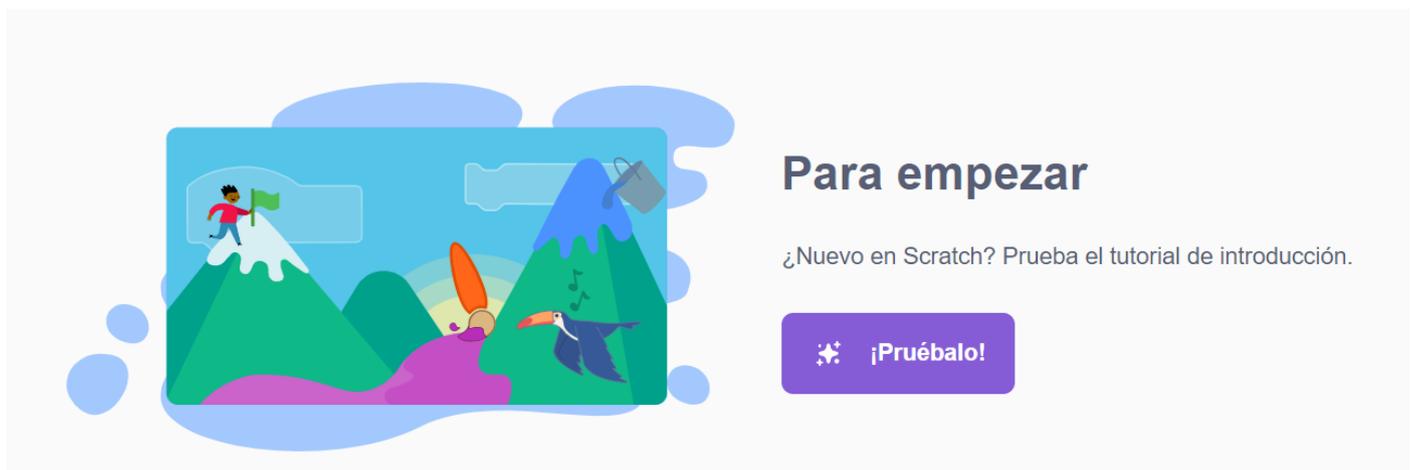
Otra opción para seguir ampliando en casa es desarrollar el siguiente programa:

- Añade dos objetos más al programa. Por ejemplo: un edificio y una flor.
- Si presionas la tecla de la flecha derecha, mover 10 a la derecha. Y si presionas la tecla de la flecha izquierda, mover -10.
- Si el gato toca uno de los objetos, dice durante dos segundos: "Ups qué torpe! He chocado con X". En X ponemos una palabra que identifique con lo que ha chocado. Tras chocar, el gato debe situarse en la posición inicial $x = 0$ e $y = 0$.

Pregunta las dudas que tengas al profesor. Recuerda que tienes Teams para contactar y preguntar dudas concretas. La mejor manera de aprender a programar es "ensayar, equivocarte, preguntar al profesor y aprender de los errores".

El siguiente enlace de la web oficial de Scratch ofrece un tutorial sencillo paso a paso, para principiantes:

<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>



The banner features a colorful illustration of a character climbing a mountain, a hot air balloon, and a toucan bird. To the right, the text reads "Para empezar" followed by "¿Nuevo en Scratch? Prueba el tutorial de introducción." and a purple button with a star icon and the text "¡Pruébalo!".