

mBlock y sus bloques de programación


Introducción

Los robots **Makeblock** se programan con el software propio de la compañía, llamado **mBlock**.

mBlock es un software fácil de aprender y usar, ya que está pensado para niños. Su diseño visual es parecido a un puzzle, donde el usuario juntando bloques (también llamadas piezas) para construir un programa. Y, como en un puzzle, unas encajan entre ellas y otras no, por lo cual es fácil identificar cuando se está haciendo algo bien o mal.

La variedad de bloques disponibles en **mBlock** permiten programar una amplia posibilidad de comportamientos.

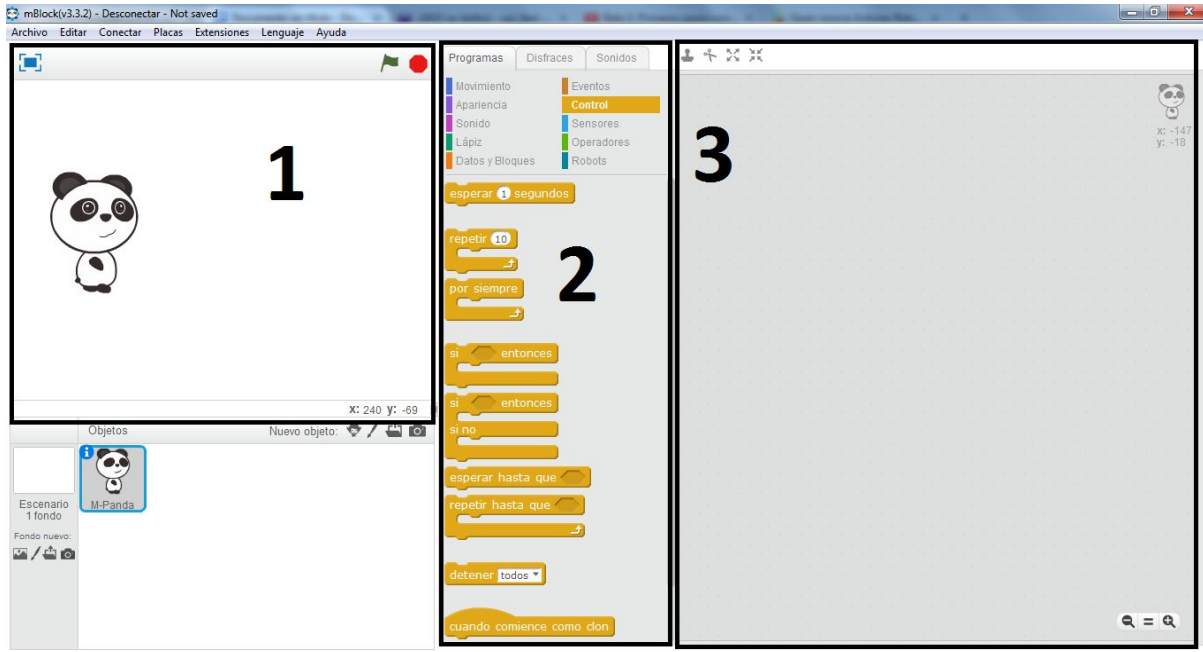
Este tipo de programación se llama **programación visual** (por bloques) y es muy distinta a la **programación por código**, la cual es la típica programación escrita.

	<pre>#include <Arduino.h> #include <Wire.h> #include <SoftwareSerial.h> #include <MeOrion.h> double angle_rad = PI/180.0; double angle_deg = 180.0/PI; MeUltrasonicSensor ultrasonic_3(3); MeDCMotor motor_9(9); MeDCMotor motor_10(10); void setup(){ } void loop(){ if((ultrasonic_3.distanceCm()) < (100)){ motor_9.run(0); motor_10.run(0); }else{ motor_9.run(255); motor_10.run(255); } }</pre>
<p>Programación en bloques</p>	<p>Programación en código (lenguaje Arduino)</p>

mBlock está basado, o es una variante de un famoso software libre llamado **Scratch 2.0**. El cuál está orientado a la educación y es muy conocido en todo el mundo.

Scratch está pensado originalmente para programar **videojuegos**. Mientras que **mBlock** está pensado para programar robots. Aunque en **mBlock** se pueden programar las dos cosas: videojuegos y robots.

La interface que presenta **mBlock** es la siguiente:



Escenario (1)

Bloques(2)

Espacio de construcción de programa(3)

Del lado izquierdo, está “el escenario” (1) del videojuego; en el centro (2) están las categorías de bloques de programación (las piezas con las que se construye el programa del videojuego o del robot); y del lado derecho está el espacio en donde se “arma/construye” el programa, dónde se colocan las piezas/bloques.

Programación de videojuegos

Para programar un videojuego, uno debe seleccionar los bloques del centro (2) y construir el programa en la ventana derecha (3). Para testear y visualizar el videojuego está la ventana del lado izquierdo o “escenario” (1).

De las distintas categorías de bloques disponibles, la única que no se puede utilizar para programar videojuegos es la que dice “**Robots**” (ya que estos bloques están diseñados específicamente para funcionar con robots)

Programación de robots

De igual forma que para programar videojuegos, para programar robots se deben elegir los bloques del centro (2) y construir el programa en la ventana derecha (3).

En este caso, se deben utilizar los bloques de la categoría Robots y también se pueden utilizar las categorías de bloques Eventos, Operadores y Control.

Para testear el programa del robot, no existe una ventana como el “escenario” (1) para los videojuegos. Lo que se tiene que hacer es cargar el programa al robot y testear directamente en él.

LOS BLOQUES

En la ventana del centro (2) se encuentran los bloques de programación, agrupados por sus distintas funcionalidades, donde cada funcionalidad está asociada a un color diferente. Vemos que “movimientos” está asociado al azul, “apariencia” al color lila, sonido al magenta, lápiz al verde mar, bloques y listas al naranja, eventos al marrón, control al naranja melocotón, sensores al azul marino, operadores al verde y robots al azul oscuro.

Estas funcionalidades se pueden dividir en dos grandes grupos.

1. Bloques para videojuegos

Para programar un videojuego se utilizan las siguientes categorías de bloques:

Movimiento, Apariencia, Sonido, Lápiz, Sensores, Eventos, Control, Operadores, Datos y Bloques.

2. Bloques para robot,

Para programar un robot se utilizan las siguientes categorías de bloques:

Robots, Eventos, Control, Operadores, Datos y Bloques.

Tabla de categorías de bloques de programación

Categoría	Notas	Categoría	Notas
Movimiento	Mueve objetos y cambia ángulos	Eventos	Contiene activadores de eventos situado al comienzo de cada grupo de instrucciones
Aparencia	Controla el aspecto visual del objeto, añade bocadillos de habla o pensamiento, cambia el fondo, ampliar o reducir	Control	Contiene los bloques de lógica de programación como los bucles, condiciones, repeticiones..
Sonido	Reproduce ficheros de audio y secuencias programables	Sensores	Los objetos pueden interactuar con el ambiente que ha creado el usuario
Lápiz	Control del ancho, color e intensidad del lápiz	Operadores	Operadores matemáticos, generador aleatorio de números, comparadores..
Datos y Bloques	Creación de variables, bloques y listas	Robots	Bloques para controlar el robot, usar sensores, girar motores...

DESCRIPCIÓN DE CATEGORÍAS DE BLOQUES

A continuación, se describen las diferentes categorías de bloques disponibles en **mBlock**. La mayoría sólo sirven para programar videojuegos, la categoría de **Robots** sólo sirve para programar robots, y hay 3 categorías que comparten los videojuegos y robots (Control, Eventos, Datos y Bloques).

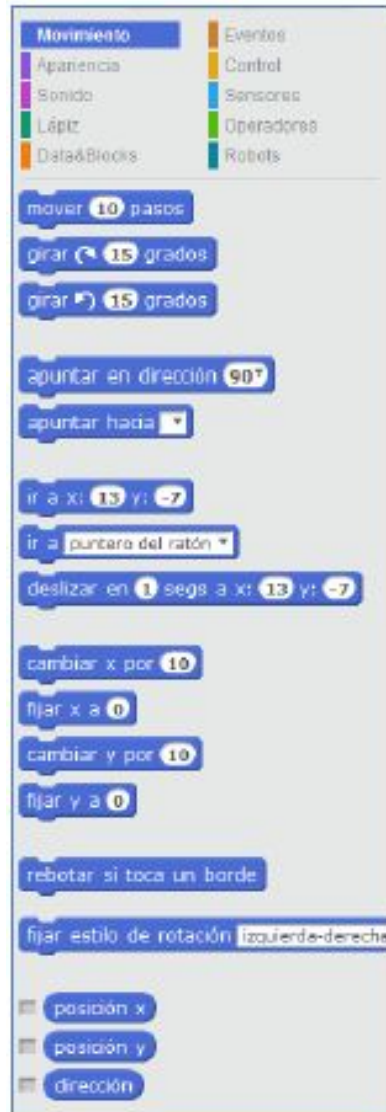
Cuando se describen los bloques para los videojuegos se menciona al Oso Panda, el Oso Panda es el “avatar” que predefine para el videojuego mBlock, o sea el “personaje” que el usuario controla cuando juega (como *Mario* de Mario Bros).

El Oso Panda o “avatar” sólo está disponible cuando se programa un videojuego, cuando se programa un robot, no es posible pre-visualizar el comportamiento del robot en el Oso Panda.

Nota: El Oso Panda es nuestro ejemplo, pero también se puede sustituir por un Gato o cualquier “avatar” que un usuario diseñe.

Bloques de Movimiento:

Con los **bloques azules** podemos hacer que el Oso Panda se desplace hacia delante o hacia atrás la distancia que queramos, que rote sobre el mismo, que cambie de dirección, moverlo a las coordenadas deseadas, etc.



Bloques de apariencia

Con los **bloques lilas** se puede hacer que el fondo del “escenario” (1) cambie de color o que el Oso Panda cambie de color, añadir un nuevo personaje, cambiar partes del cuerpo de los personajes, hacer que hablen y muestren un mensaje, cambiar tamaños, etc.

The image shows a Scratch script for a character named Panda-b. The script is organized into a sequence of actions:

- Appearance:** A purple block labeled "Apariencia" is highlighted in the top-left category menu.
- Script:** The script consists of the following blocks:
 - Two "decir Hello por 2 segundos" blocks.
 - Two "pensar Hmm... por 2 segundos" blocks.
 - "mostrar" and "esconder" blocks.
 - "cambiar disfraz a Panda-b" and "siguiente disfraz" blocks.
 - "cambiar fondo a fondo1" block.
 - "cambiar efecto color por 25" and "establecer efecto color a 0" blocks.
 - "quitar efectos gráficos" block.
 - "cambiar tamaño por 10" and "fijar tamaño a 100 %" blocks.
 - "enviar al frente" and "ir 1 capas hacia atrás" blocks.
 - Three checkboxes: "≠ de disfraz", "nombre de fondo", and "tamaño".

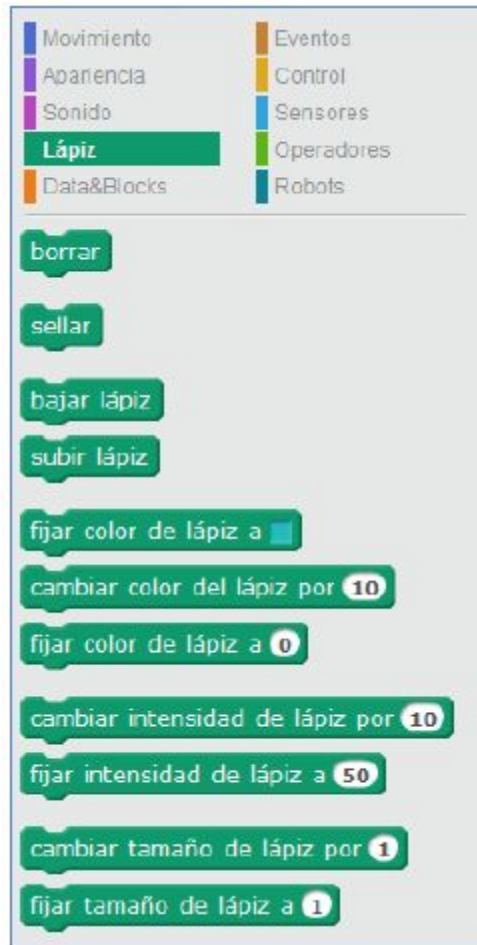
Bloques sonido:

Con los **bloques magentas** se realiza todo tipo de sonidos y se puede controlar el volumen de ellos



Bloques de Lápiz

Con los **bloques verde mar** el “escenario” (1) se convierte en una pantalla interactiva tipo Paint y se puede pintar el escenario como se quiera.



Datos y Bloques

Con los **bloques naranja** se crean bloques de variables, de listas de datos y bloques dentro del programa. Esto se utiliza para realizar programas más complejos y completos:



Variables



Listas



Qué es una variable?

Una variable es un espacio de memoria en nuestro programa que utilizamos cuando queremos almacenar un valor y trabajar con éste durante la ejecución del programa. Explicado con un ejemplo ilustrativo, si queremos hacer un videojuego donde se tenga 3 vidas, podemos crear una nueva variable llamada Vida y hacer que cada vez que pase algo decrezca el valor en 1.

¿qué usos podemos darle a una variable?

- Almacenar el número de preguntas acertadas en un juego de preguntas.
- Almacenar el total de puntos que tenemos en un juego de cartas.
- Almacenar el número de repeticiones que llevamos en una iteración.

Qué es una lista

Una lista es lo mismo que una variable, pero en vez de almacenar sólo un valor, podemos almacenar tantos valores como queramos.

¿qué usos podemos darle a una variable?

- Almacenar las respuestas en un juego de preguntas.
- Almacenar el total de puntos que hemos conseguido en cada uno de los niveles de un juego.
- Almacenar una lista de palabras válidas.

Bloques eventos

Los **bloques marrón** sirven para activar el inicio de cualquier programa que hayamos hecho utilizando *input* externos (como el ratón o las teclas del ordenador) o inputs internos como envío de mensajes del propio programa a otro. Harían la función del botón play en un reproductor de música, cuando lo activamos, empieza todo a funcionar.



Bloque Control

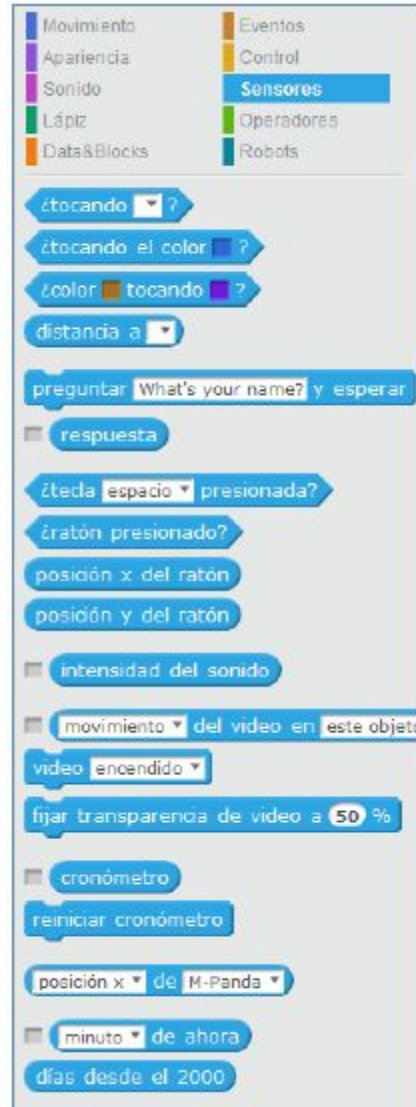
Con los **bloques de melocotón** se realiza la lógica de programación, como los bucles, que una acción se repita varias veces, que espere un tiempo para hacer la siguiente acción o indicar que el programa realice una acción si se cumplen ciertas condiciones y si no se cumplen se realice otra.

Con la lógica de programación definimos las reglas del mundo del videojuego o las del funcionamiento del robot, como por ejemplo si queremos que se pare o gire cuando nuestro robot o el avatar detecte un obstáculo en su camino.



Bloque Sensores

Con los **bloques azul marino** permite detectar si el Oso Panda realiza las siguientes acciones: toca otro Objeto, tocar un color determinado, distancia entre un objeto y el oso. También permite detectar cuando un color determinado toca otro color, la entrada de variables por medio del teclado, si se está presionando alguna tecla del ratón. Verificando todas estas posibles acciones, se puede hacer que el Oso salte, pare, retroceda, que el juego acabe, que sume puntos, que cambie la pantalla...



Bloques Operadores

Con los **bloques verdes** se realizan comparaciones de valores, sumas, restas, contadores, uniones de palabras o de números, etc. Por ejemplo, para definir la condición para que el robot o Oso panda se pare a una cierta distancia de un obstáculo, se tendrá que usar el bloque de comparación “menor que”.

Estos bloques, combinándolos con los de sensores, nos permiten saber si estamos a una distancia menor o mayor de otro objeto, y así poder hacer una o otra acción.



Bloques Robots

Los bloques Robots, de color azul fuerte, es un grupo de bloques exclusivo, creado por Makeblock tanto para las placas propias como la mCore, la Orion como para las placas Arduinos, las cuales son el “cerebro” de los robots Makeblock.

Este grupo de bloques no existe en el programa estándar de Scratch.

A continuación, hacemos una breve descripción de los bloques Robot para el **mBot** (robot de Makeblock). Makeblock amplía y actualiza este grupo a medida de que introduce nuevos sensores en sus robots.

Bloque mBot Program

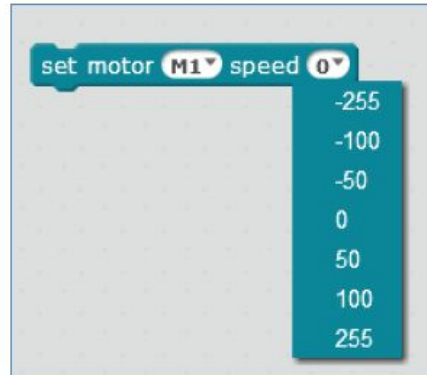
Es básicamente el bloque que envía el mensaje de “Ejecutar el programa cargado en el robot” cuando se enciende. Sustituye al bloque de banderita verde cuando se programa un videojuego. Gracias a este bloque, una vez pasado el programa, el robot puede trabajar sin estar conectado al ordenador via cable o bluetooth.

Para cargar un programa en la placa Arduino, el mBot debe estar previamente conectado al ordenador mediante un cable USB, después se hace clic con el botón derecho sobre el bloque **mBot Program** y se selecciona "Upload to Arduino".

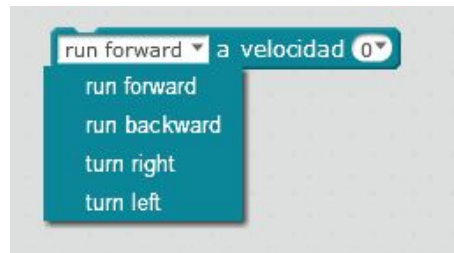


Otros bloques para el robot mBot son:

- 1) **Bloque de motores:** Sirve para controlar los dos motores (M1 y M2) del mBot, de manera independiente y especificar sus velocidades, donde 255 es la máxima velocidad en el sentido horario, 0 es la mínima (quieto) y los números negativos significan que el sentido de giro del motor es antihorario, inverso al giro de cuando el número es positivo.



Aunque también existe otro bloque que permite controlar los dos motores simultáneamente y elegir directamente la dirección y velocidad del robot:

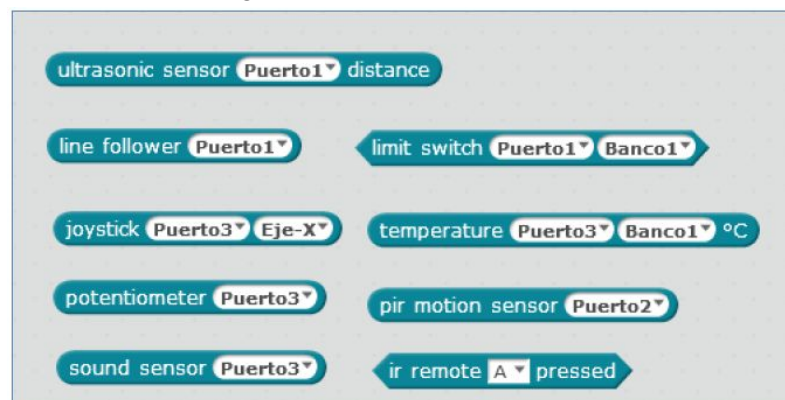


2) **Bloque Led RGB:** Con este bloque podemos controlar los colores de la luz que emite el diodo del mBot:



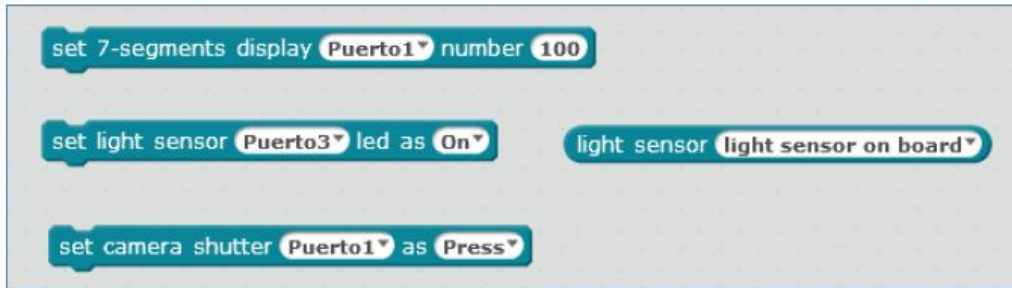
3) **Sensores y control remoto:**

En la imagen siguiente están los bloques para los siguientes sensores: de distancia (ultrasonido), seguidor de línea, de temperatura, de sonido y sensor PIR. También se puede programar el pulsador, un potenciómetro, un joystick e incluso, programar el robot de forma remota con el mando a distancia que viene en el Kit para que según que botón le demos haga una cosa o la otra.



Nota: Es posible se deba esperar unos segundos hasta que el sensor comience a funcionar de manera correcta.

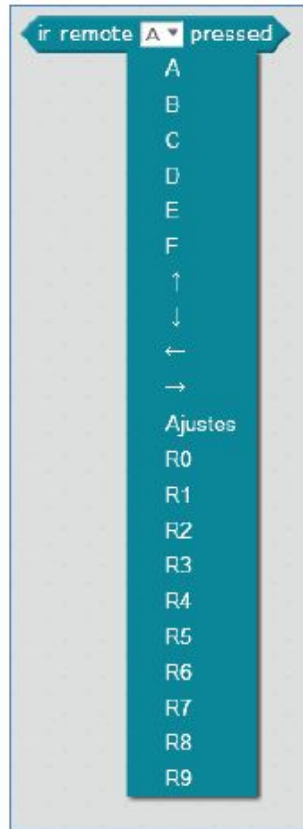
También se puede programar el sensor de luz, el sensor de vídeo y el display 7 segmentos con los siguientes bloques:



Makeblock está incluyendo más bloques a medida que va sacando al mercado nuevos sensores. Desde la versión 3.3.1, actualmente está la versión 3.3.2, y se pueden utilizar los bloques para los sensores de gas, llamas, táctil, humedad, etc.:



De forma remota, se puede controlar el robot, programando a gusto cada una de las teclas del mando a distancia que viene dentro del kit:



4) Bloque de música: Se puede realizar cualquier nota que permita el zumbador

