

Extensiones para objetos bajo mBlock5

Diciembre 2024



Susana Oubiña Falcón

Índice

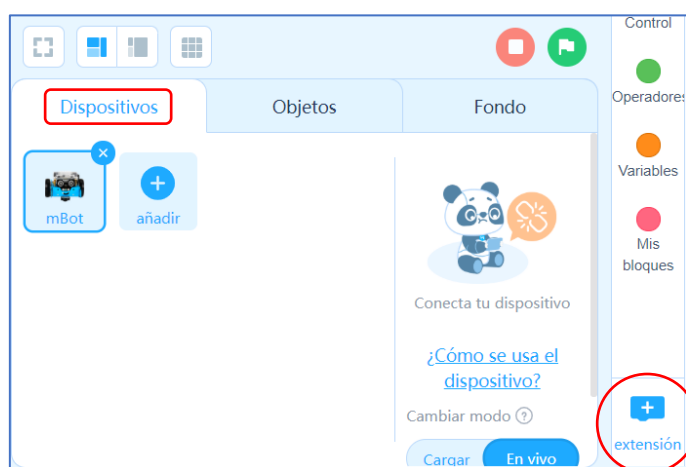
1.	Introducción	2
2.	Extensiones para objetos o sprites	5
2.1.	Arte	5
2.1.1.	Extensión Lápiz.....	5
2.1.2.	Extensión música.....	6
2.2.	Inteligencia Artificial	8
2.2.1.	Extensión Texto a Voz y Traductor.....	9
2.2.2.	Servicios Cognitivos.....	13
2.2.3.	Máquina educable.....	18
2.2.4.	Machine Learning 2.0.....	21
2.3.	Datos de ciencia.....	24
2.3.1.	Hoja de cálculo de Google o Google Sheets.....	25
2.3.2.	Gráfico de datos	36
2.3.3.	Datos de clima.....	49
2.4.	IoT o Internet de las Cosas.....	51
2.4.1.	Video Sensing.....	51
2.4.2.	Mensaje en la nube de usuario.....	53
2.4.3.	Cargar en modo difusión.....	53
	Anexos.....	67

1. Introducción

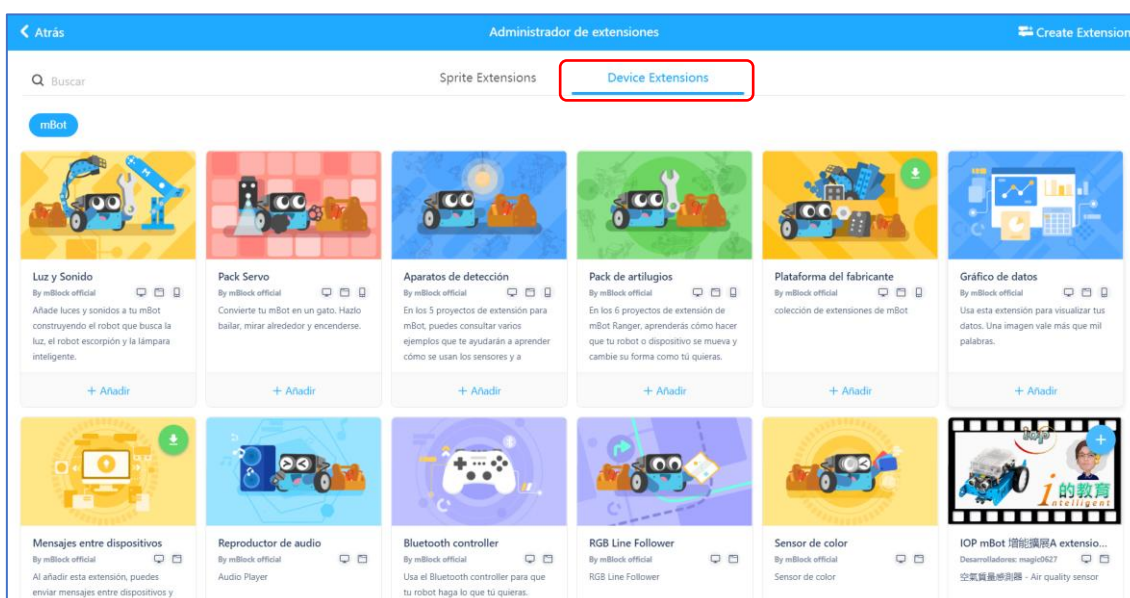
El mundo de las extensiones en *mBlock* siempre ha sido su punto fuerte y ha hecho que destaque entre los diferentes softwares de programación. Este documento se ha trabajado bajo el software mBlock5 de Makeblock y se centrará en sus extensiones.

Una extensión podemos definirla como un conjunto de bloques que han sido diseñados específicamente para programar o controlar de forma sencilla (a través de comandos y no picando código) determinadas acciones sobre el objeto o componentes electrónicos concretos que se conectan a un dispositivo (robot o placa). Por lo tanto, debemos diferenciar entre extensiones para dispositivos y extensiones para objetos y me explico:

- Si escogemos como dispositivo por ejemplo el mBot, podemos acceder a sus extensiones haciendo clic en el botón azul de extensiones inferior derecho del programa, representado con el signo más:

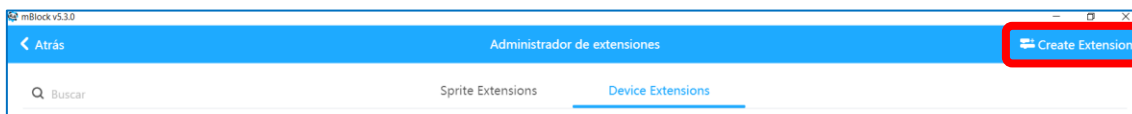


En la versión del software actual, nos encontramos con esta imagen:

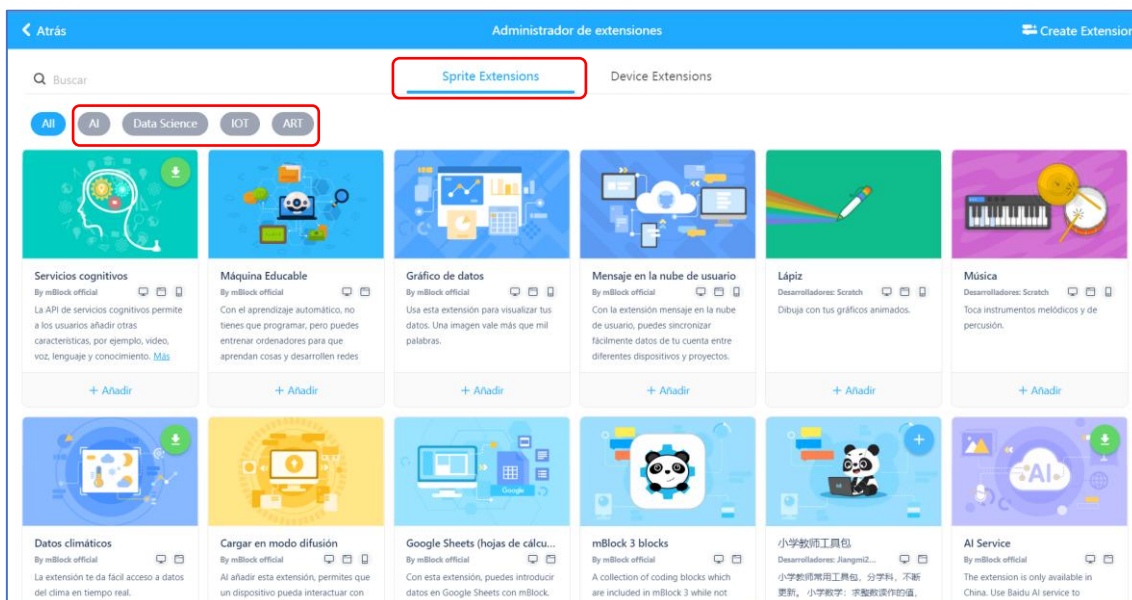
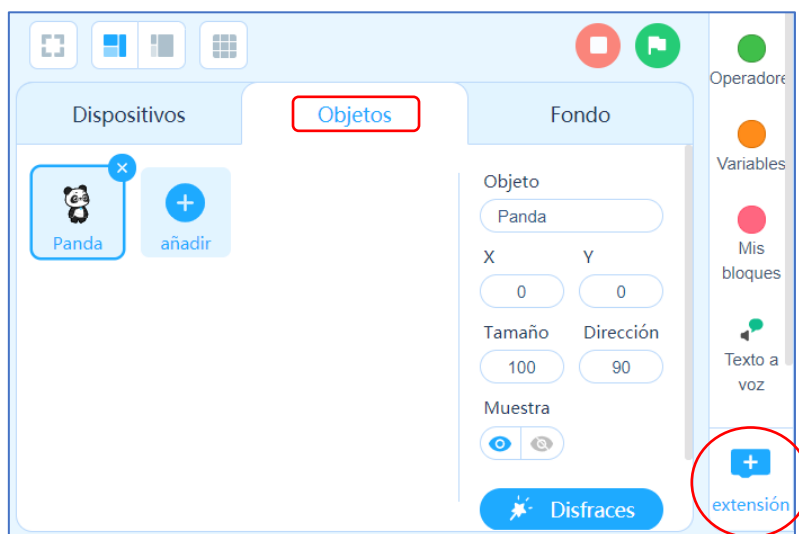


Como puede verse, podemos “Añadir” cualquiera de las extensiones que se han diseñado para el mBot, bien fuera creada por la propia casa Makeblock o por otro usuario registrado. Es decir, nosotros podemos diseñar nuestra propia extensión. ¿Cómo? a través de la pestaña superior

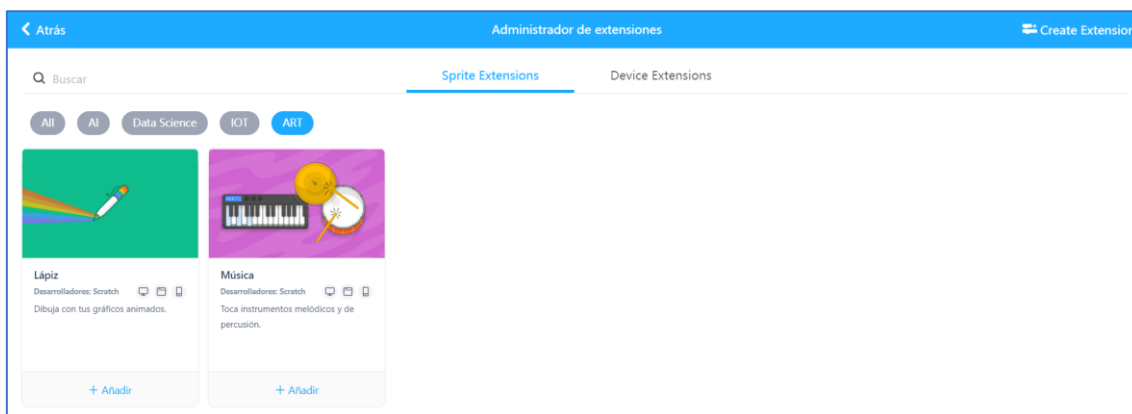
derecha “Create Extension” o entrando directamente en *mBlock Extension Builder*. Sin duda, un curso muy interesante y que no trataremos en este documento.



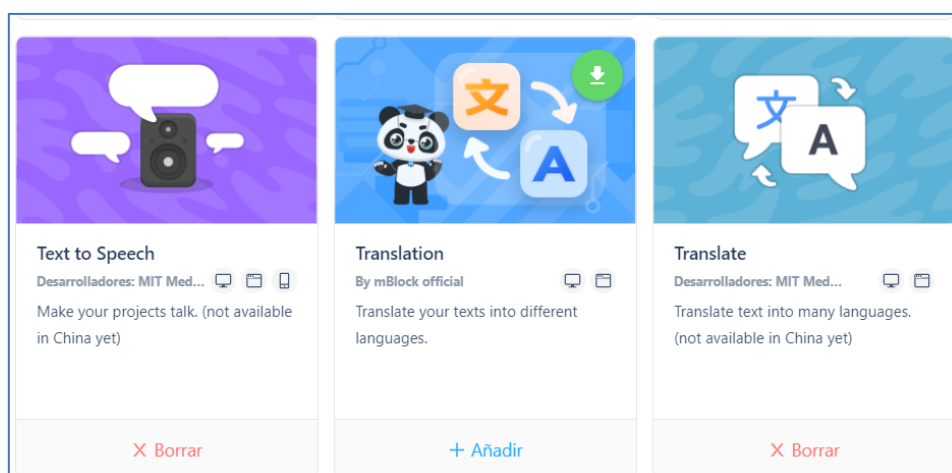
- Si seleccionamos el objeto en lugar de dispositivos y hacemos clic en el botón azul de extensiones, observamos que también podemos incluir en nuestros programas, pero sólo para objetos, extensiones muy interesantes relativas a inteligencia artificial, IoT, ciencia y arte:



En la imagen anterior nos encontramos con la opción de la extensión Lápiz y Música, que en la versión mBlock3 se incluía en los bloques principales pero que, en esta versión se muestra como una extensión a parte, ambas en el subapartado de arte:



Para incluir una extensión en nuestro programa accederemos a la pestaña extensiones y haremos clic en **Añadir** y si queremos borrarla, obviamente, seleccionaremos **Borrar** desde el propio centro de extensiones:



El icono con la flecha verde que se muestra en la imagen anterior nos informa que existen actualizaciones en esa extensión por lo que, antes de Añadirla debemos actualizarla haciendo clic en el círculo verde:



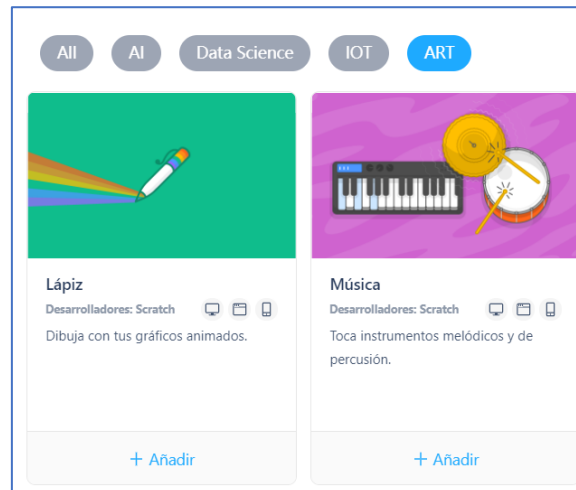
En el punto 2 de este documento se describirán las extensiones para objetos ya que algunas, en mi opinión, son muy aprovechables.

2. Extensiones para objetos o sprites

Veremos las que considero más interesantes de los diferentes bloques.

2.1.Arte

Sin duda, son las más conocidas. Aquí nos encontramos con las extensiones de dibujo (lápiz) y música.

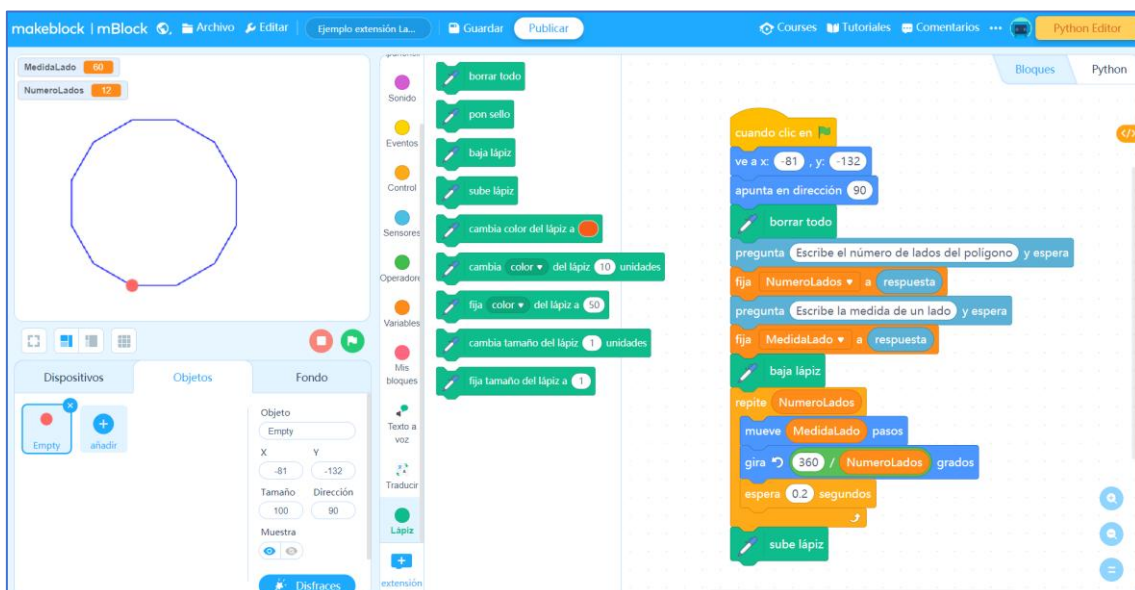


2.1.1. Extensión Lápiz

Con ella se incluyen o implementan comandos para que los objetos puedan dibujar sin dificultad en el escenario. Como ya comenté en la introducción, es una extensión muy conocida y utilizada en su software predecesor.

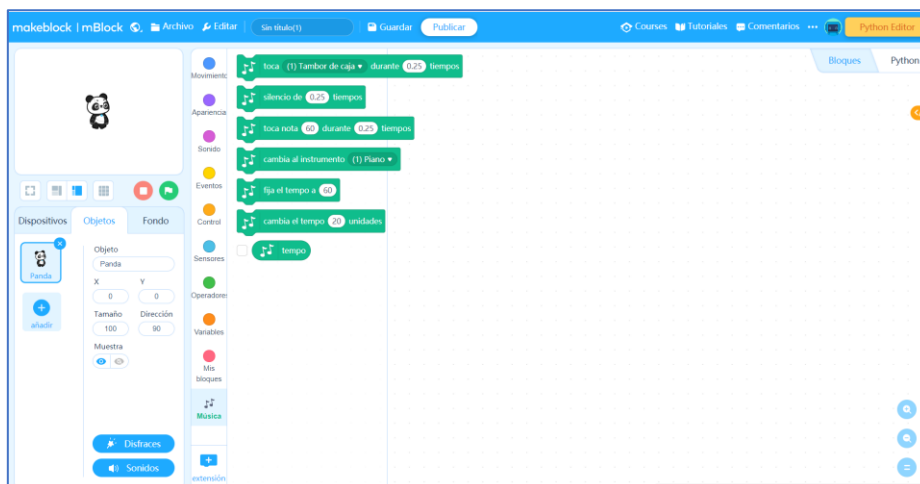
mBlock3 ya disponía de un propio bloque dentro del programa para pintar y que incluso de llamaba “Lápiz”.

En el siguiente ejemplo se ha creado un objeto punto de color rojo y tras decidir cuantos lados debe tener nuestro polígono y la medida de su lado, dibuja el polígono regular deseado. En la siguiente imagen he querido dibujar un dodecágono (12 lados) regular de 60 pixeles de largo en cada lado:

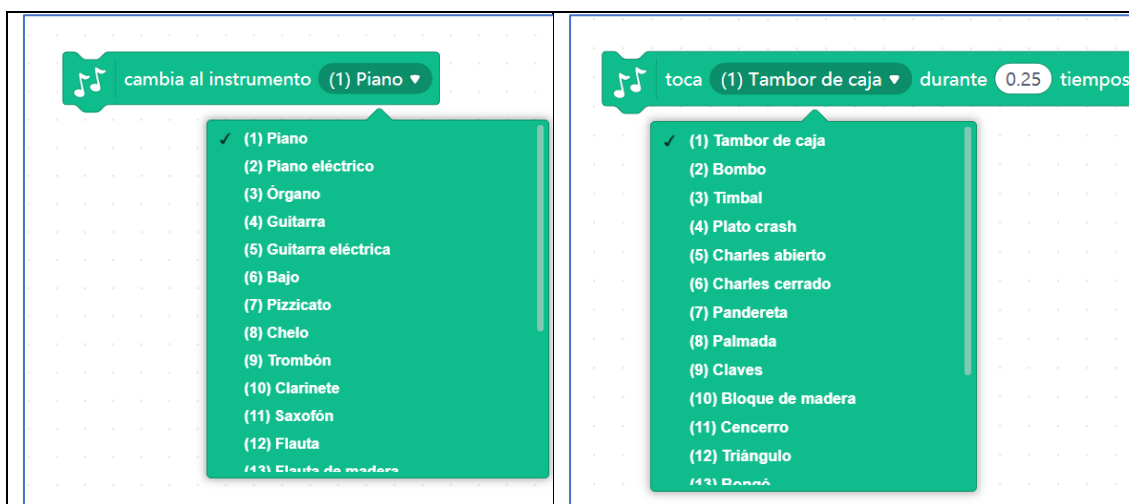


2.1.2. Extensión música

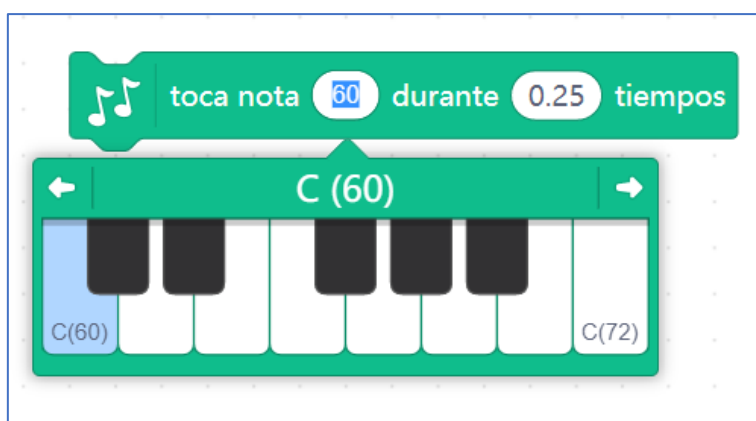
Obviamente, con ella podemos crear música e incluso simularla con diferentes instrumentos:



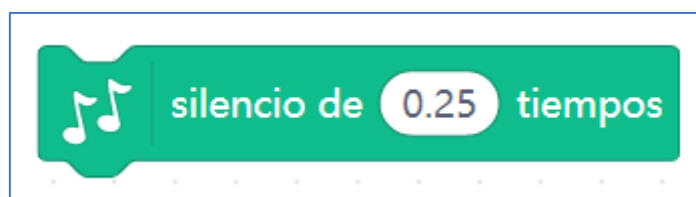
Los siguientes comandos permiten elegir entre múltiples instrumentos, inclusive para la percusión:



En toda extensión de música no puede faltar la inclusión de un comando para reproducir las notas musicales en sus diferentes tiempos (blanca, negra, corchea, etc):

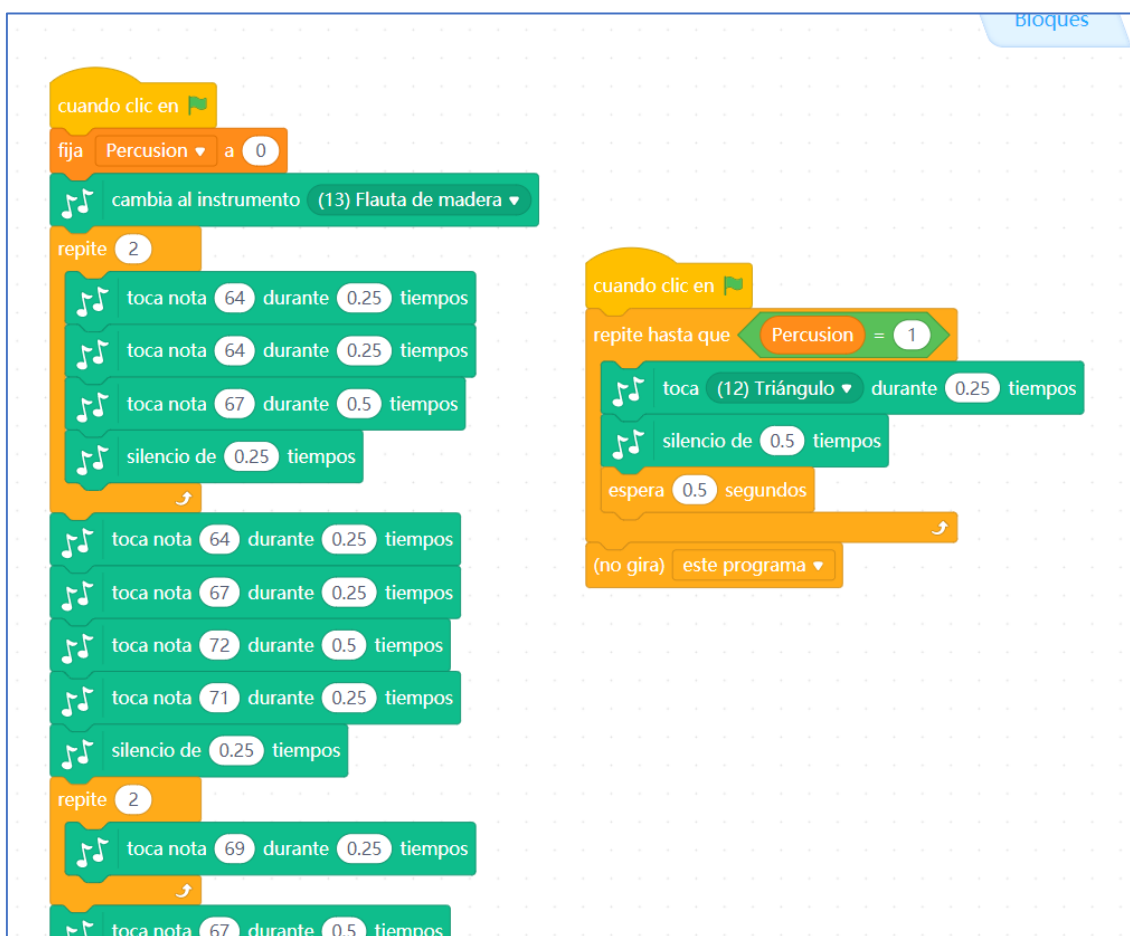


También podemos incluir silencios o esperas:



De la misma forma que la extensión “Lápiz”, esta extensión también es muy conocida ya que se incluía en los bloques iniciales de programación de mBlock3.

En la vida real, una canción gana en grandeza si hay otro músico que realice la percusión. En la siguiente imagen se observa precisamente eso, una percusión a la vez se programa una canción que, en nuestro ejemplo, sonará bajo la simulación de una flauta de madera:



Con la variable “Percusión” activamos o desactivamos su sonido:



2.2. Inteligencia Artificial

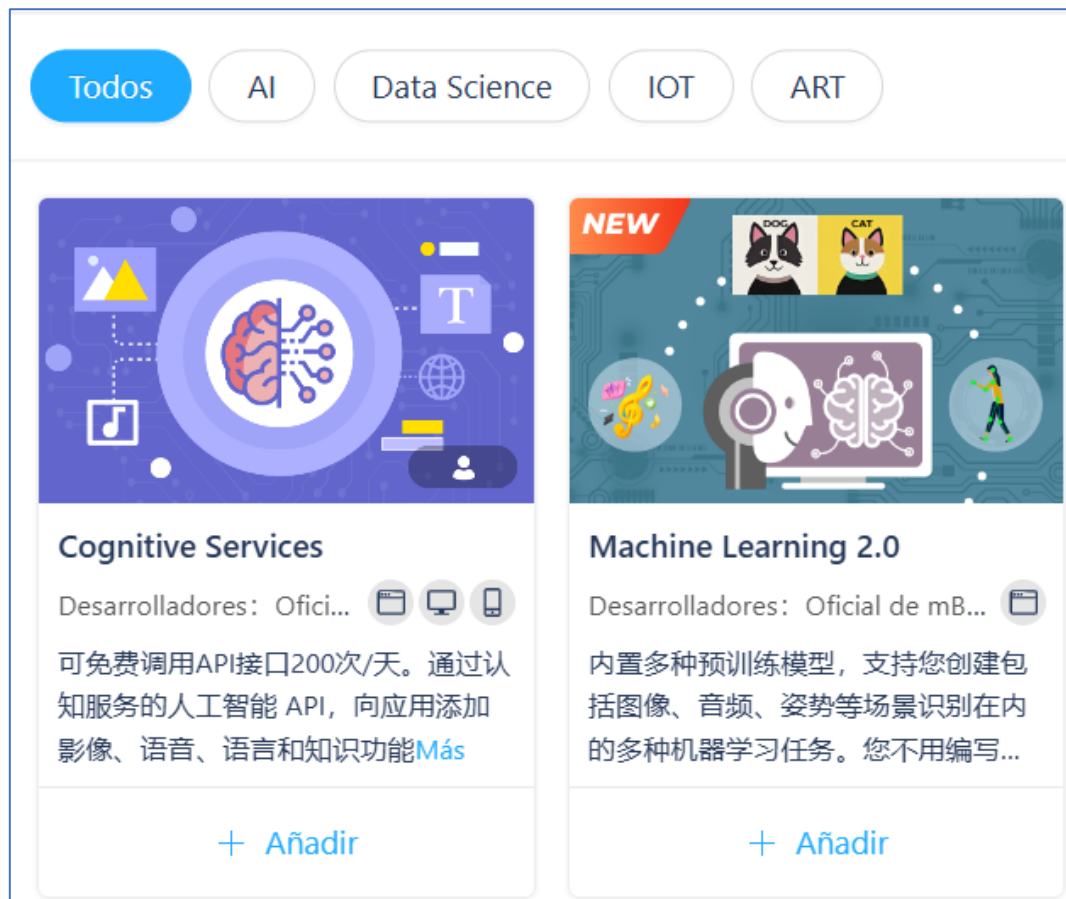
En esta temática nos encontramos con cuatro interesantes extensiones. Hoy en día, nuestros teléfonos móviles (smartphone) usan la IA para personalizar y acercar el producto a los deseos de sus propietarios. El uso de los asistentes virtuales que responden a preguntas, dan recomendaciones y ayudan a organizar las rutinas de sus propietarios son ejemplos de IA que se ha generalizado.

Podemos definir la IA como la capacidad que posee una máquina de presentar las mismas habilidades que los seres humanos. Y con esto me refiero a conceptos como el razonamiento, el aprendizaje, la creatividad, los sentimientos, etc.

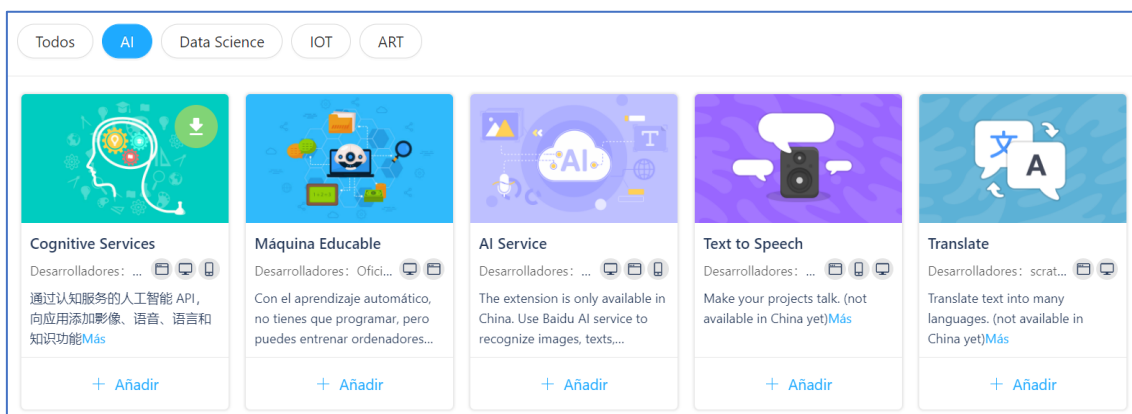


Extensiones AI en mBlock5 online

En el apartado “Todos” nos encontramos con una interesante y nueva aplicación: *Machine Learning 2.0*



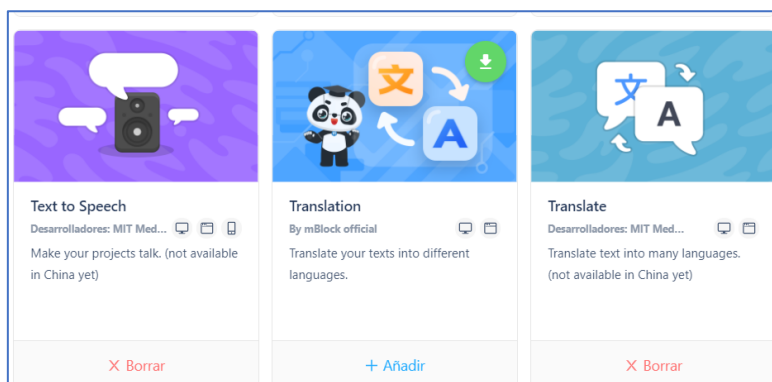
Disponible sólo en la **versión online**



Extensiones AI en mBlock5 offline

2.2.1. Extensión Texto a Voz y Traductor

Estas extensiones se incluyen dentro del grupo de Inteligencia Artificial del objeto y gracias a ellas nuestros sprites pueden hablar en diferentes idiomas e incluso traducir una frase o palabra eligiendo el usuario como traductor una larga lista de 47 idiomas diferentes.



Estas extensiones incorporan los siguientes comandos:

Texto a Voz (Text to Speech)	Traductor (Translation)

Con ambas extensiones se pueden diseñar programas interesantes. Por ejemplo, el siguiente script traduce de español al inglés y lo pronuncia con acento inglés:



Inclusive podemos hacer que el usuario elija con qué acento quiere que hable el programa (se puede escoger dentro de un rango de 23), que traduzca una frase o palabra y que a su vez la pronuncie en uno de los 47 idiomas diferentes con los que puede hablar. El programa que ejecuta estas acciones es el siguiente:



Si ejecutamos el programa haciendo clic en la bandera verde, primero se nos pide que escojamos un "acento" dentro de los 23 disponibles en la extensión: En la imagen se observa que hemos escogido el número 5 que se corresponde con inglés. (Ver imagen derecha).

A continuación, el script nos pide que seleccionemos el idioma con el que nos hablará el objeto "Panda". En la imagen se observa que hemos escogido el número 17, que se corresponde con euskera.



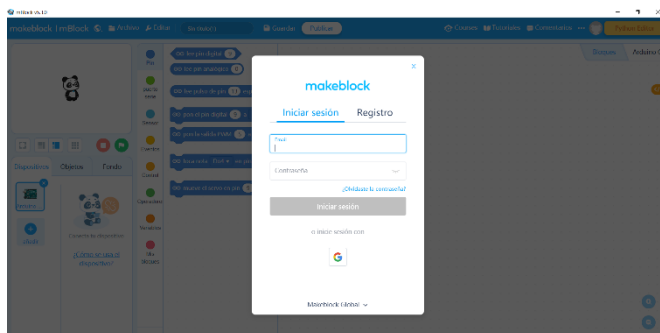


Finalmente, nos traducirá a euskera la frase que hemos escrito. En el ejemplo de la imagen el texto que traduce a euskera y lo pronuncia en euskera con acento inglés es: “Hola, me llamo Susana”:



2.2.2. Servicios Cognitivos

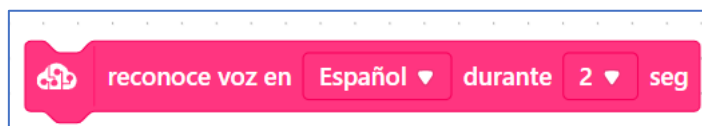
Esta extensión pertenece al bloque de Inteligencia Artificial y según dice la casa Makeblock, funciona mejor de forma online. Es decir, iniciando sesión con la cuenta en mBlock5.



La extensión nos ofrece una serie de servicios que requieren del uso de la webcam o del micrófono de nuestro PC. Estos servicios son:

Reconocimiento de voz, reconocimiento de texto impreso, reconocimiento de texto escrito a mano, reconocimiento de imágenes, reconocimiento de objetos comunes, reconocimiento de la edad de la persona y reconocimiento y detección de emociones. Y me explico:

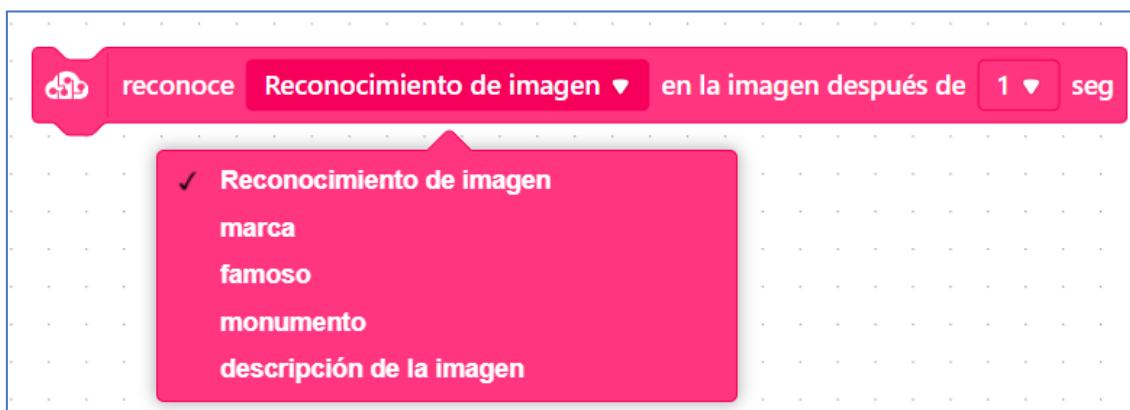
- **Reconocimiento de voz:** utiliza el micrófono para registrar nuestra voz y que esta se pueda usar para dar órdenes en un programa.



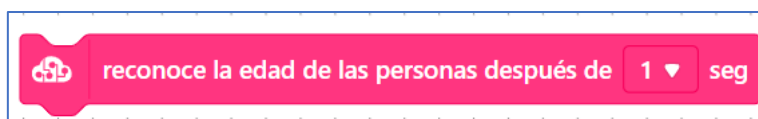
- **Reconocimiento de texto impreso:** utiliza la webcam para reconocer palabras escritas en un papel y tras ese reconocimiento podremos tomar decisiones sobre el desarrollo de un programa. Eso sí, si es texto impreso éste lo reconocerá en varios idiomas pero, si es texto escrito a mano, sólo lo reconocerá si está escrito en inglés.



- **Reconocimiento de imagen:** diferencia entre distintas imágenes y objetos que coloquemos delante de la webcam.



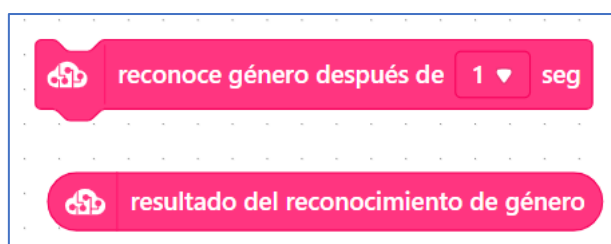
- **Reconocimiento de edad:** utiliza la webcam para decirnos aproximadamente la edad de la persona que detecta.



- **Grado de felicidad:** Observa nuestra cara a través de la webcam valorando entre 0 y 100 la felicidad que trasmites en ese momento. También permite diferenciar distintas emociones como sorpresa, tristeza, ira, etc.



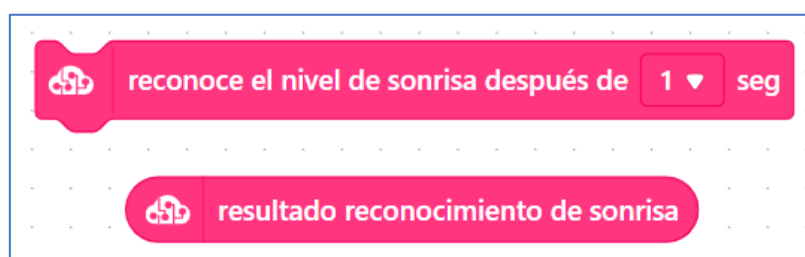
- **Reconocimiento de género:** diferencia entre género femenino y masculino.



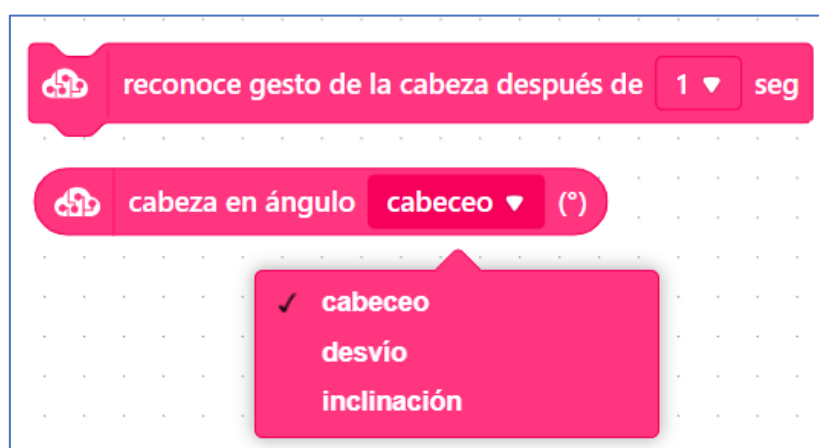
- **Detección de gafas:** nos dice si llevamos gafas y de que tipo.



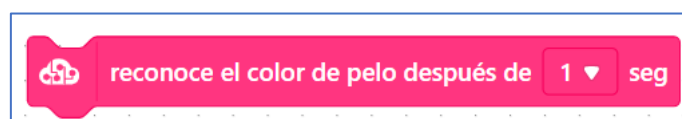
- **Nivel de sonrisa**, nos valora en una escala de 0 a 100 la sonrisa que lucimos en ese momento.



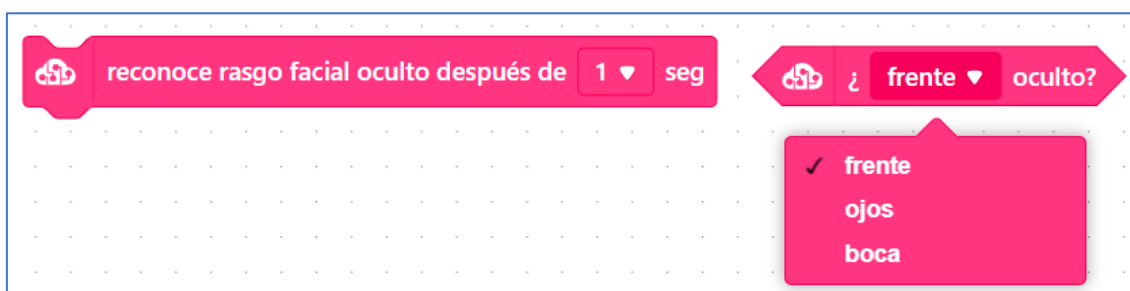
- Distingue **gestos de la cabeza** relativos a diferentes ángulos de inclinación de la misma.



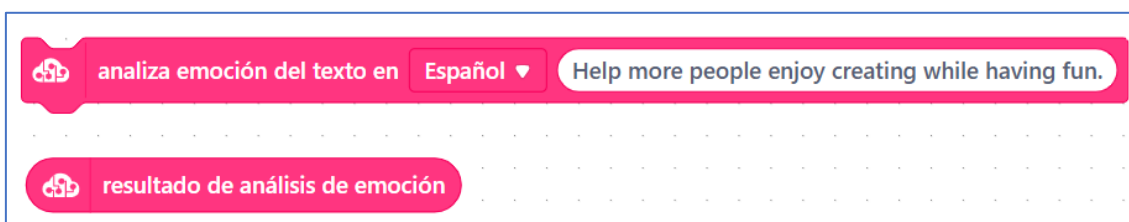
- Distingue el **color del pelo**.



- Reconoce **cambios en la cara relativos a la frente, cara y ojos**.



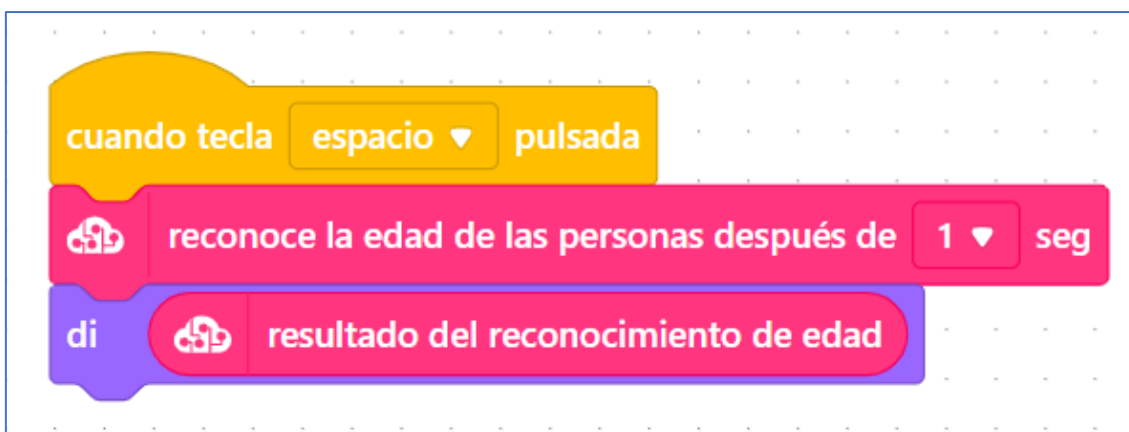
- Analiza si un texto escrito en diferentes idiomas es agradable o repulsivo.



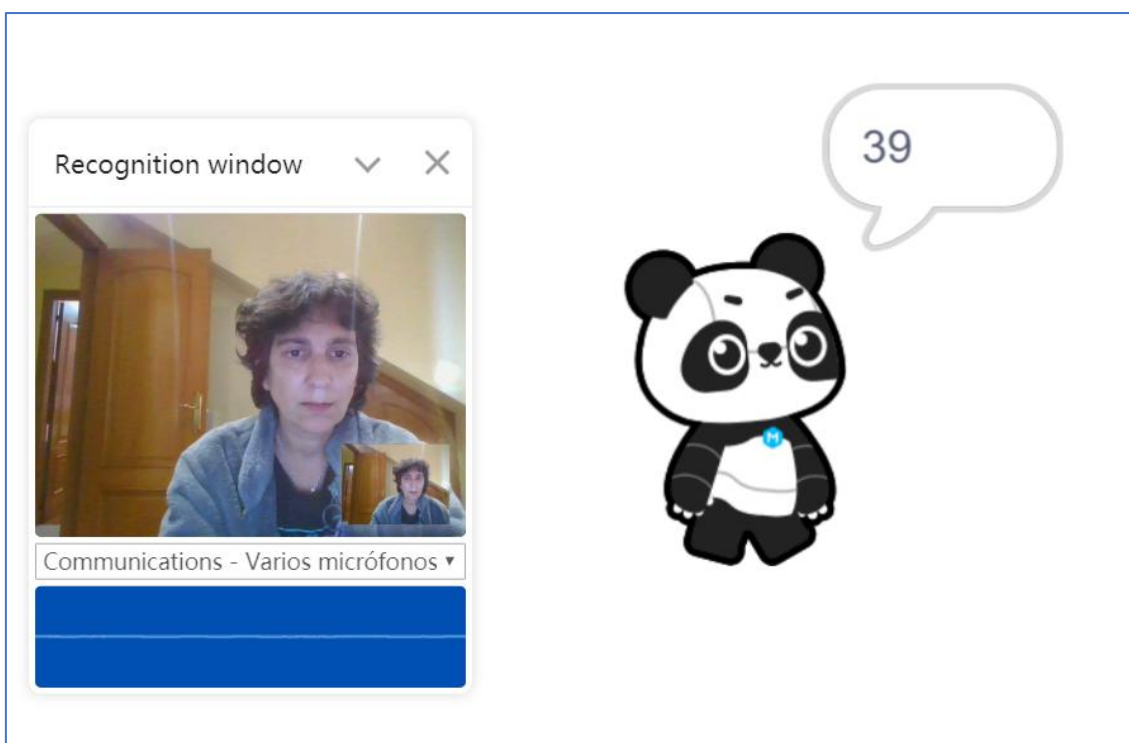
La verdad es que es aplicable en un extenso campo de situaciones.

Reto 1: Utilizar el reconocimiento de edad de una persona.

Para ello, simplemente, creamos para un objeto de mBlock5 este pequeño script:



El resultado es el siguiente:

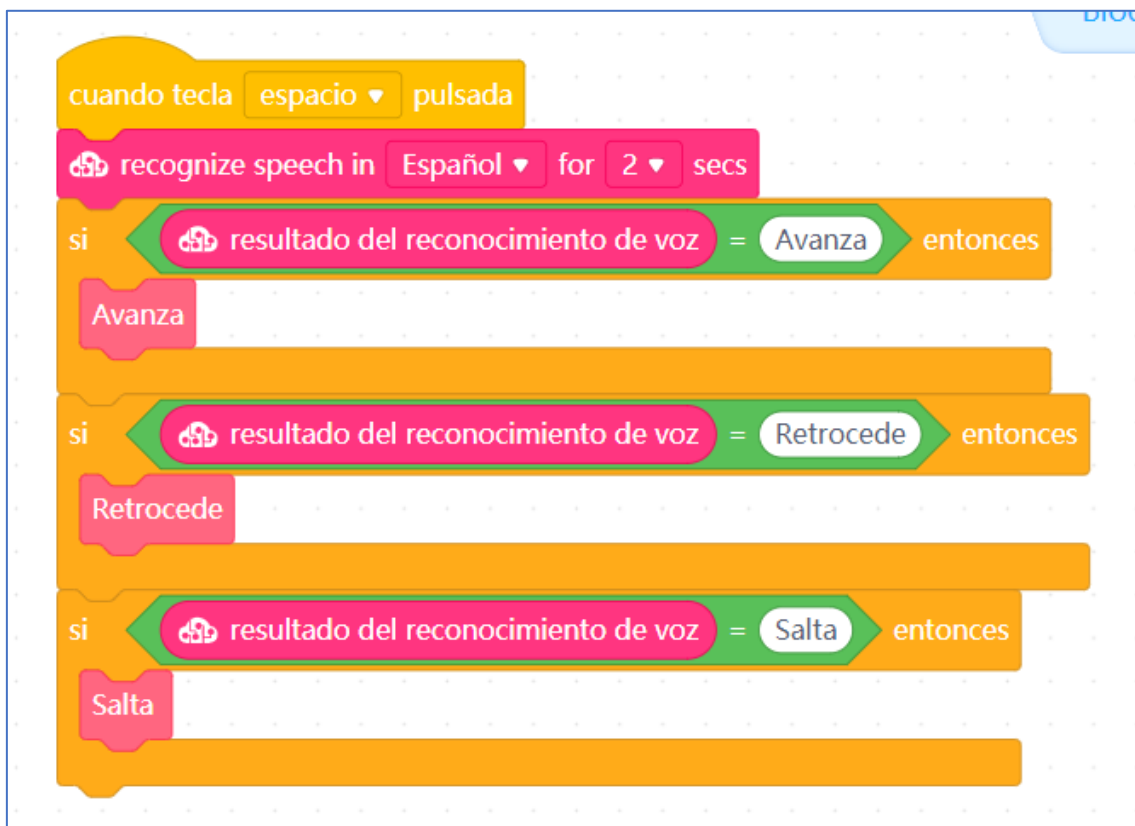


Reto 2: Utilizaremos el reconocimiento de voz para que el programa ejecute unas acciones previamente programadas para cada palabra detectada.

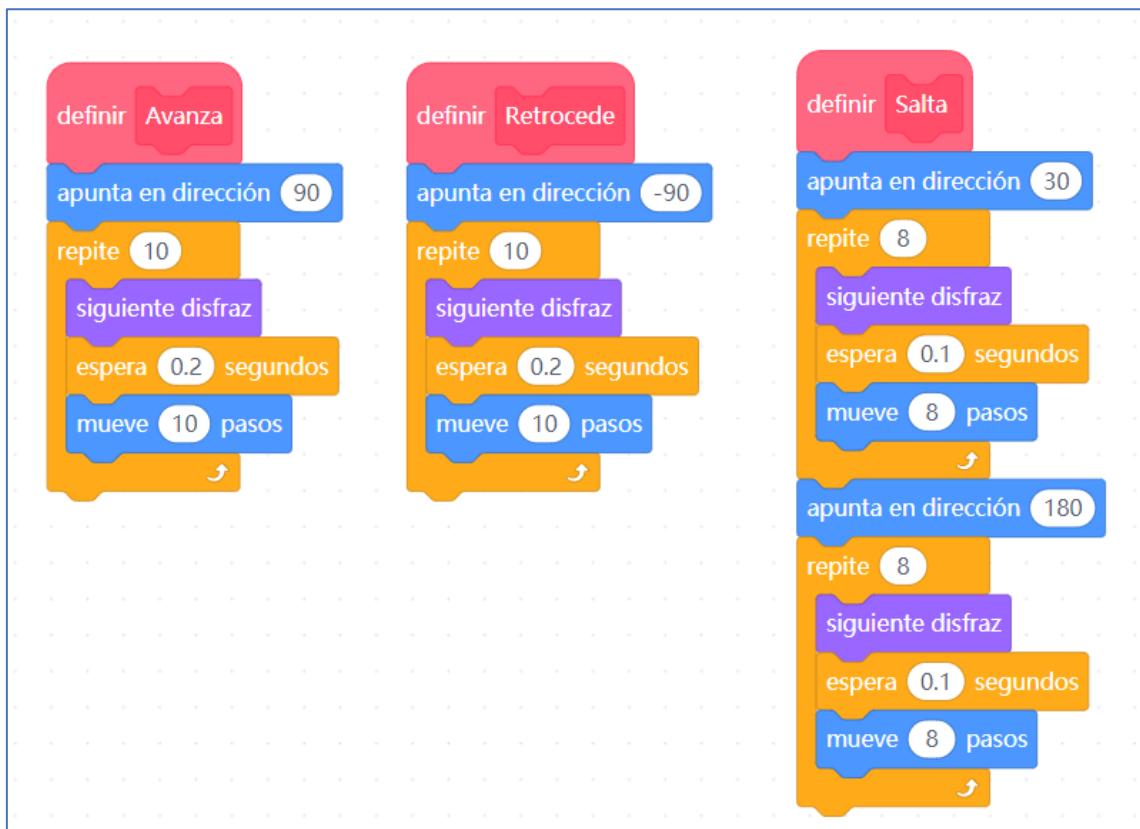


Escenario detectando “Avanza”

Cada vez que pulse la tecla espacio mBlock dispondrá de 2 segundos donde escuchará. Si escucha la palabra “Avanza” ejecutará el bloque diseñado para avanzar, si escucha la palabra “Retrocede” ejecutará el bloque correspondiente para retroceder y si escucha la palabra “Salta” ejecutará el bloque programado para saltar:

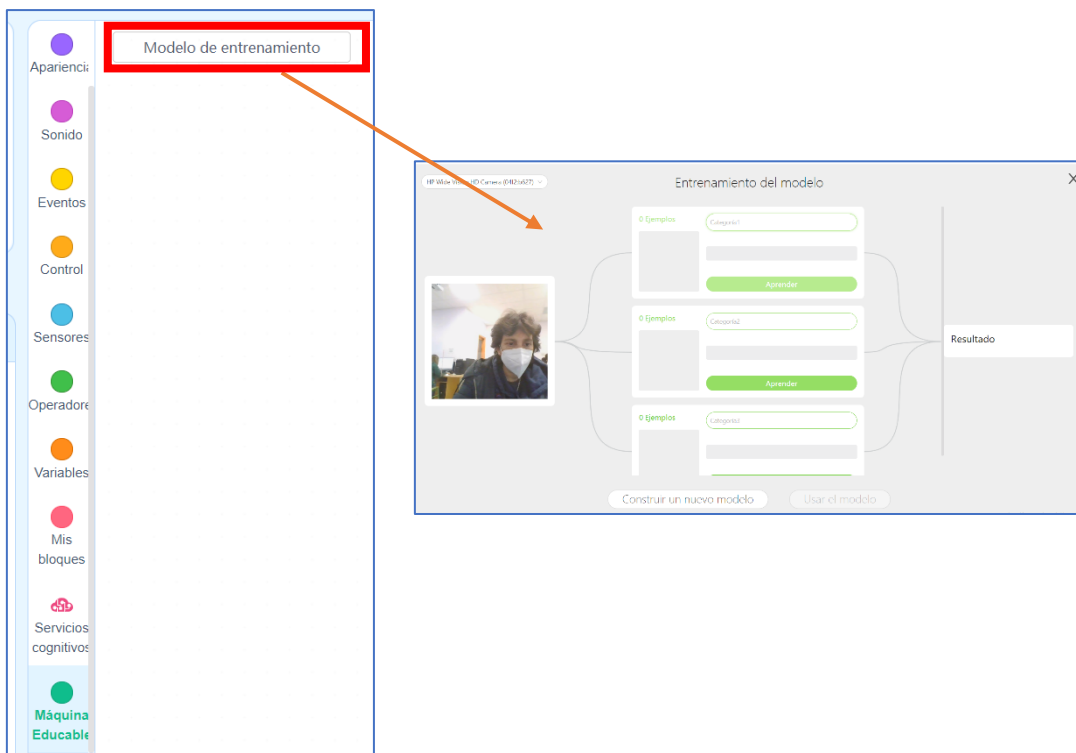


Los subprogramas para esas acciones son:

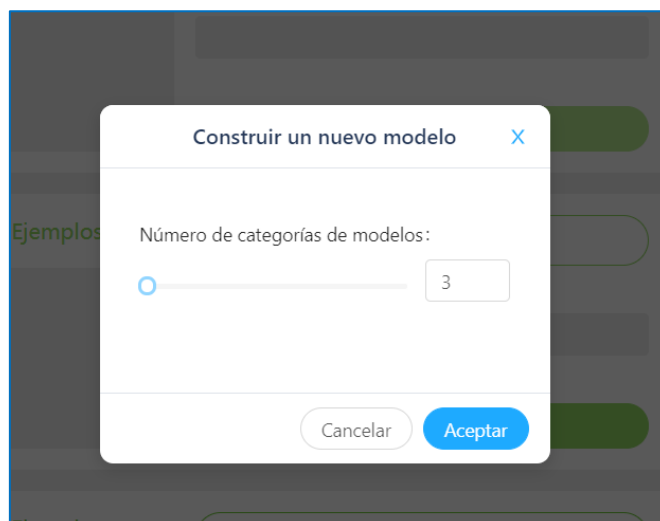


2.2.3. Máquina educable

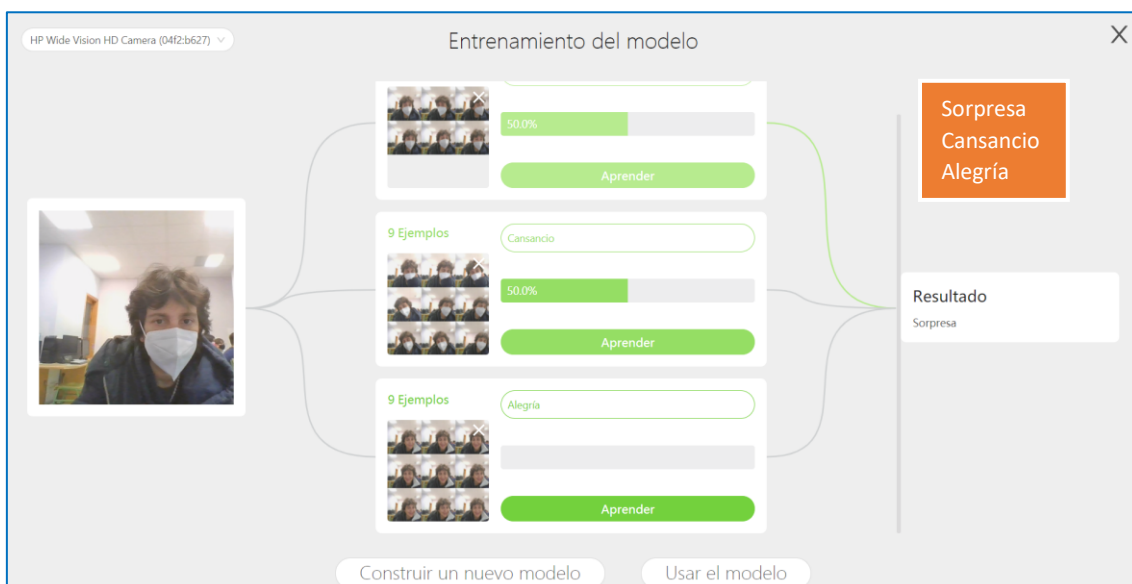
Primero debemos comenzar creando un modelo de entrenamiento para que nuestro dispositivo aprenda:



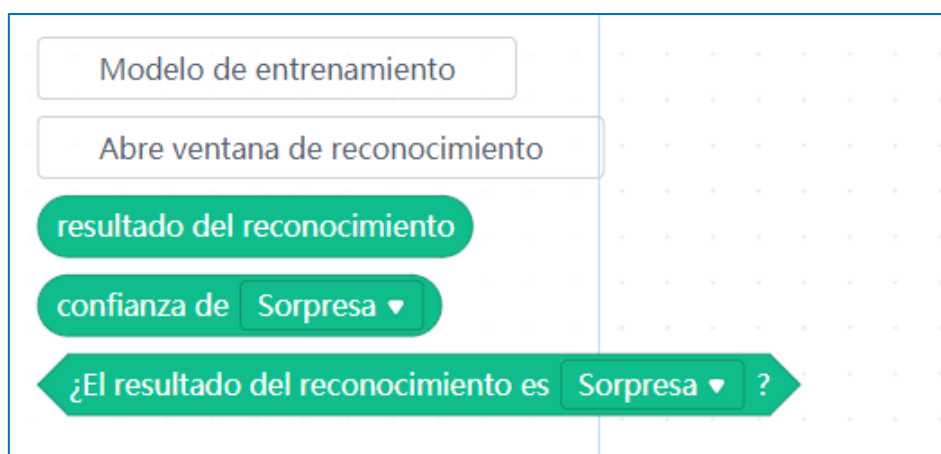
Decidimos el número de categorías a entrenar: (En la imagen se observa que he elegido 3)

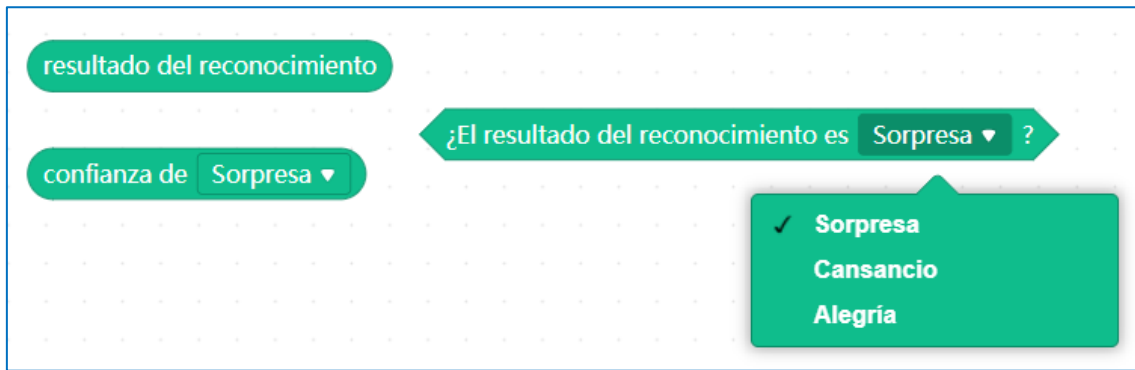


Definimos las categorías y entrenamos el modelo introduciendo imágenes que las representen:

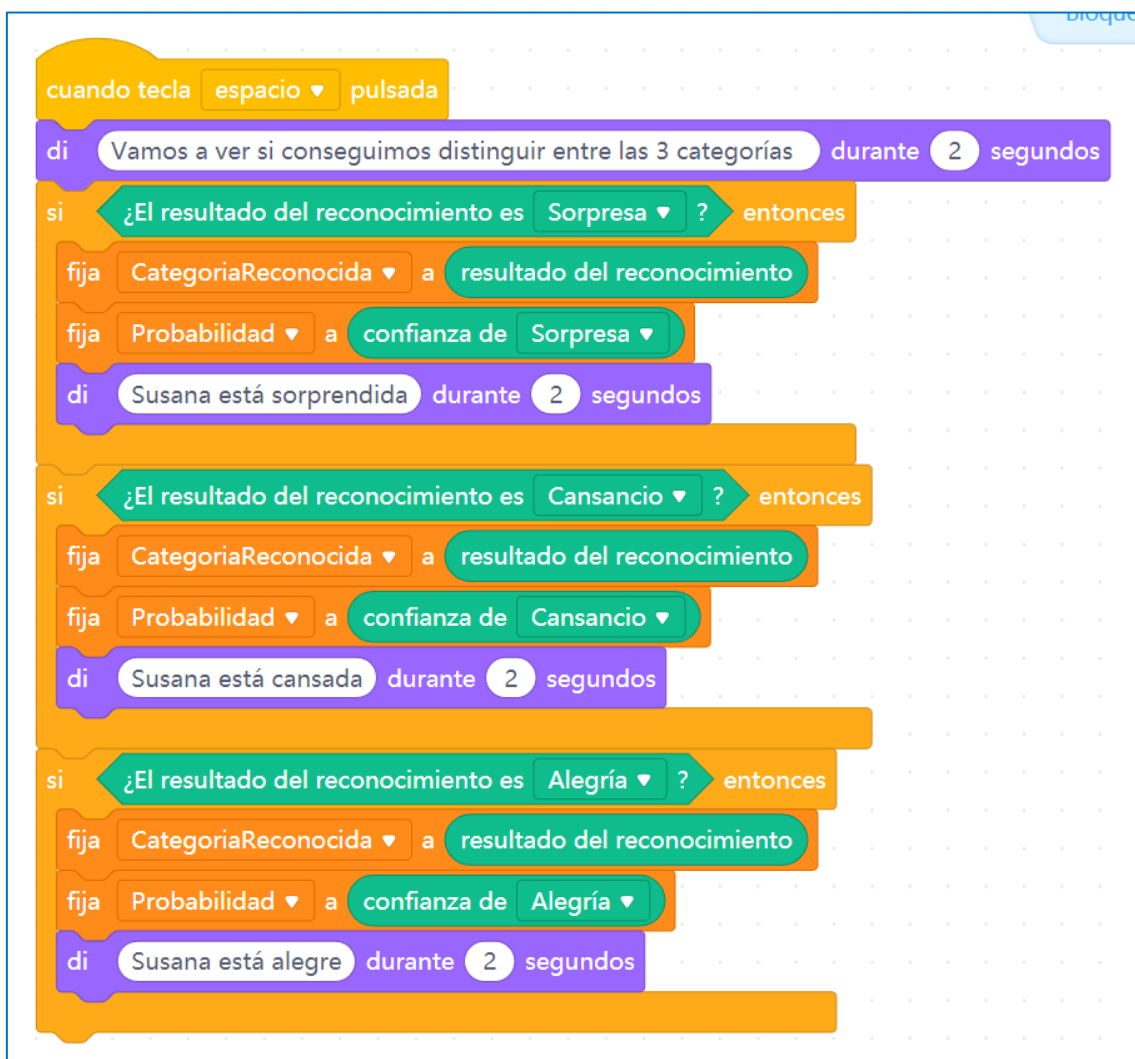


A continuación, hacemos clic en “Usar el modelo” creado. Vemos que se nos han creado nuevos comandos relacionados con las categorías que nuestro PC ha aprendido a reconocer:

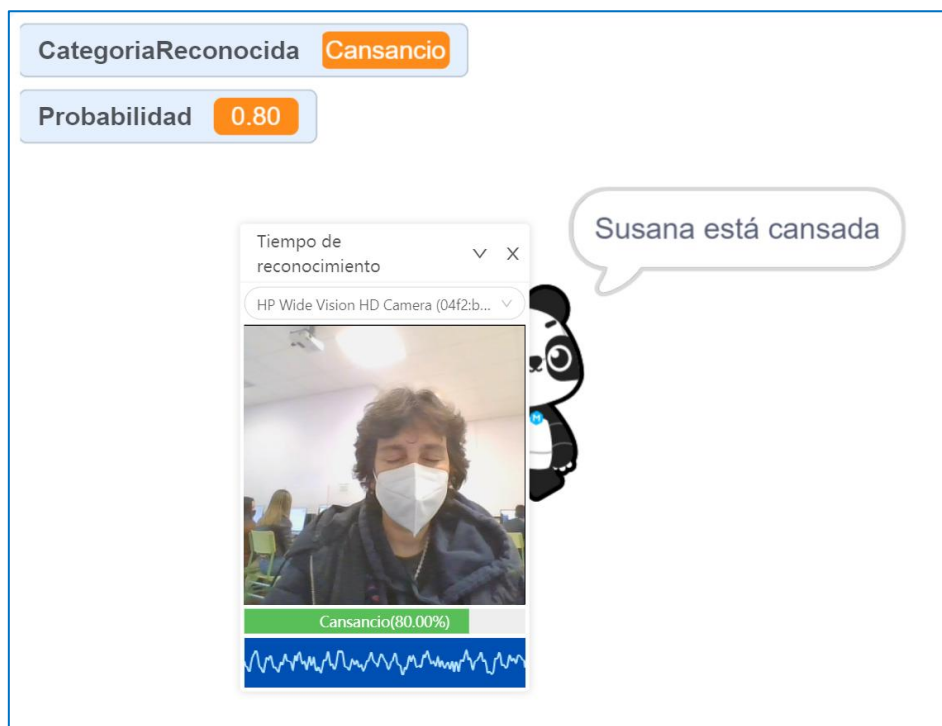




Finalmente, programamos el modelo donde se observan las tres categorías (sorpresa, cansancio y alegría) así como, la probabilidad de que la imagen que ve la Webcam represente una de esas categorías:

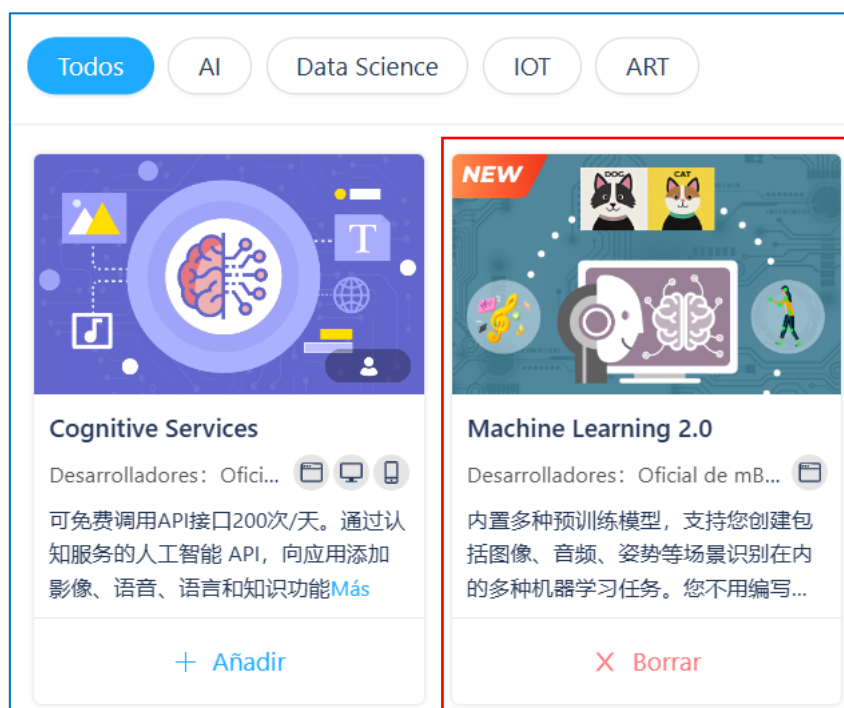


Vamos a comprobar si acierta o no:

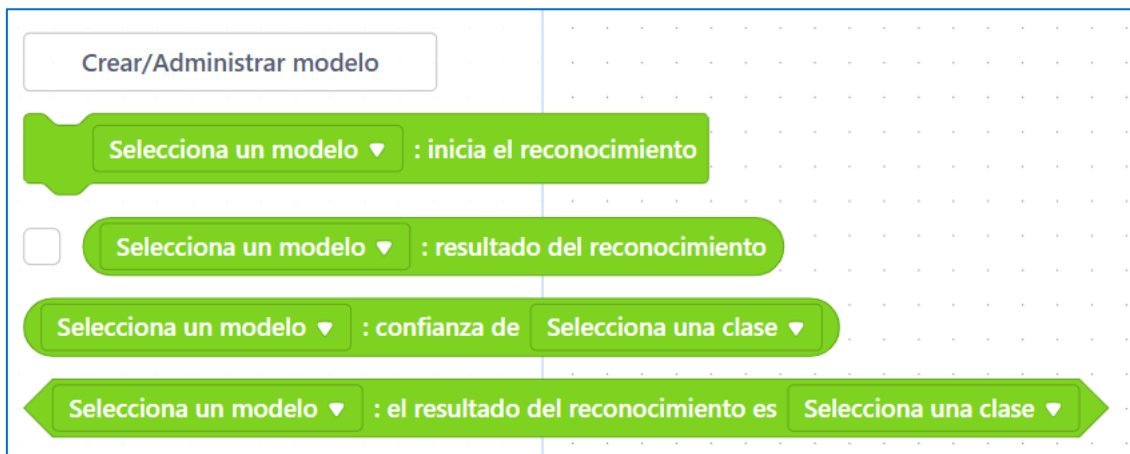


2.2.4. Machine Learning 2.0

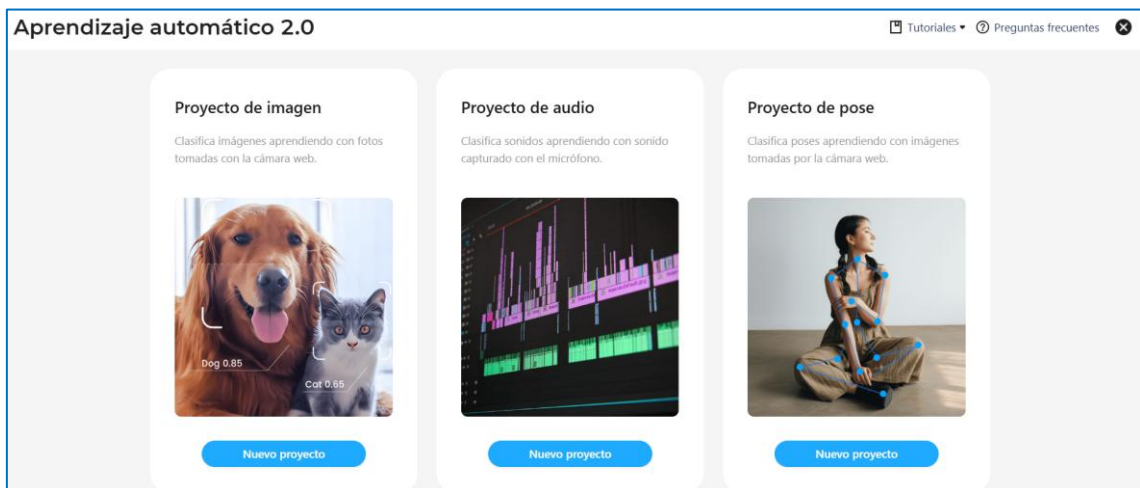
Actualmente, sólo es posible utilizar esta extensión en la versión mBlock online (introduciendo nuestro usuario y contraseña). Podemos añadirla buscándola en el apartado “Todos”:



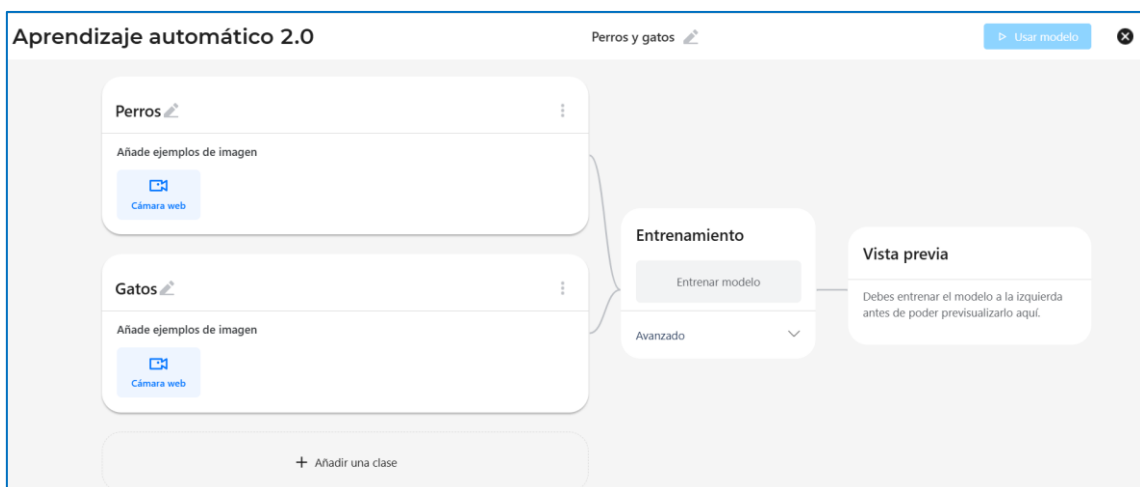
Observamos que podemos crear y/o administrar proyectos



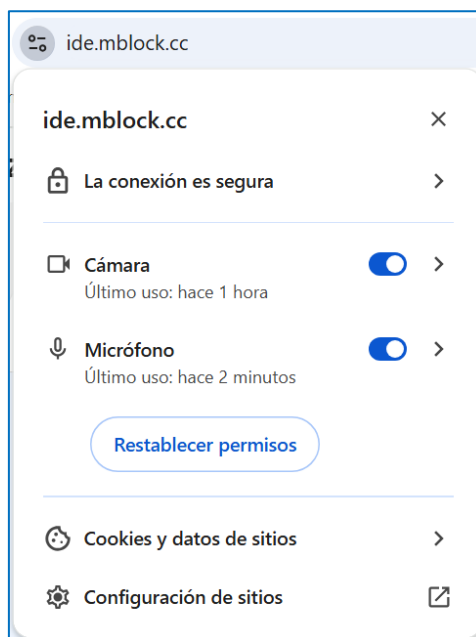
Comenzamos haciendo clic en el recuadro *Crear/Administrar modelo* que nos proporcionará acceso a tres posibilidades (como si entráramos en Teachable machine): trabajar [con imágenes](#), [audios](#) o incluso [poses](#). (Los links se dirigen a pequeños videos que muestran como introducir los datos, entrenarlos y exportar el modelo)



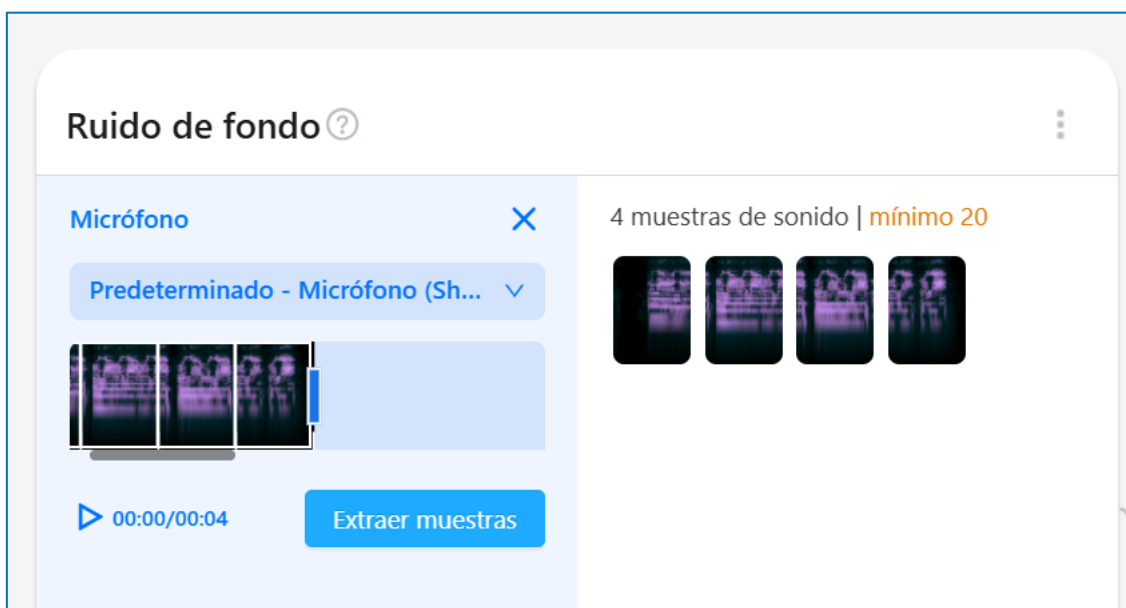
En el proyecto de imágenes y de poses, la subida de las mismas se realiza habilitando la webcam:



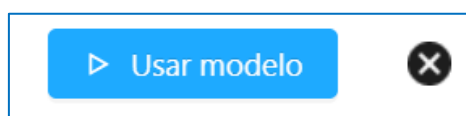
Relativo al audio, debemos permitir el acceso al micrófono:

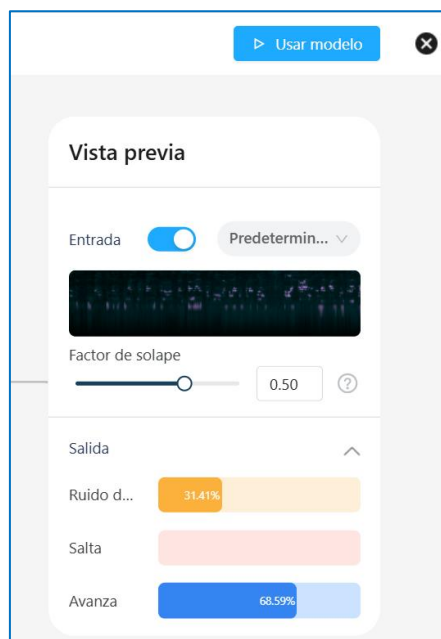


Y escoger el micrófono adecuado:



Tras entrenarlo (con imágenes, audios o poses), podemos comprobar si funciona bien o no (Vista previa). Finalmente, podemos *Usar el modelo* y programarlo bajo mBlock:

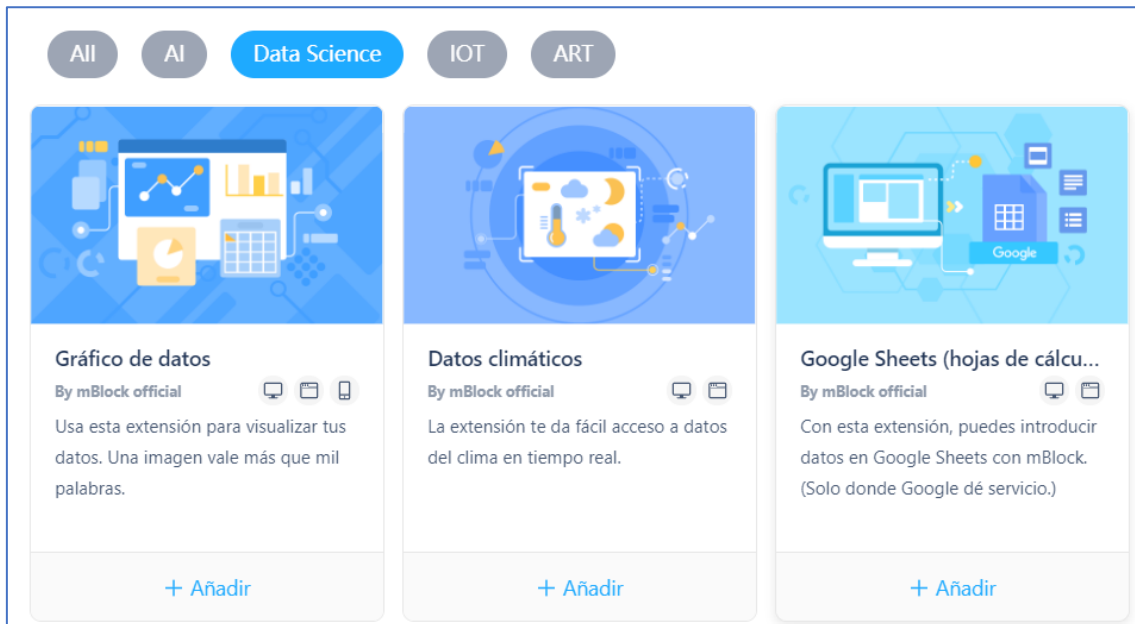




Proyecto entrenado "SaltaAvanza"

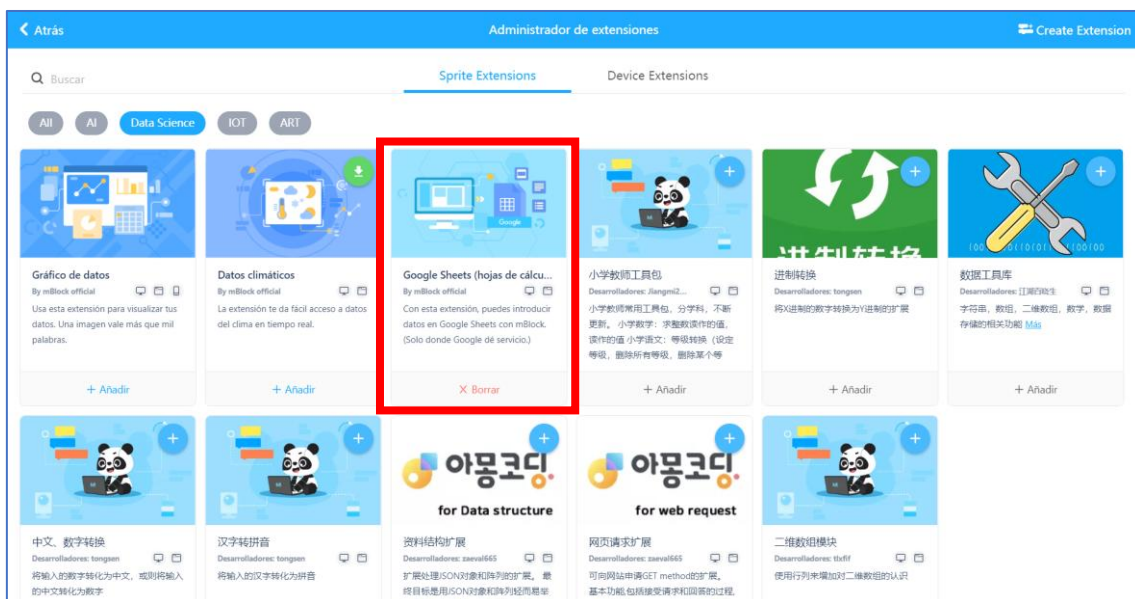
2.3. Datos de ciencia

mBlock5 nos ofrece tres interesantes extensiones que ayudan a gestionar los datos que, por lo general, recibiremos de los sensores.

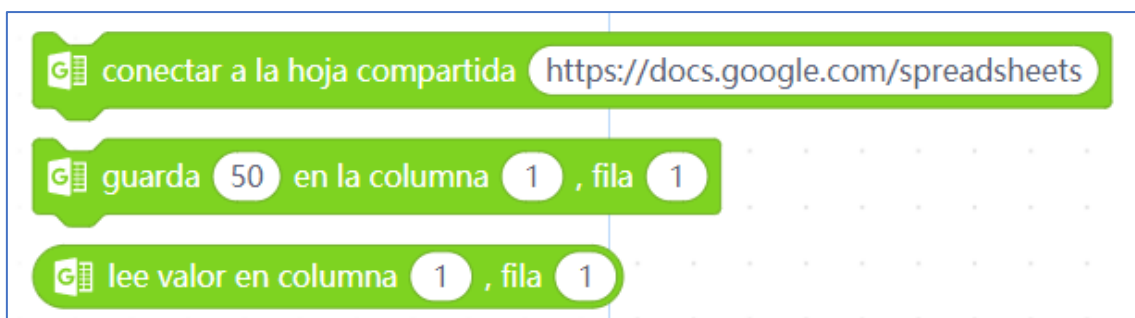


2.3.1. Hoja de cálculo de Google o Google Sheets

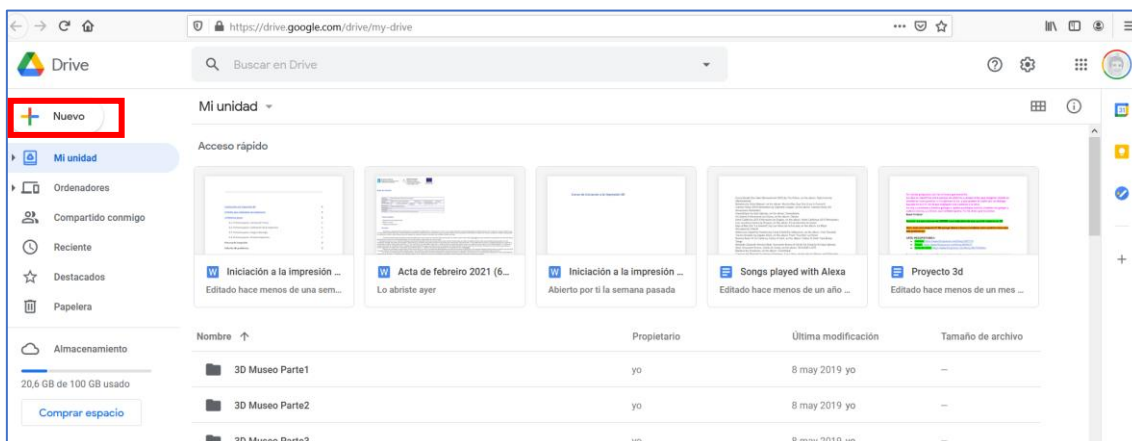
Esta es una extensión que me encanta ya que le veo muchas aplicaciones. En la siguiente imagen se observa que ya la he incluido (añadido) en mi programa:



Los comandos de la extensión son los siguientes:



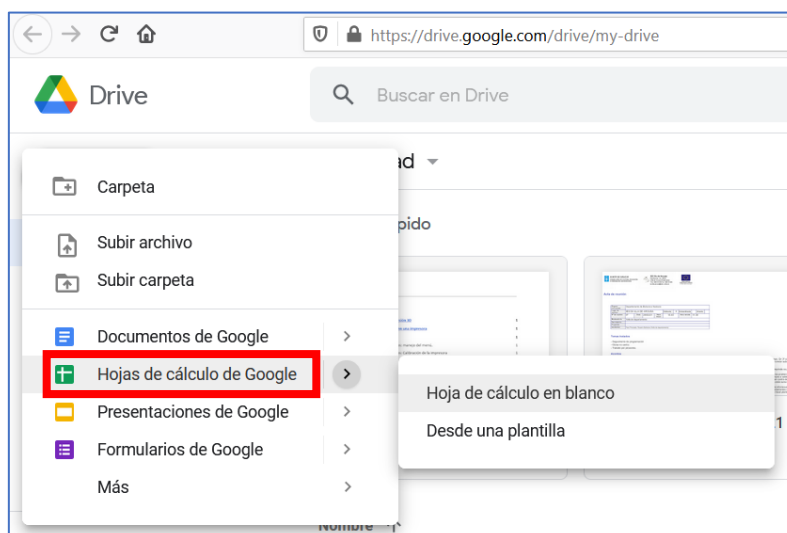
Para trabajar con ella es indispensable disponer de una cuenta de correo electrónico *Gmail* y entrar en nuestro *Drive* creando un archivo compartido en el formato de una hoja de cálculo.



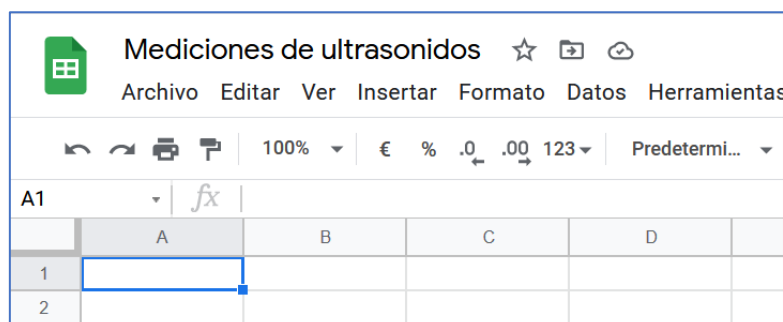
Vamos a ver dos ejemplos de uso.

Reto 1: Grabar distancias con el sensor de ultrasonidos

Tras entrar en *Drive* hacemos clic en “Nuevo” y escogemos *Hojas de cálculo de Google* > *Hojas de cálculo en blanco*:



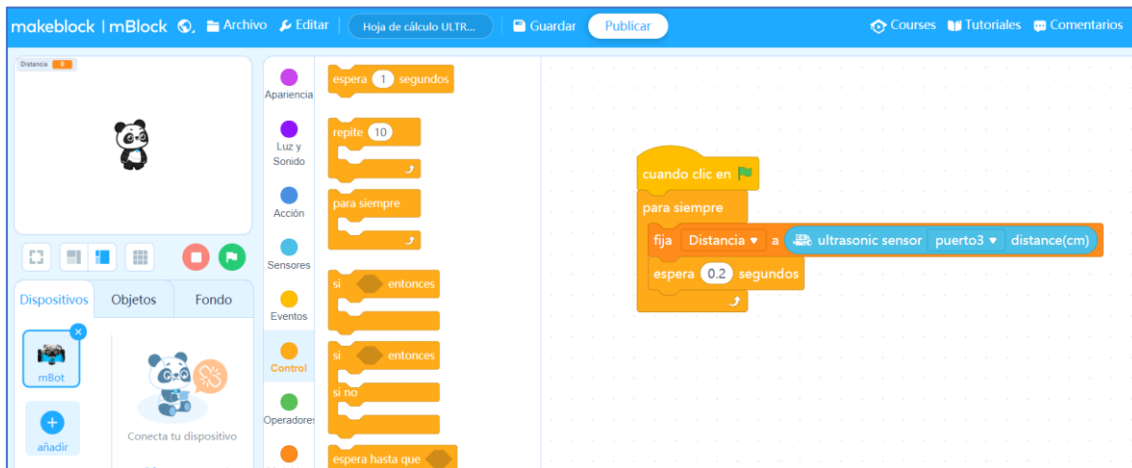
Ya tenemos un archivo y, como es lógico, vamos a ponerle un nombre. Lo llamaremos “Mediciones de ultrasonidos” a nuestra hoja de cálculo:



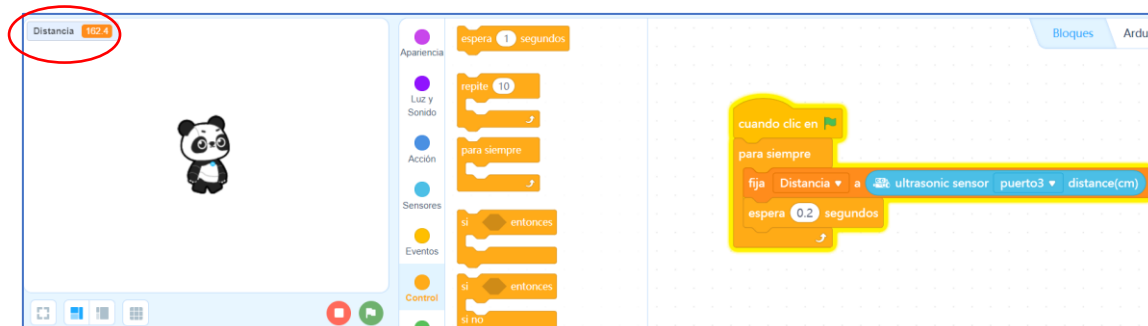
El nombre del archivo lo dice todo; necesitamos un sensor de ultrasonidos y obviamente, el sensor debe pertenecer a un dispositivo que puede ser el robot mBot o el Ranger (que ya lo

incluyen en el propio robot) o por ejemplo una placa Arduino Uno (en este caso deberíamos montar el circuito eléctrico). Yo usaré un mBot y por lo tanto, escojo este dispositivo.

En mi caso, tras elegir el dispositivo mBot sobre *mBlock5*, su pequeño script que me daría las mediciones del sensor de ultrasonidos cada 0.2s estando el sensor conectado al puerto 3 de la placa, sería el siguiente:



Conectamos el mBot y vemos si funciona nuestro programa simple: En la siguiente imagen observamos que por ahora todo va bien ya que va enviando la distancia a la variable "Distancia" que podemos ver en el escenario del programa.

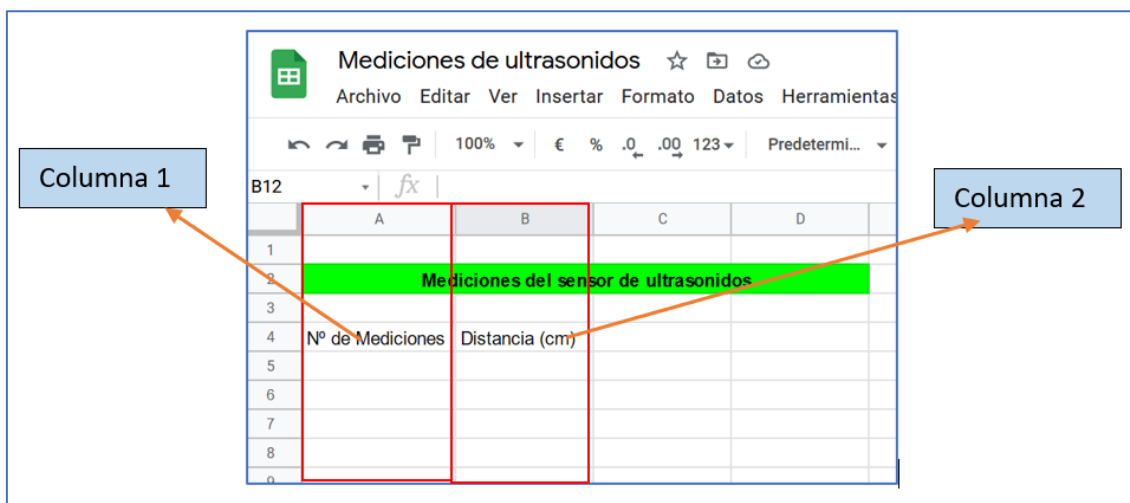


A continuación, pasamos a la pestaña Objetos y descargamos la extensión Google Sheets (pensar que es una extensión pensada para objetos).

La idea es, por ejemplo, que el mBot consiga enviar a la hoja de cálculo los datos de 10 mediciones del sensor de ultrasonidos y que estas se guarden en la hoja de cálculo. Podríamos hacer que se guardaran de forma horizontal (en filas) o vertical (en columnas). En nuestro reto lo vamos a hacer de forma vertical (en diferentes filas dentro de una misma columna).

Para ello, en el apartado objetos de mBlock5 escribimos un pequeño script cuya función será grabar las distancias y el orden de las medidas en nuestro archivo de la hoja de cálculo:

NOTA importante: Usaremos la columna 1 para el número de medidas y la columna 2 para el valor de las distancias.



Queremos que los datos se escriban a partir de la fila número 5, justo debajo de “Nº de Mediciones” y de “Distancia (cm)”.

Bloques

Debemos introducir el link de nuestra hoja de cálculo.
El enlace debe estar compartido de modo que:

- Cualquier persona con el enlace
- Y con permisos de EDITOR

La variable ColumnaMedidas coincide con una variable que podríamos llamar Número de Medidas y por eso no la creo

```

cuando clic en [bandera]
  conectar a la hoja compartida https://docs.google.com/spreadsheets
  fija ColumnaMedidas a 1
  fija Filas a 5
  fija ColumnaDistancias a 2
  repite 10
    guarda ColumnaMedidas en la columna 1, fila Filas
    guarda Distancia en la columna 2, fila Filas
    cambia Filas por 1
    cambia ColumnaMedidas por 1
    espera 0.5 segundos
  
```

Finalmente, solo nos queda introducir en el script el enlace correcto. Ojo, no es un enlace cualquiera, es un link **compartido y editable**.

Para conseguirlo primero hacemos clic en botón verde superior derecha “Compartir”:

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Mediciones del sensor de ultrasonidos									
3											
4		Nº de Mediciones	Distancia (cm)								
5											

Se nos abre la siguiente imagen y en ella escogemos la opción “[Cambiar a cualquier persona con el enlace](#)”:

Compartir con personas y grupos

Añadir personas y grupos

Susana Oubiña (tú) *Propietario*

[Enviar comentarios a Google](#) **Hecho**

Obtener enlace

Restringido Solo las personas añadidas pueden abrir este enlace

[Cambiar a cualquier persona con el enlace](#) [Copiar enlace](#)

Se nos ofrecerán tres opciones y entre ellas escogemos el modo **Editor** (por defecto nos sale el modo Lector):

Obtener enlace

[https://docs.google.com/spreadsheets/d/1i4bHMP3rZxH6zX71nV9lgr ...](https://docs.google.com/spreadsheets/d/1i4bHMP3rZxH6zX71nV9lgr...) [Copiar enlace](#)

Cualquier persona con el enlace **Editor**

Cualquier usuario de Internet con este enlace puede editar esto


[Enviar comentarios a Google](#)

Lector


Comentador

Editor

Copiamos el enlace y lo pegamos en nuestro Script.




Obtener enlace




<https://docs.google.com/spreadsheets/d/1i4bHMP3rZxH6zX71nV9lgr ...>
Copiar enlace

Debemos introducir el link de nuestra hoja de cálculo.
El enlace debe estar compartido de modo que:

- Cualquier persona con el enlace
- Y con permisos de EDITOR

cuando clic en 


 conectar a la hoja compartida <https://docs.google.com/spreadsheets/d/1i4bHMP3rZxH6zX71nV9lgr ...>


fija ColumnaMedidas ▼ a 1

fija Filas ▼ a 5

fija ColumnaDistancias ▼ a 2

repite 10

 guarda ColumnaMedidas en la columna 1 , fila Filas

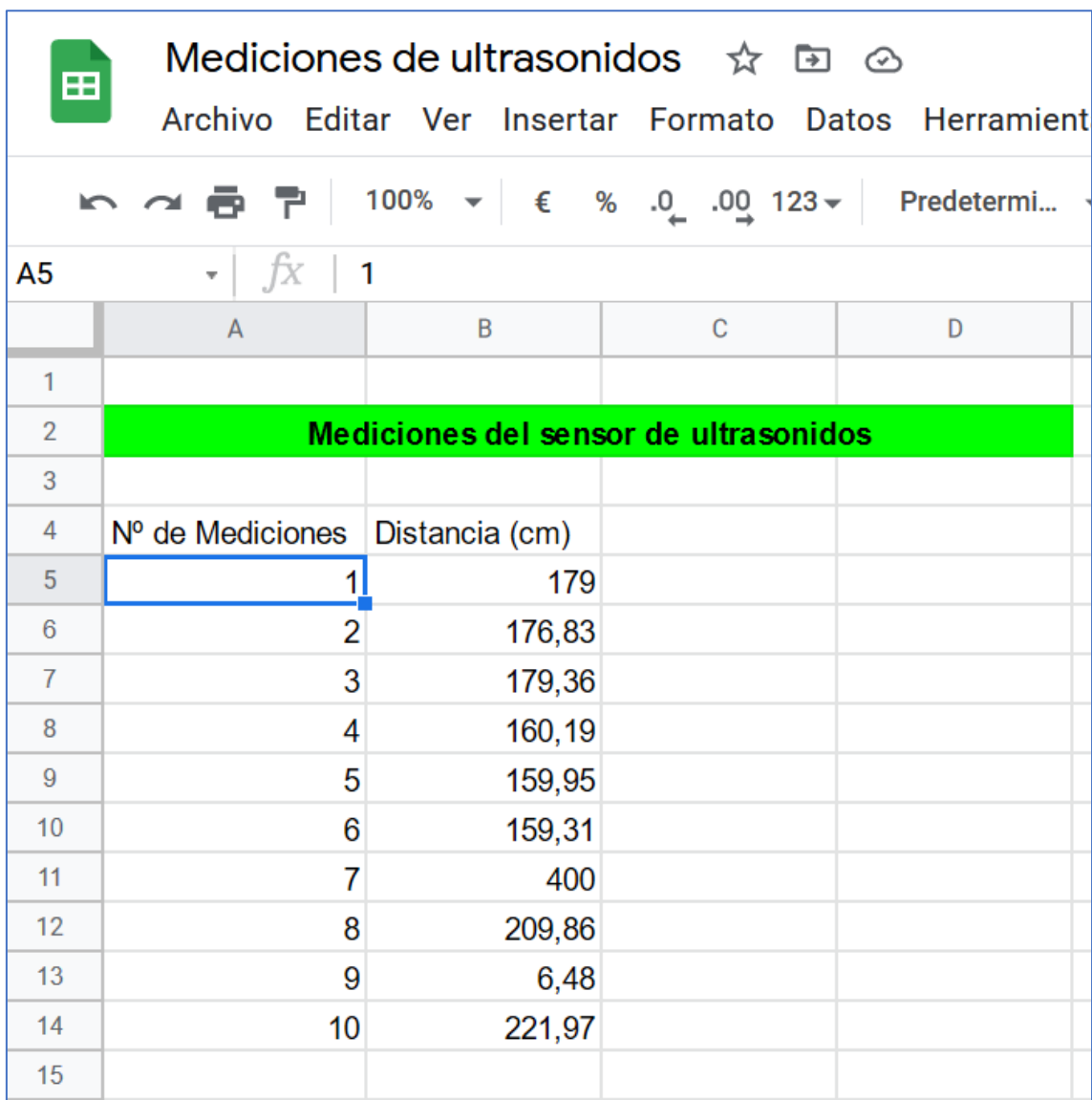
 guarda Distancia en la columna 2 , fila Filas

cambia Filas ▼ por 1

cambia ColumnaMedidas ▼ por 1

espera 0.5 segundos

Tras ejecutar el programa haciendo clic en la bandera verde, podemos ver que las mediciones de las distancias se han grabado en nuestra hoja de cálculo “Mediciones de ultrasonidos”:



Mediciones de ultrasonidos

Archivo Editar Ver Insertar Formato Datos Herramientas

100% € % .0 .00 123 Predetermi...

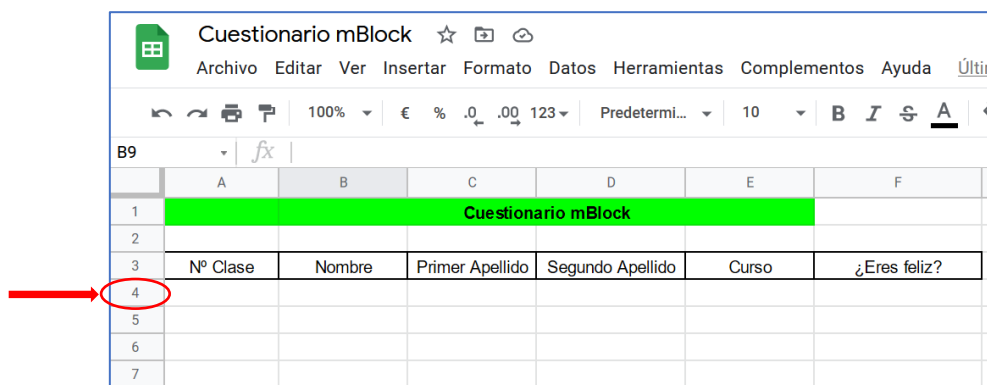
A5 fx 1

	A	B	C	D
1				
2	Mediciones del sensor de ultrasonidos			
3				
4	Nº de Mediciones	Distancia (cm)		
5	1	179		
6	2	176,83		
7	3	179,36		
8	4	160,19		
9	5	159,95		
10	6	159,31		
11	7	400		
12	8	209,86		
13	9	6,48		
14	10	221,97		
15				

Reto 2: Cuestionario

Vamos a crear un cuestionario y compartirlo con nuestro alumnado, de modo que, ellos puedan contestar a las preguntas y que no borren los datos que otros han escrito.

Antes de nada, creamos la hoja de cálculo en *Google Drive*. Supongamos unas cuestiones o campos a cubrir por el alumno/a como las que se muestran en la imagen siguiente:



Cuestionario mBlock

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Últim

100% € % .0 .00 123 Predetermi... 10 B I S A

B9 fx

	A	B	C	D	E	F
1	Cuestionario mBlock					
2						
3	Nº Clase	Nombre	Primer Apellido	Segundo Apellido	Curso	¿Eres feliz?
4						
5						
6						
7						

Para asociar la fila en la que se guardarán los datos de cada alumno lo haremos utilizando su Nº Clase que guardaremos en una variable “Fila” aumentada en 3 unidades. Y me explico.

Como queremos que el alumno/a con numero de clase 1 escriba en la fila 4 (ver imagen anterior), sólo debemos sumarle 3 al número de clase y asociar esta suma a la variable “Fila”:



Tras ejecutarlo, nos encontramos con los siguientes datos grabados en nuestra hoja de cálculo **compartida y editable** de Google:

	A	B	C	D	E	F
1	Cuestionario mBlock					
2						
3	Nº Clase	Nombre	Primer Apellido	Segundo Apellido	Curso	¿Eres feliz?
4						
5						
6						
7						
8	5	Susana	Oubiña	Falcón	2ESO B	si
9						
10						
11						

A poco que pensemos, obviamente, cualquiera podría entrar en el link de la hoja de cálculo y borrar algún dato. En la siguiente imagen, sin iniciar sesión (es decir, con sólo el link y un navegador), he accedido a la hoja de cálculo y he borrado un dato. Pensar que no he iniciado sesión. En fin, ¡borrar un dato no está bien!

	A	B	C	D	E	F	G	H	I	J	K	L
1	Cuestionario mBlock											
2												
3	Nº Clase	Nombre	Primer Apellido	Segundo Apellido	Curso	¿Eres feliz?						
4												
5												
6												
7												
8	5	Susana	Oubiña	Falcón		si						
9												
10												

¿Cómo solventar este problema?

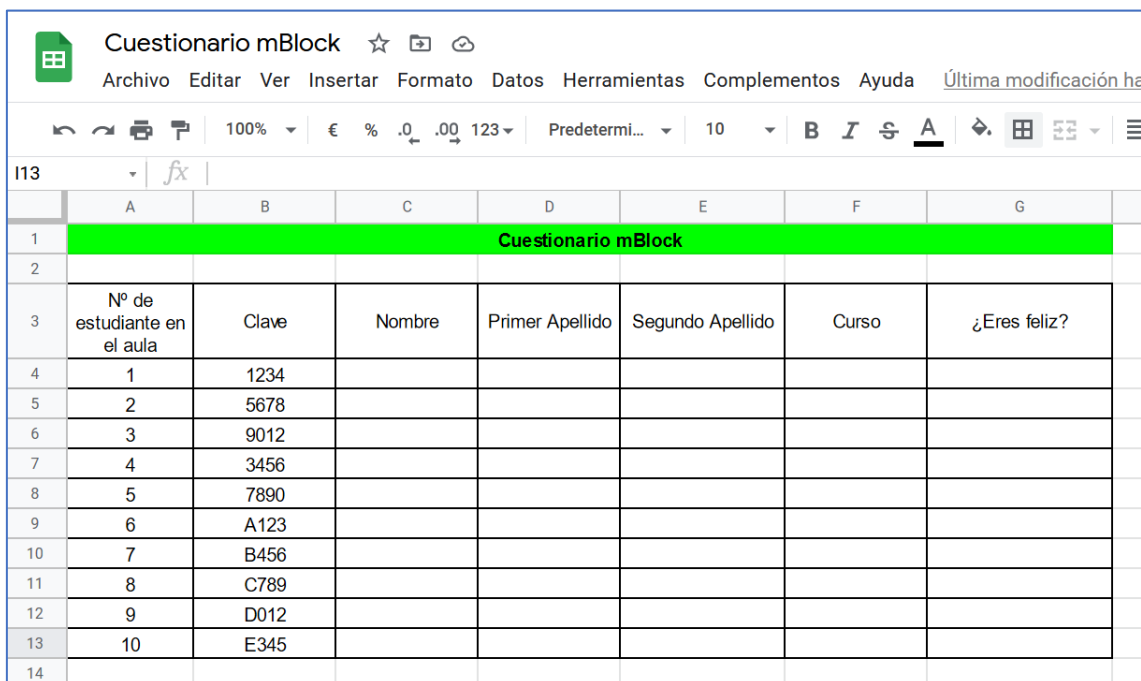
Solo se puede solucionar si, justo después de que el alumno/a incluya sus respuestas, entramos en la hoja de cálculo y cambiamos su modo “Editor” a modo “Lector”. Esta es la dura realidad. Si nos interesa realizar un cuestionario debemos ser rápidos o pillos, implementamos el programa en el aula y, tras recibir los datos, cambiamos el link de modo “Editor” a modo “Lector”. Tras este cambio, ellos no podrán hacer cambios.

Aunque tengamos claro la realidad, vamos a intentar hacer creer al alumno que la cosa es “seria” y para ello incluiremos en el programa anterior una contraseña de acceso a la hoja de cálculo. Todos sabemos que es innecesaria porque disponen del link en modo “Editor”.

Supongamos que tenemos 10 estudiantes y cada estudiante conoce su número de estudiante y su clave para acceder a las cuestiones planteadas en el programa.

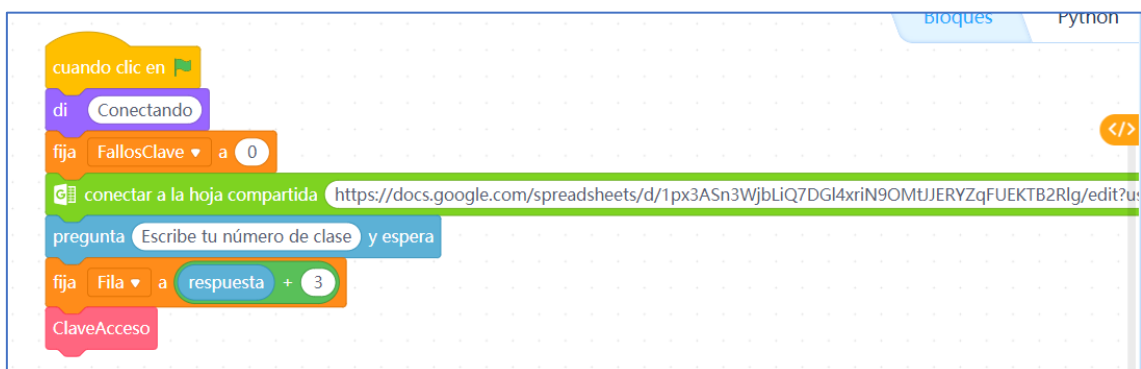
Aunque el alumnado sabe su número de estudiante y su clave, ya que previamente se la hemos dado, vamos a dar tres posibilidades de errores a la hora de incorporar la clave. Para ello necesitamos que mBlock5 las vaya contabilizando. Para ello crearemos una nueva variable que denominaremos “FallosClave” y que inicializamos a 0.

En la siguiente imagen se observa que hemos introducido el número de estudiante en el aula y su clave de acceso:



	A	B	C	D	E	F	G
1	Cuestionario mBlock						
2							
3	Nº de estudiante en el aula	Clave	Nombre	Primer Apellido	Segundo Apellido	Curso	¿Eres feliz?
4	1	1234					
5	2	5678					
6	3	9012					
7	4	3456					
8	5	7890					
9	6	A123					
10	7	B456					
11	8	C789					
12	9	D012					
13	10	E345					
14							

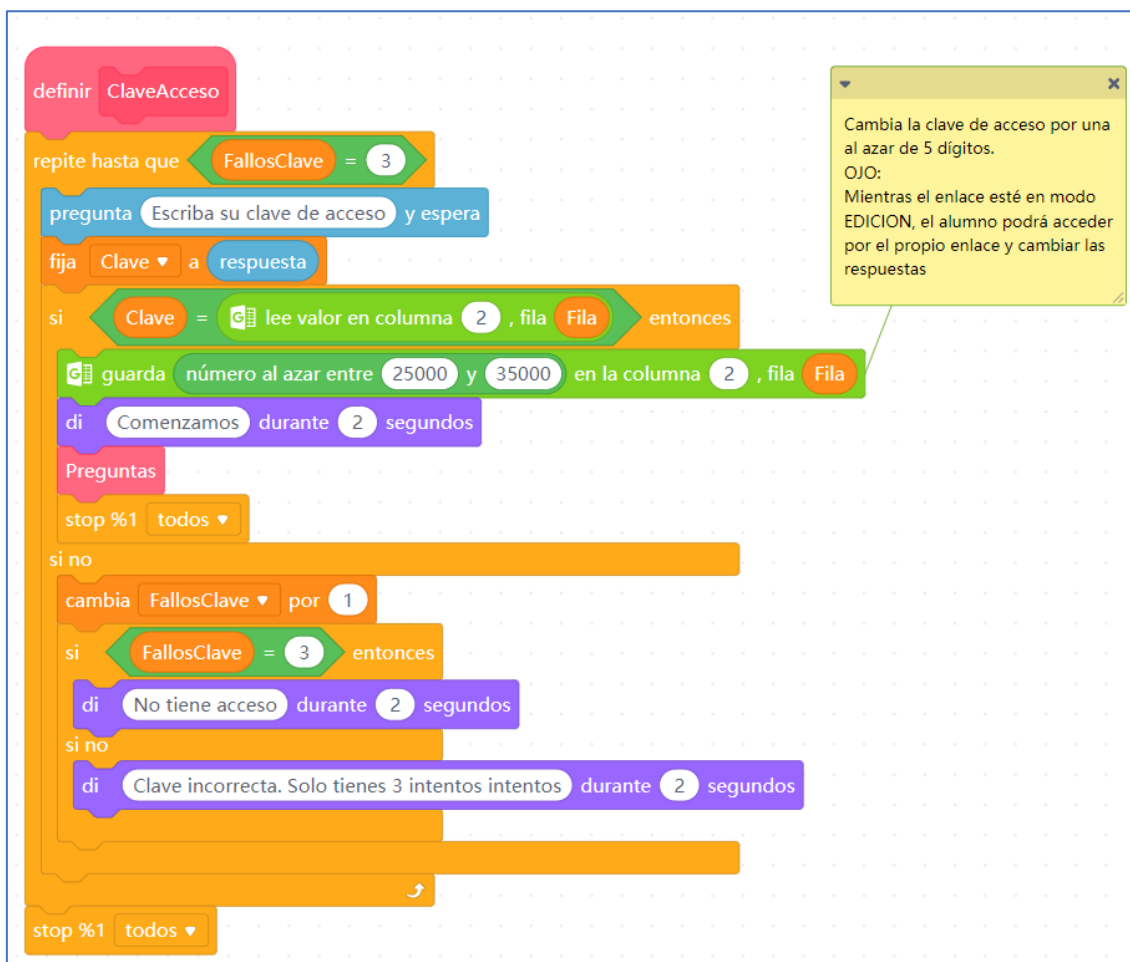
El programa modificado con la clave sería el siguiente:



Como podemos ver en el pequeño script de la imagen anterior, ya hemos identificado la fila en donde escribirá el alumno y, por lo tanto, sabemos **qué dato relativo a la clave debe leer**. Pensar que cada alumno tiene su clave y cada alumno escribe en una fila.

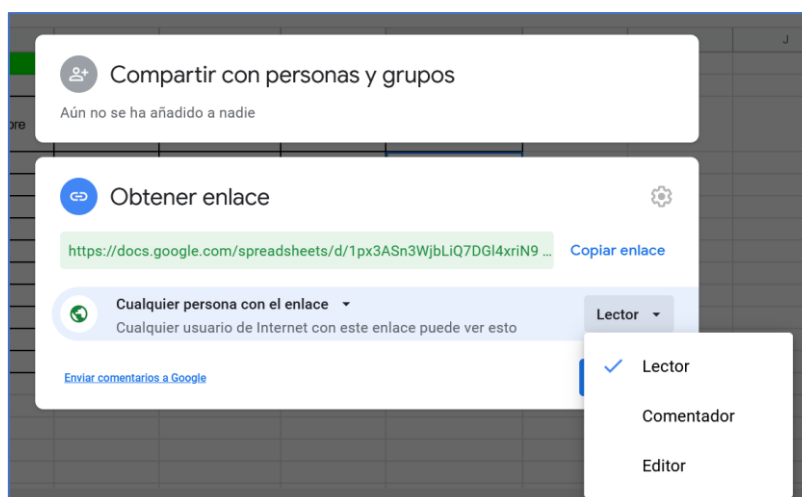
El subprograma **ClaveAcceso** sopesa la clave del alumno. Si el alumno introduce la clave correcta, se la cambia y accede a las preguntas que debe responder y quedan grabadas. El hecho de cambiar la clave de acceso sólo es una “fricada” ya que el enlace muestra por ahora un archivo editable.

En el caso de que el alumno/a introduzca mal la contraseña, el programa se lo dice y le deja probar 3 veces. Si falla, el programa se acaba y no puede cumplimentar las cuestiones a través de mBlock5.



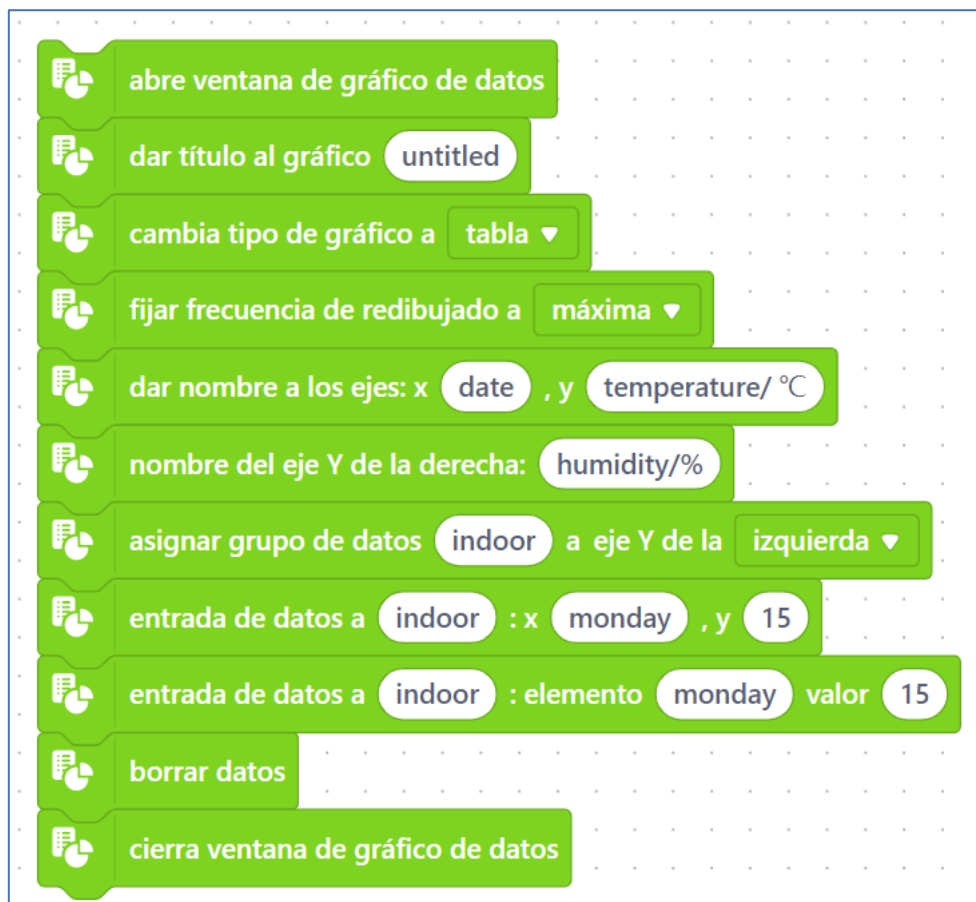
Aun a costa de repetirme es importante matizar que, en el subprograma se observa que sólo se accede al bloque de las preguntas si ingresa correctamente su clave. Clave que es leída de la columna correspondiente en la hoja de cálculo y que, inmediatamente, se cambiará por otra de 5 dígitos. Este cambio de clave no conseguirá que el alumno no pueda hacer trampa cambiando sus respuestas de forma directa accediendo al link de la hoja de cálculo. No olvidemos que lo hemos compartido en modo “Editor”.

La única forma de que no haga cambios es, tan pronto se ingresen los datos, volver a acceder al link y cambiarlo a modo “Lector”:



2.3.2. Gráfico de datos

Esta extensión la podemos encontrar dentro del bloque de gestión de datos de ciencia. Su utilidad es graficar información. Para ello han desarrollado los comandos que se muestran en la siguiente imagen.



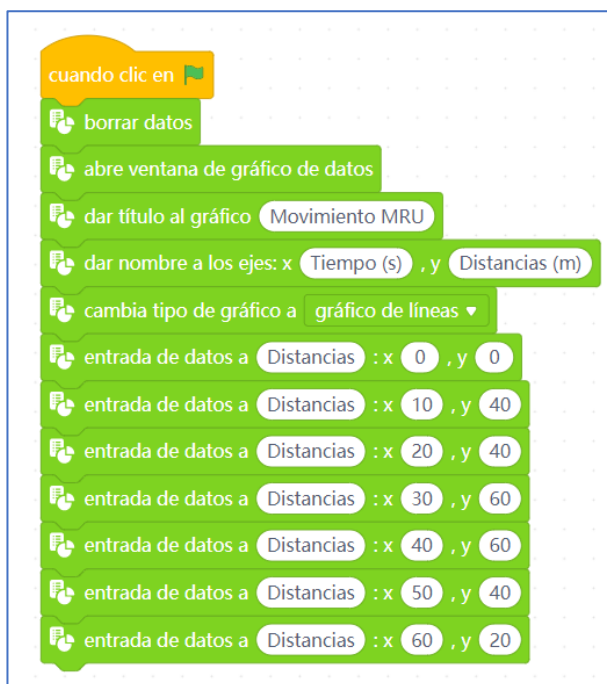
Esta extensión presenta las siguientes características:

- Los datos los podemos importar a través de un archivo de extensión .csv
- Obviamente, los datos también los podemos introducir o recoger en tiempo real utilizando los comandos de otras extensiones o genéricos de mBlock como pueden ser a través de listas.
- Podemos ver los datos introducidos en forma de tablas de datos y por supuesto, en forma de gráficos.
- Los gráficos que implementa son: gráficos de líneas, gráficos de barras, gráficos de doble eje o gráficos de sectores.
- En número máximo de datos que muestra son 500.
- El número máximo de grupos de datos es de 15 y cada grupo puede incluir un máximo de 300 datos.

La mejor forma de entenderlo es con ejemplos.

Reto 1: Graficar un MRU

Vamos a comenzar con un reto simple para graficar un movimiento rectilíneo y uniforme. Tras entrar en mBlock5 y añadir la extensión a nuestro objeto, ingresemos el siguiente programa:



Inicialmente borramos los datos para asegurarnos que la extensión sólo gestionará los nuevos datos que le vamos a introducir.

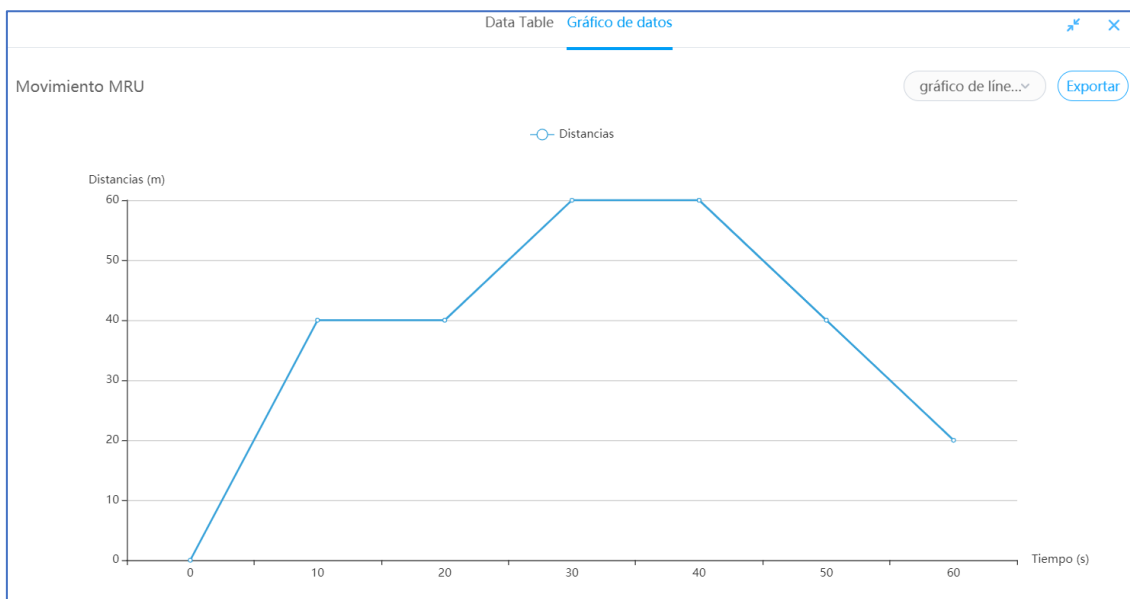
Definimos el nombre de la gráfica así como, el de sus ejes (X e Y).

Elegimos el tipo de gráfica que debe crear a partir de los datos (en este ejemplo usamos el gráfico de líneas) y finalmente introducimos los datos.

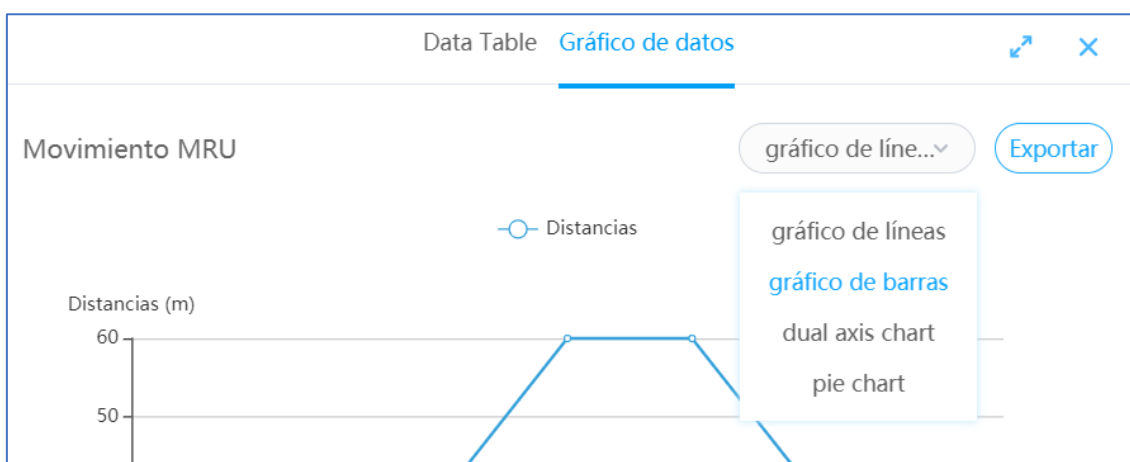
Tras hacer clic en la bandera verde el programa comienza a ejecutarse y se nos abre una ventana de gráfico de datos que nos muestra los datos introducidos de forma secuencial en una tabla:

Data Table Gráfico de datos	
Movimiento MRU	
<div>Importar</div> <div>Exportar</div>	
Tiempo (s)	Distancias
0	0
10	40
20	40
30	60
40	60
50	40
60	20

Para visionar el gráfico sólo debemos hacer clic en la pestaña “Gráfico de datos”:



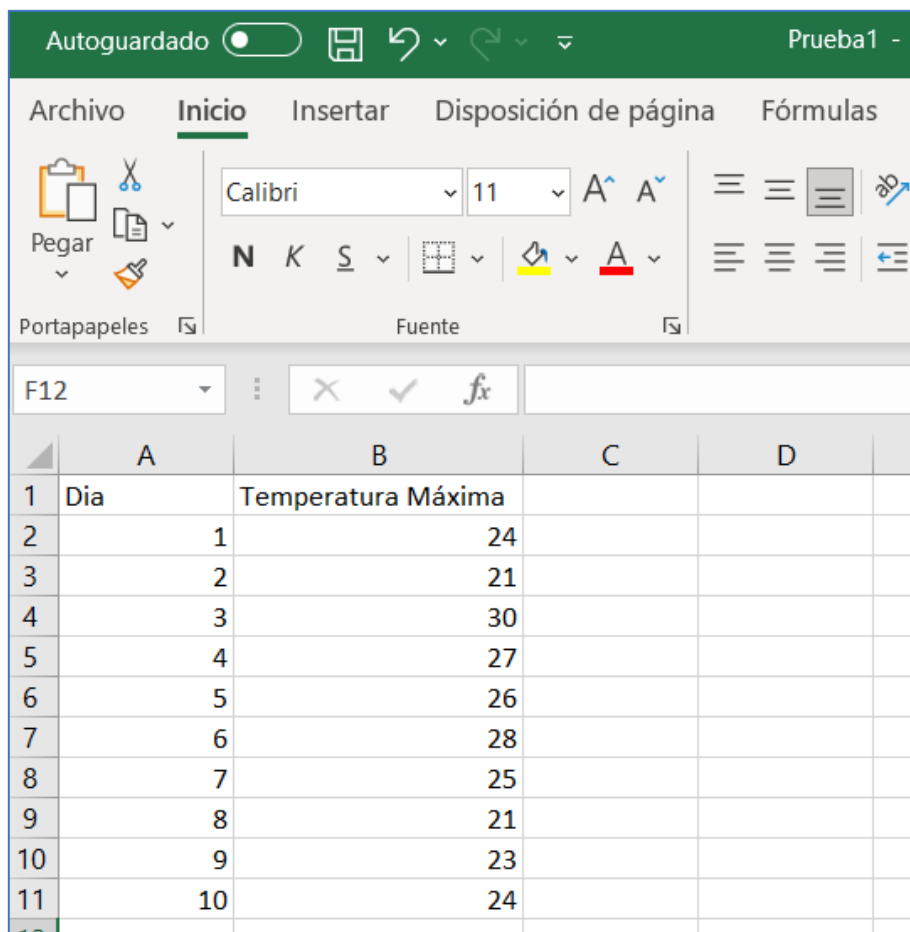
Aunque específicamente en el programa le hemos dicho al software que genere este tipo de gráfico ya que es lo que mejor se ajusta a la visibilidad de estos datos, podríamos cambiarlo y ver otro tipo de gráficos sin necesidad de comenzar de cero. Simplemente debemos escoger otro tipo de diagrama como por ejemplo, el diagrama de barras:



Reto 2: Exportar gráfico o datos e importar datos

Una opción muy interesante es la relativa a la exportación, activa tanto para el gráfico como la tabla de datos y se nos exportaría como un archivo [png para los gráficos](#) y un archivo [.csv para la tabla de datos](#).

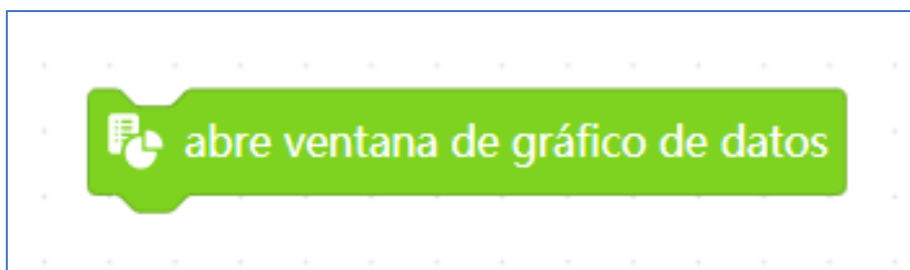
Supongamos que ahora queremos importar unos datos que tenemos en una hoja de cálculo. Obviamente, este archivo debemos guardarlo con la extensión .csv. En mi caso, el archivo se llama "Prueba1".



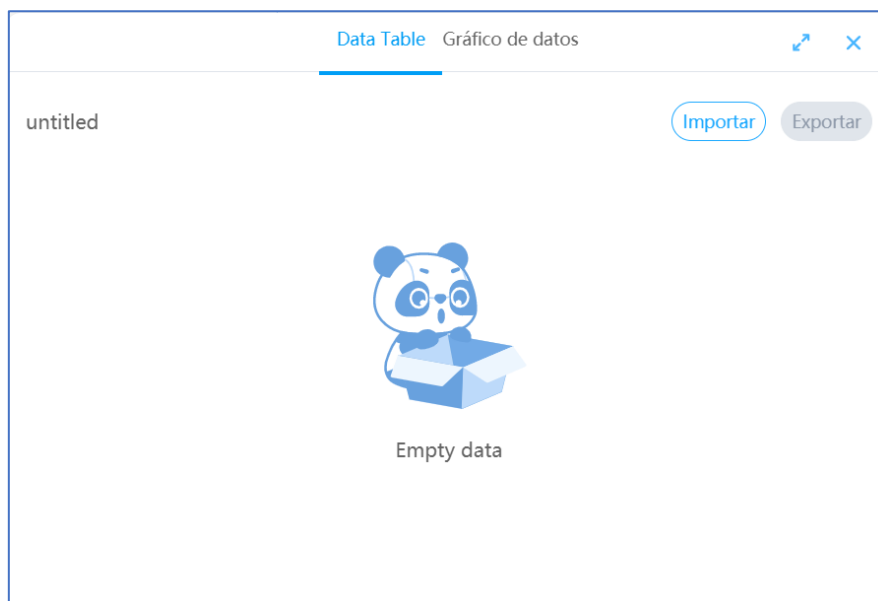
The screenshot shows the MBlock5 interface with a spreadsheet titled "Prueba1". The spreadsheet has columns A, B, C, and D. Column A is labeled "Dia" and contains numbers 1 through 10. Column B is labeled "Temperatura Máxima" and contains temperature values: 24, 21, 30, 27, 26, 28, 25, 21, 23, and 24. The interface includes a menu bar with "Archivo", "Inicio", "Insertar", "Disposición de página", and "Fórmulas". The "Inicio" menu is active, showing options for font (Calibri, size 11), bold (N), italic (K), underline (S), and text color (A). The spreadsheet is currently showing rows 1 through 11.

	A	B	C	D
1	Dia	Temperatura Máxima		
2	1	24		
3	2	21		
4	3	30		
5	4	27		
6	5	26		
7	6	28		
8	7	25		
9	8	21		
10	9	23		
11	10	24		

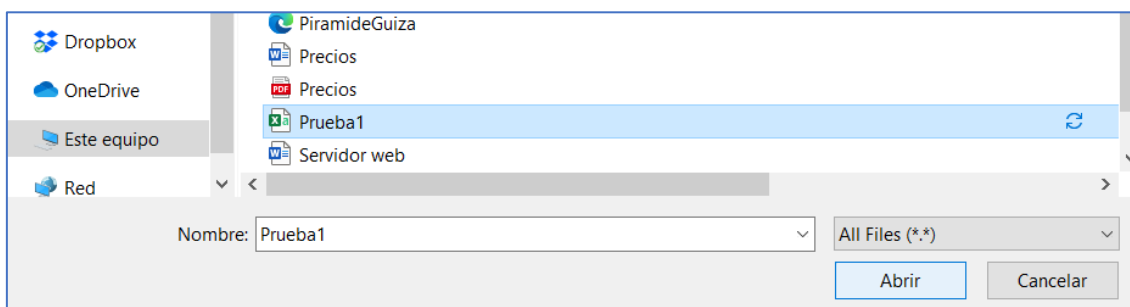
Para importarlos hacemos clic en el siguiente comando:



Se nos abre la siguiente ventana que nos muestra que no disponemos de datos. Vamos a importar los datos haciendo clic en el botón Importar

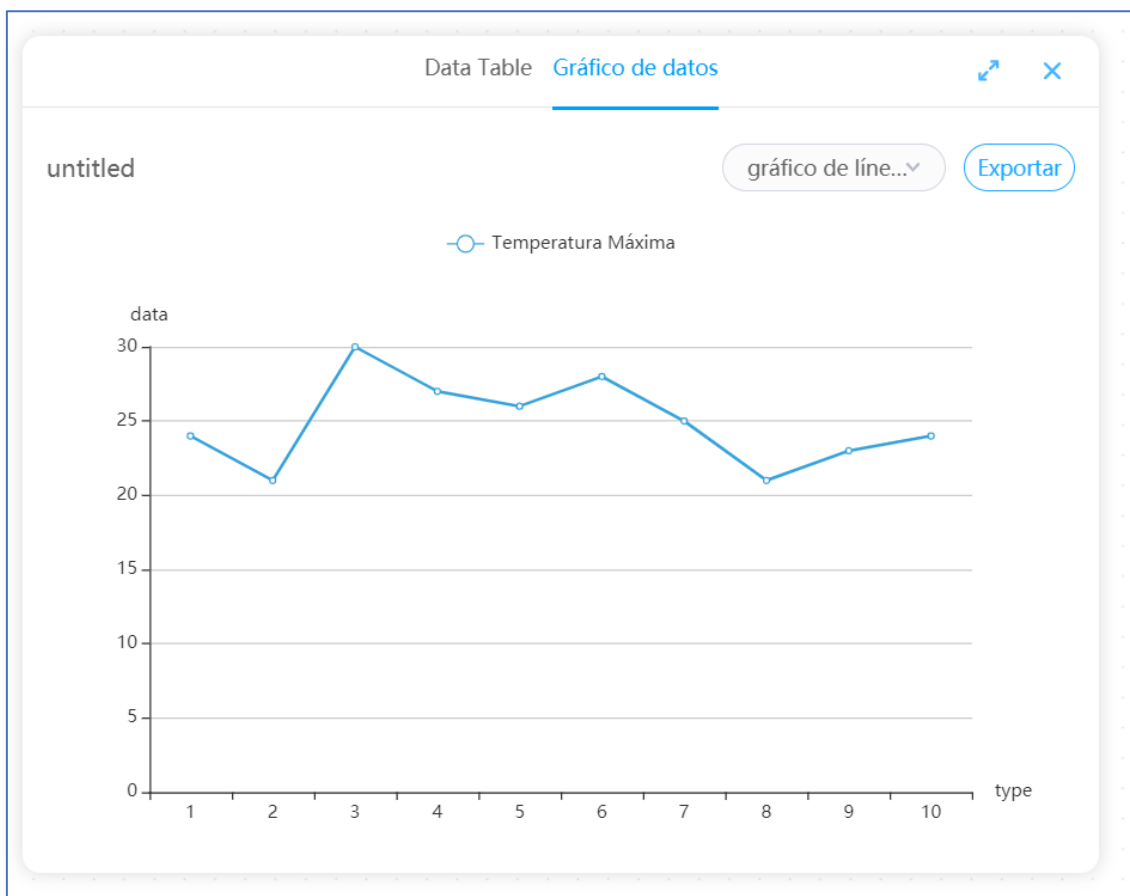


Seleccionamos el archivo “Prueba1” que es en donde hemos guardado los datos:

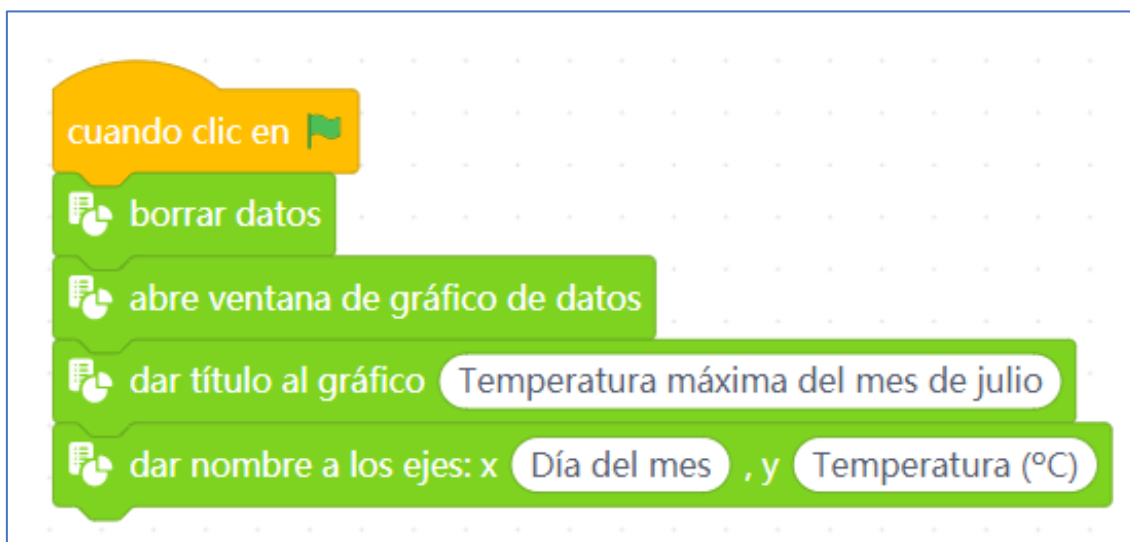


Podemos observar que se nos han guardado y ya podemos graficarlos:

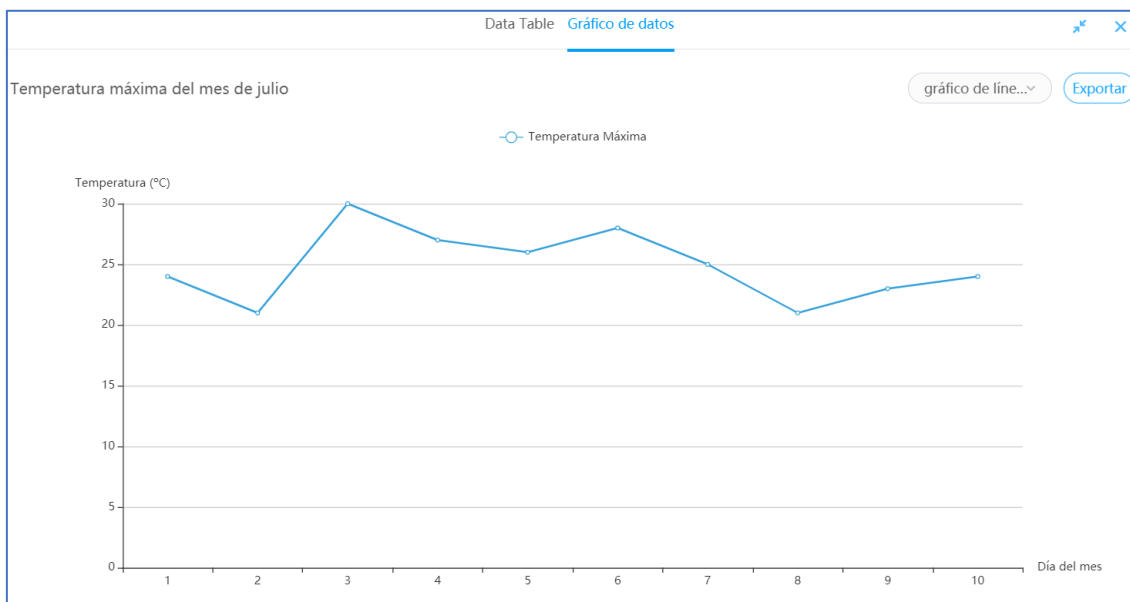
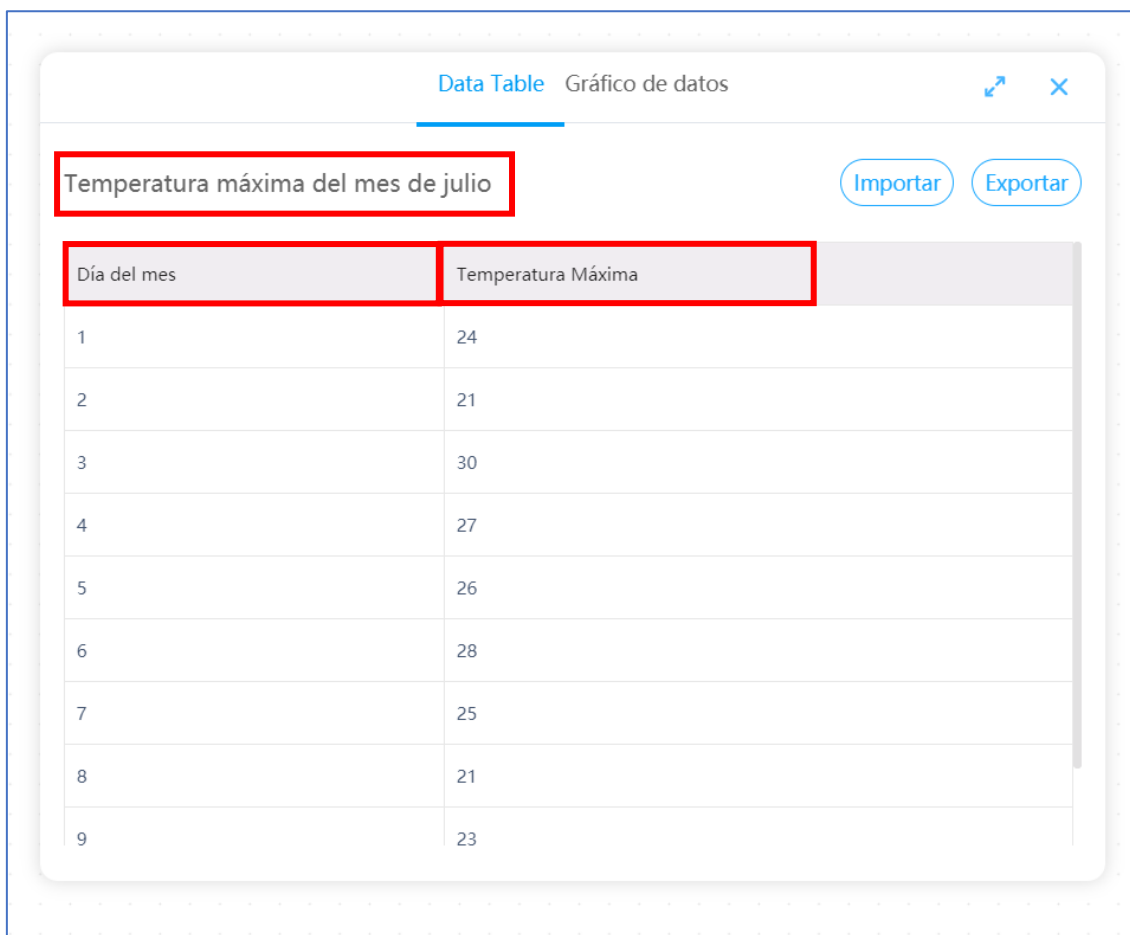
type	Temperatura Máxima
1	24
2	21
3	30
4	27
5	26
6	28
7	25
8	21
9	23



Podemos mejorarlo si creamos antes este programa y volvemos a subir el archivo Prueba1.csv:

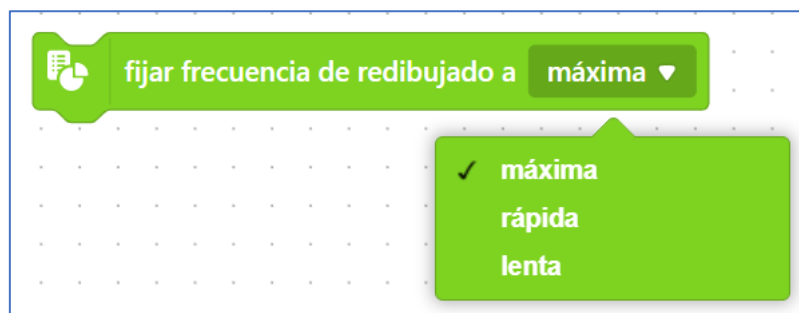


El resultado es el siguiente:



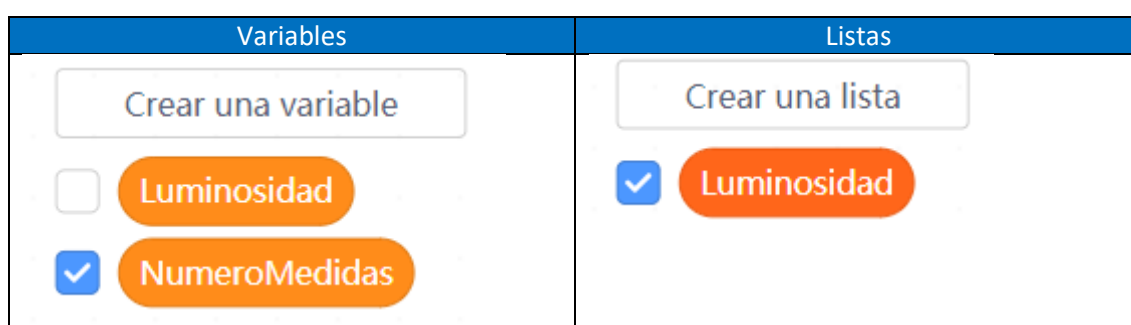
Reto 3: Graficar a tiempo real mediciones de un sensor

Pero, lo genial de esta extensión no es la aplicación que hemos visto hasta ahora. Pensar que podemos grabar datos a tiempo real en una hoja de cálculo, que esos datos sean recogidos por mBlock5 y graficarlos. De ahí que esta extensión disponga del siguiente comando para refrescar los datos y recogerlos.



Con el siguiente reto buscamos graficar 15 medidas de luminosidad de nuestro hogar. Yo voy a usar el robot mBot pero podríamos hacer lo mismo con cualquier otro dispositivo o robot que se conecte a mBlock5.

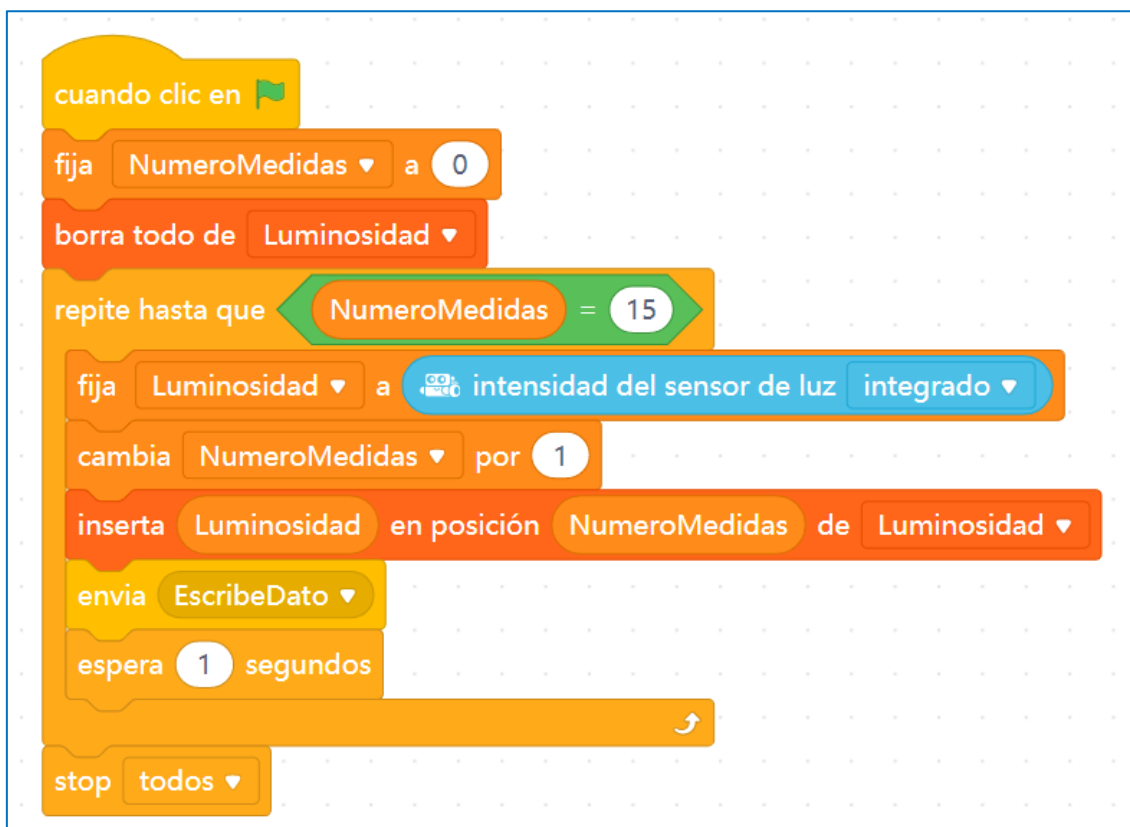
Después de conectar el robot comenzamos a crear nuestro programa en el apartado dispositivos. Necesitamos crear 2 variables y una lista:



Nuestro robot realizará 15 medidas de la luminosidad del hogar. Las medidas se ingresarán en la lista “Luminosidad” cada segundo y se enviarán a tiempo real al objeto “PandaGraficar” en el cual hemos descargado la extensión “Gráfico de datos”:



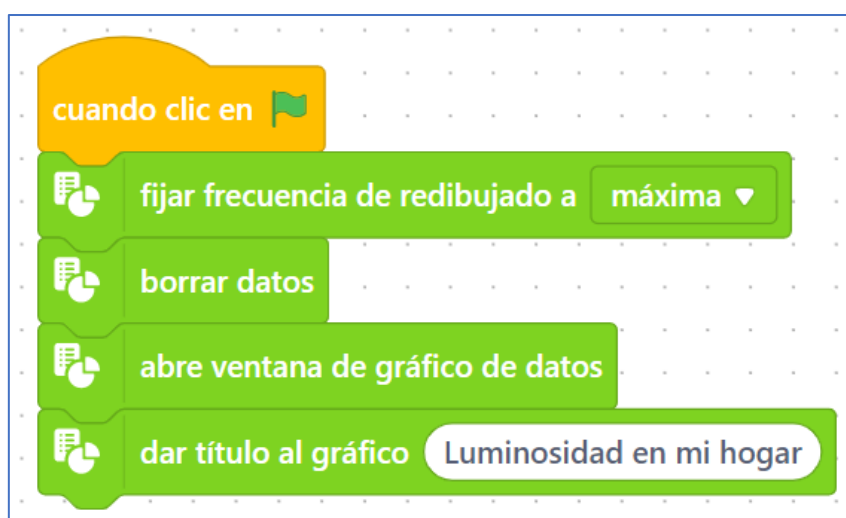
Los programas, tanto para en el dispositivo mBot como para el Sprite “PandaGraficar” son:

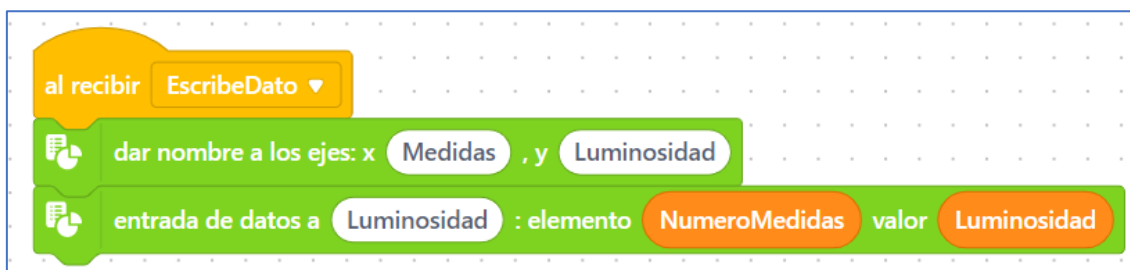
*Script para el dispositivo*

Al accionar la bandera verde la variable “**NumeroMedidas**” se inicializa al valor 0 y se borran los datos de la lista “**Luminosidad**” para asegurarnos que sólo graficaremos los nuevos datos que va a grabar.

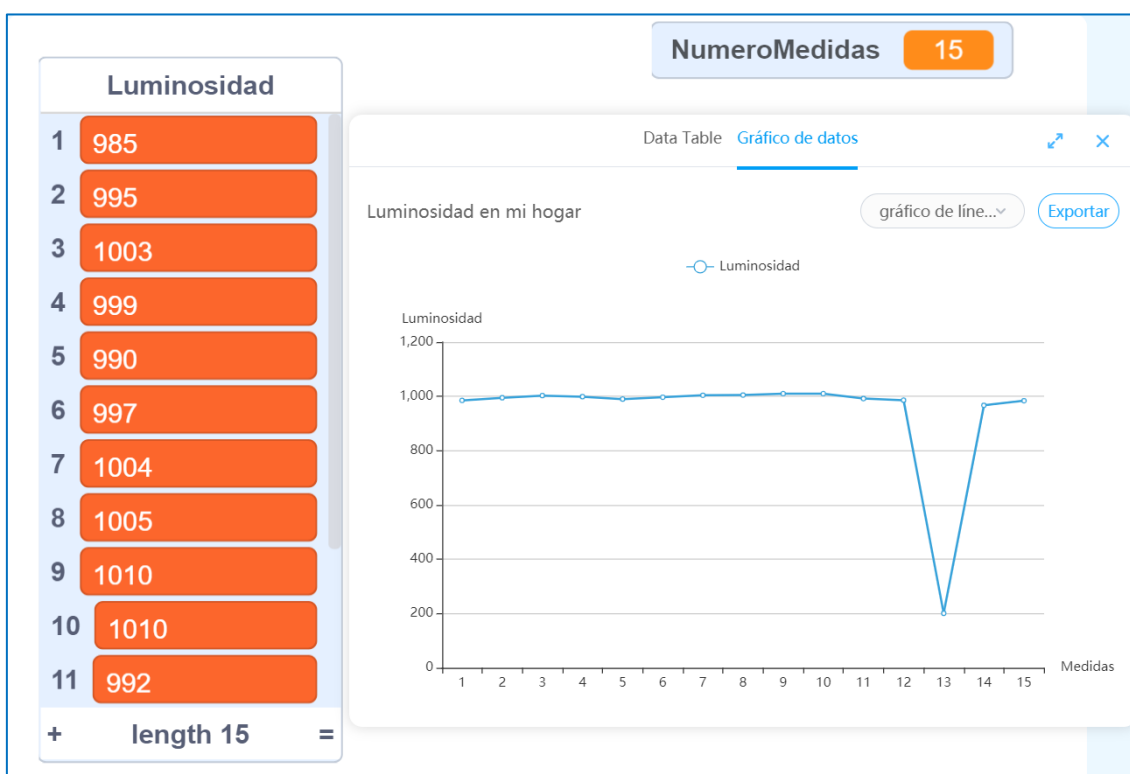
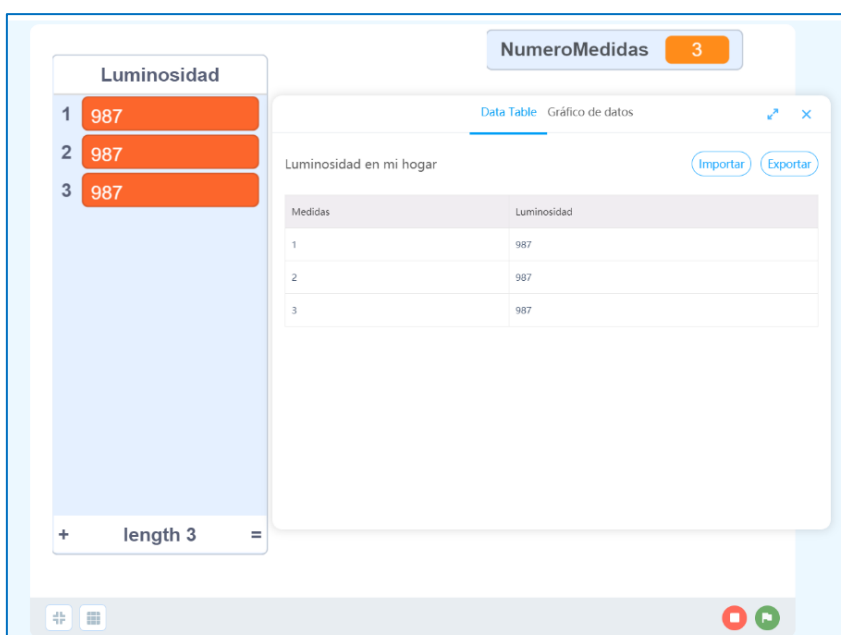
Creamos un bucle que sólo se ejecutará 15 veces y en él, tras recoger el dato e incluirlo en la lista “**Luminosidad**” lo envía al objeto con el mensaje “**EscribeDato**”. Tras pasar 1 segundo, vuelve a recoger un nuevo dato.

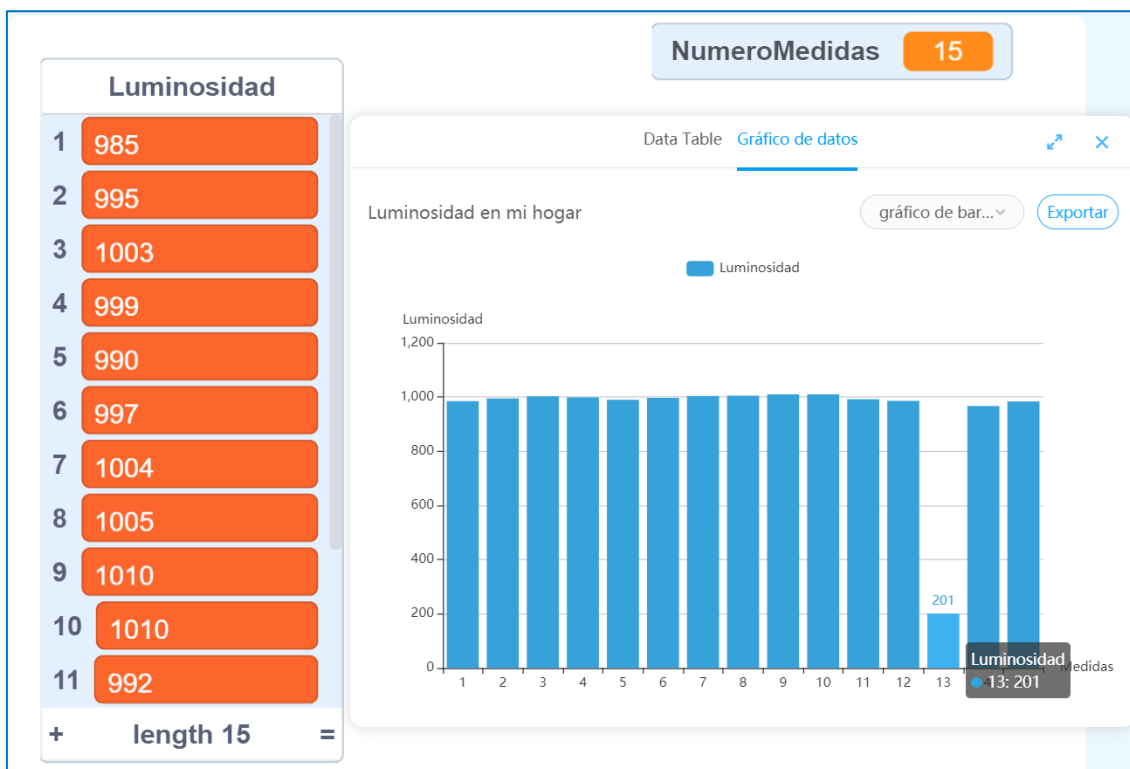
Cuando el objeto recibe el mensaje “**EscribeDato**” introduce en la tabla de datos a tiempo real el valor del número de medida y de su luminosidad.

*Script para el objeto (Parte1)*



Script para el objeto (Parte2)





Reto 4: Graficar dos grupos de datos para un mismo eje x

En este reto introduciremos dos grupos de datos relativos a las temperaturas máxima y mínima de Cambados en una semana de julio (en este caso son valores inventados). Los resultados de estas mediciones son los que se muestran en la siguiente tabla de valores:

Día de la semana	Temperatura Máxima	Temperatura Mínima
Lunes	22	18
Martes	25	20
Miércoles	27	22
Jueves	26	21
Viernes	25	20
Sábado	24	21
Domingo	25	22

Introducimos estos datos en mBlock5 en el apartado objetos usando la extensión “Gráfico de Datos” y ejecutamos el programa:

Data Table Gráfico de datos

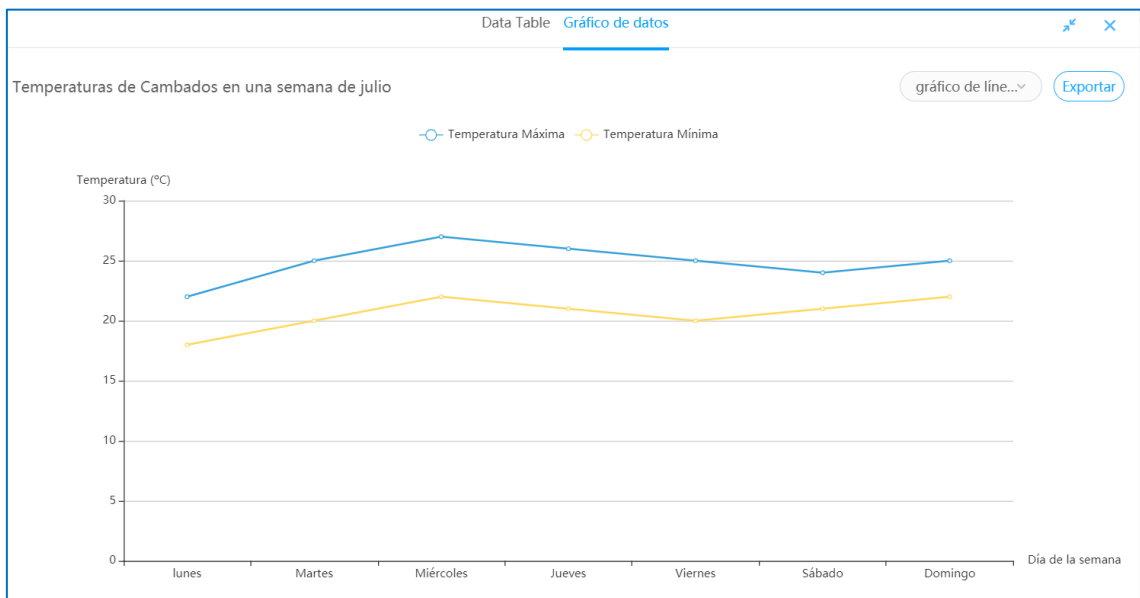
Temperaturas de Cambados en una semana de julio

Día de la semana	Temperatura Máxima	Temperatura Mínima
Lunes	22	18
Martes	25	20
Miércoles	27	22
Jueves	26	21
Viernes	25	20
Sábado	24	21
Domingo	25	22

Script:

```

cuando clic en
  borrar datos
  abrir ventana de gráfico de datos
  dar título al gráfico Temperaturas de Cambados en una semana de julio
  cambiar tipo de gráfico a gráfico de líneas
  entrada de datos a Temperatura Máxima : x Lunes : y 22
  entrada de datos a Temperatura Máxima : x Martes : y 25
  entrada de datos a Temperatura Máxima : x Miércoles : y 27
  entrada de datos a Temperatura Máxima : x Jueves : y 26
  entrada de datos a Temperatura Máxima : x Viernes : y 25
  entrada de datos a Temperatura Máxima : x Sábado : y 24
  entrada de datos a Temperatura Máxima : x Domingo : y 25
  entrada de datos a Temperatura Mínima : x Lunes : y 18
  entrada de datos a Temperatura Mínima : x Martes : y 20
  entrada de datos a Temperatura Mínima : x Miércoles : y 22
  entrada de datos a Temperatura Mínima : x Jueves : y 21
  entrada de datos a Temperatura Mínima : x Viernes : y 20
  entrada de datos a Temperatura Mínima : x Sábado : y 21
  entrada de datos a Temperatura Mínima : x Domingo : y 22
  
```



Reto 5: Implementa un gráfico de doble eje

El gráfico de doble eje es un tipo de gráfico especial que se puede utilizar para mostrar varios grupos de datos. Los datos de estos grupos de datos pueden registrarse en diferentes unidades, pero pueden compartir el mismo eje x.

Vamos a reutilizar el ejemplo anterior y aumentarlo con un nuevo grupo de datos llamado "Humedad" sobre el mismo eje x (días de la semana).

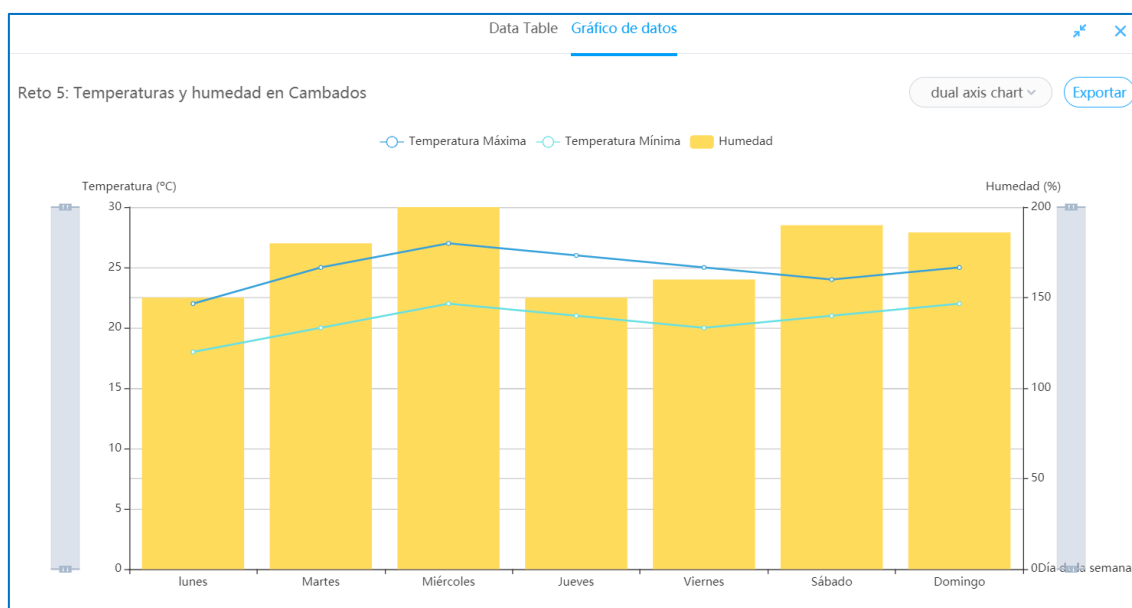
Día de la semana	Temperatura Máxima	Temperatura Mínima	Humedad
Lunes	22	18	150
Martes	25	20	180
Miércoles	27	22	200
Jueves	26	21	150

Viernes	25	20	160
Sábado	24	21	190
Domingo	25	22	186

Añadimos al programa anterior el siguiente código:



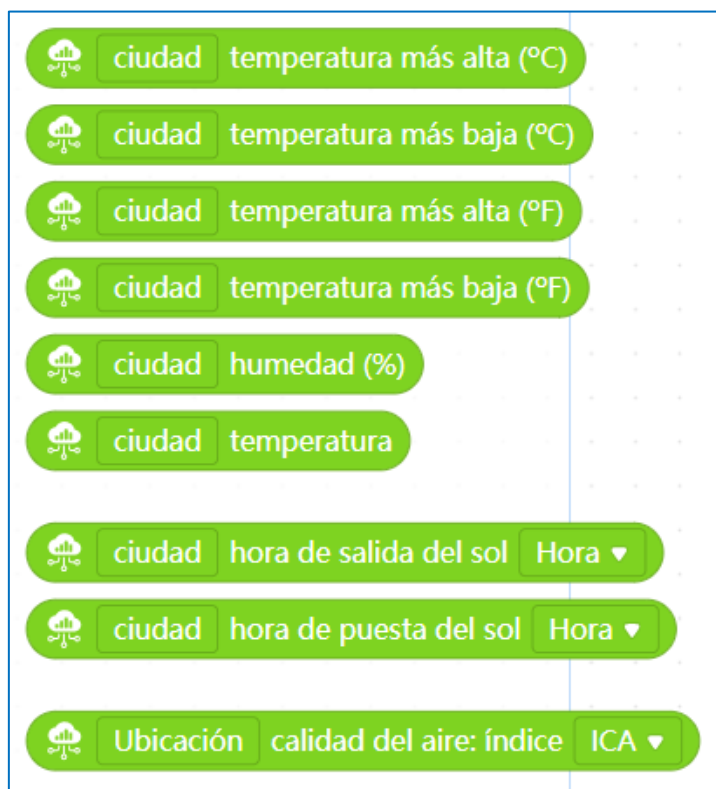
Y como resultado obtenemos el gráfico deseado de dos ejes:



2.3.3. Datos de clima

Esta extensión nos ofrece un conjunto de bloques que nos permitirá conocer e incluir en nuestra programación la temperatura tanto máxima como mínima, humedad, la hora de salida y puesta del sol en una ciudad y también la calidad del aire. Este último parámetro depende de unas ubicaciones preestablecidas.

La extensión es muy fácil de usar. Presenta los siguientes comandos:



Vamos a usarla para conocer estos datos climáticos en Cambados. Un posible escenario sería el siguiente:



Haciendo clic en los dos objetos Sol, los dos Botones de Temperatura y nubes obtenemos los datos programados para Cambados.

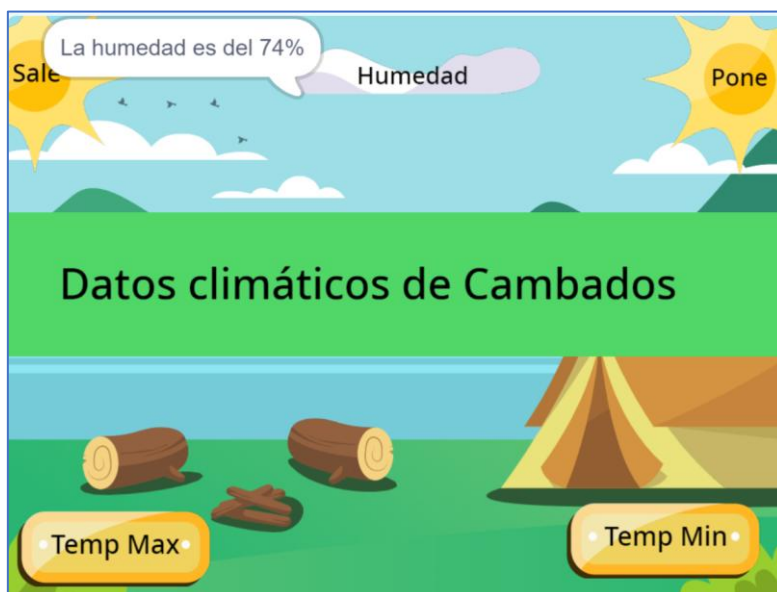
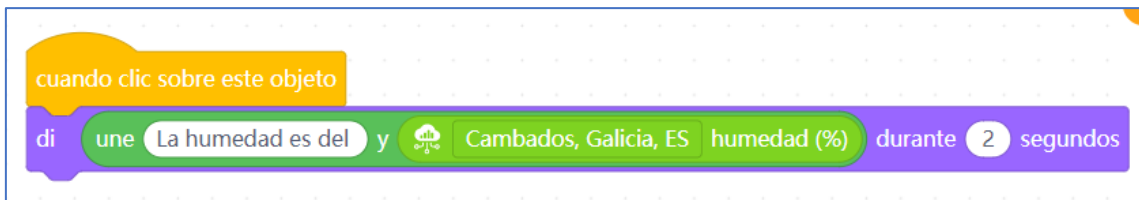
Para programar a qué hora y minuto se pone el sol en “Cambados” creo el siguiente programa en el objeto “SolPoniendose”. De la misma forma se programaría el objeto “SolSaliendo”:



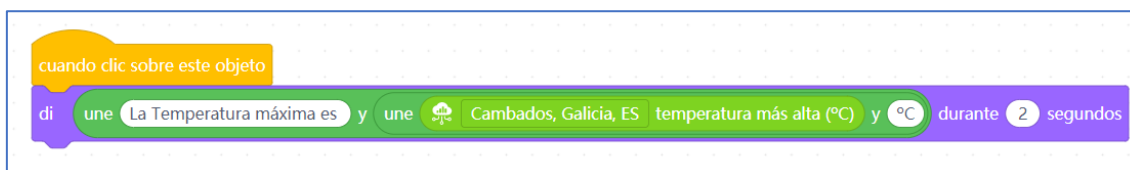
Tras hacer clic en los objetos Sol se nos mostraría la hora y minuto:



Para programar la humedad uso el objeto nubes y en é escribo el siguiente programa:



Finalmente, programamos ambos botones para la temperatura máxima y mínima. Uno de sus scripts es el siguiente:



Cuyo resultado es:



2.4. IoT o Internet de las Cosas

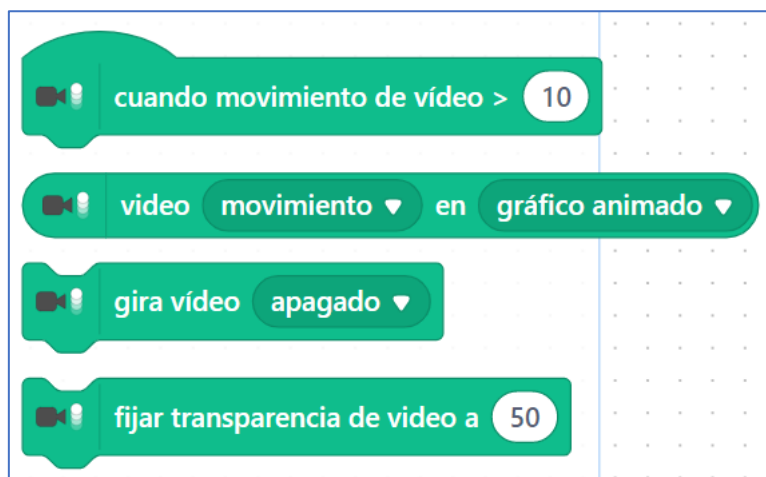
Cuando hablamos de Internet de las Cosas pensamos en un mundo interconectado, un mundo en donde los sensores perciben y están conectados entre si intercambiando datos a través de internet. Nos centraremos en las tres primeras extensiones que observamos en la siguiente imagen:



2.4.1. Video Sensing

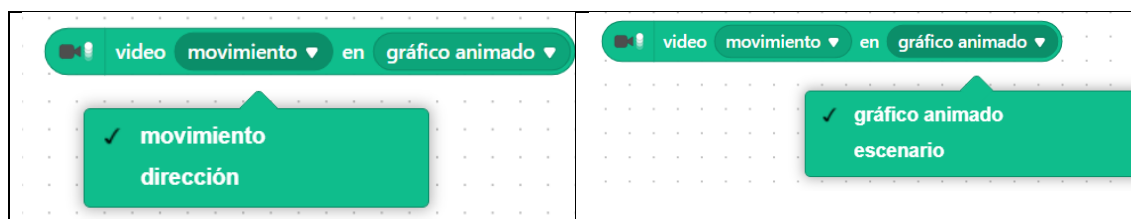
Video Sensing pertenece al grupo de extensiones clasificadas en el campo de IoT o Internet de las Cosas. Esta extensión nos permite interactuar con un objeto basándonos en el movimiento que in situ detecta la webcam.

Los comandos de la extensión son los siguientes:



El primer comando permite que el sprite comience a ejecutar una acción cuando la cámara web detecte un movimiento que supere un valor numérico (por defecto el valor es 10, pero obviamente podemos cambiarlo).

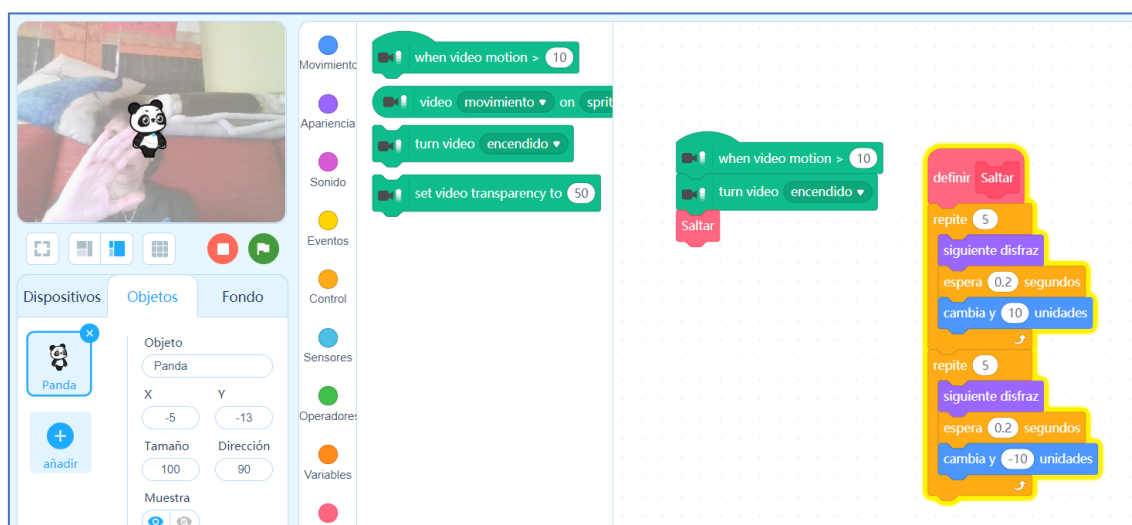
Con el comando **Vídeo movimiento/dirección sobre el objeto/escenario** podemos decidir si el movimiento es en el escenario o en nuestro personaje.



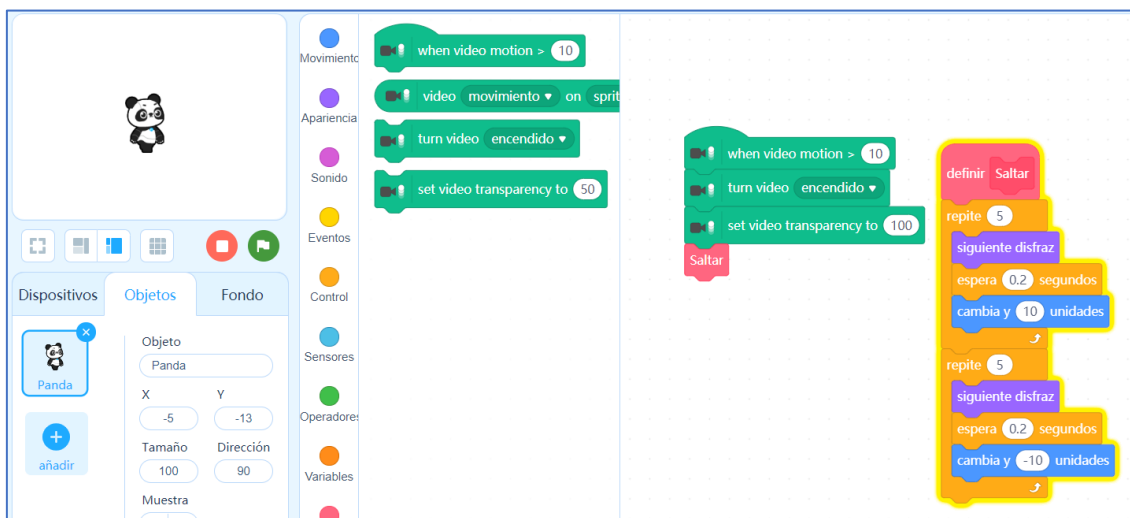
Con el bloque **Cambiar vídeo** podemos encender o apagar la cámara web e inclusive invertir la imagen.

En el siguiente ejemplo, si nos fijamos en el escenario, allí se observa que estoy moviendo la mano. Como he fijado la detección a un valor bajo (el 10 de defecto), se ejecutará la acción de “saltar”.

En la imagen siguiente la transparencia del vídeo está a 50 (que es su valor por defecto).



Si no quisiéramos que en el escenario se viera la imagen que detecta la webcam sólo deberíamos poner la transparencia a 100. Obviamente, la webcam seguiría detectando el movimiento.



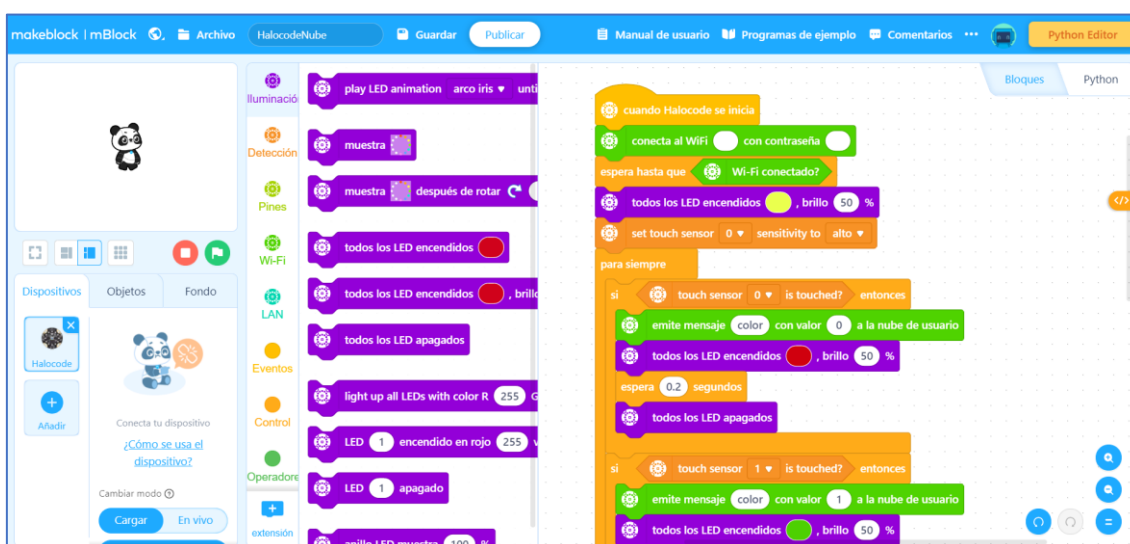
2.4.2. Mensaje en la nube de usuario

Con esta extensión podemos intercambiar datos entre dispositivos usando la nube de mBlock5.

En la versión de mBlock5 online podemos guardar nuestros programas en su nube. Para ello debemos registrarnos y se nos pedirá un email y que formalicemos una contraseña.

Podemos generar datos y enviarlos a la nube de mBlock y, cuando los solicitemos mBlock nos devuelve esos datos.

Al probar esta extensión veo que solamente funciona con el dispositivo *Halocode* (que es la placa IoT de Makeblock) y por ese motivo no se explica en este tutorial.



2.4.3. Cargar en modo difusión

Esta extensión se creó para que un sprite interactúe con un dispositivo en **modo carga**. En mBlock5 los comandos de los sprites son independientes de los comandos de los dispositivos.

Esta extensión es muy importante ya que tras cargar el programa en el dispositivo (placa Arduino, mBot, Ranger, etc) éste se comunicará con nuestro PC enviándole mensajes que, por supuesto, pueden incluir información de sensores. Veremos que la extensión “cargar en modo difusión” necesita de otra destinada a dispositivos llamada “Mensajes entre dispositivos” y que ambas se complementan:

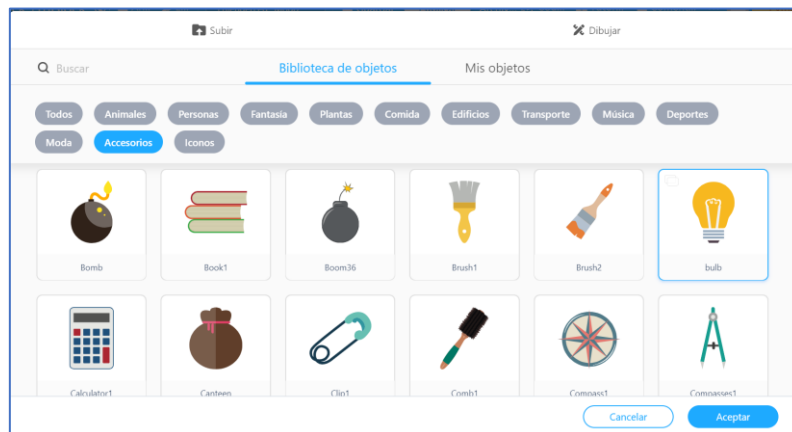
Extensión de Sprites u objetos	Extensión de dispositivos
 <p>Cargar en modo difusión By mBlock official   </p> <p>Al añadir esta extensión, permites que un dispositivo pueda interactuar con un gráfico animado en modo</p> <p>+ Añadir</p>	 <p>Mensajes entre disposit... By mBlock official  </p> <p>Al añadir esta extensión, puedes enviar mensajes entre dispositivos y gráficos animados para que</p> <p>+ Añadir</p>
<p>Comandos de la extensión “Cargar en modo difusión”</p>  <ul style="list-style-type: none"> envía mensaje de modo de carga <input type="text" value="message"/> envía mensaje de modo de carga <input type="text" value="message"/> con valor <input type="text" value="1"/> al recibir mensaje de modo de carga <input type="text" value="message"/> valor del mensaje de carga <input type="text" value="message"/> 	<p>Comandos de la extensión “Mensajes entre dispositivos”</p>  <ul style="list-style-type: none"> emite mensaje <input type="text" value="message"/> emite mensaje <input type="text" value="message"/> con valor <input type="text" value="1"/> al recibir mensaje <input type="text" value="message"/> valor del mensaje <input type="text" value=""/>

La mejor forma de entender estas extensiones es utilizándolas. Vamos a verlas con unos ejemplos.

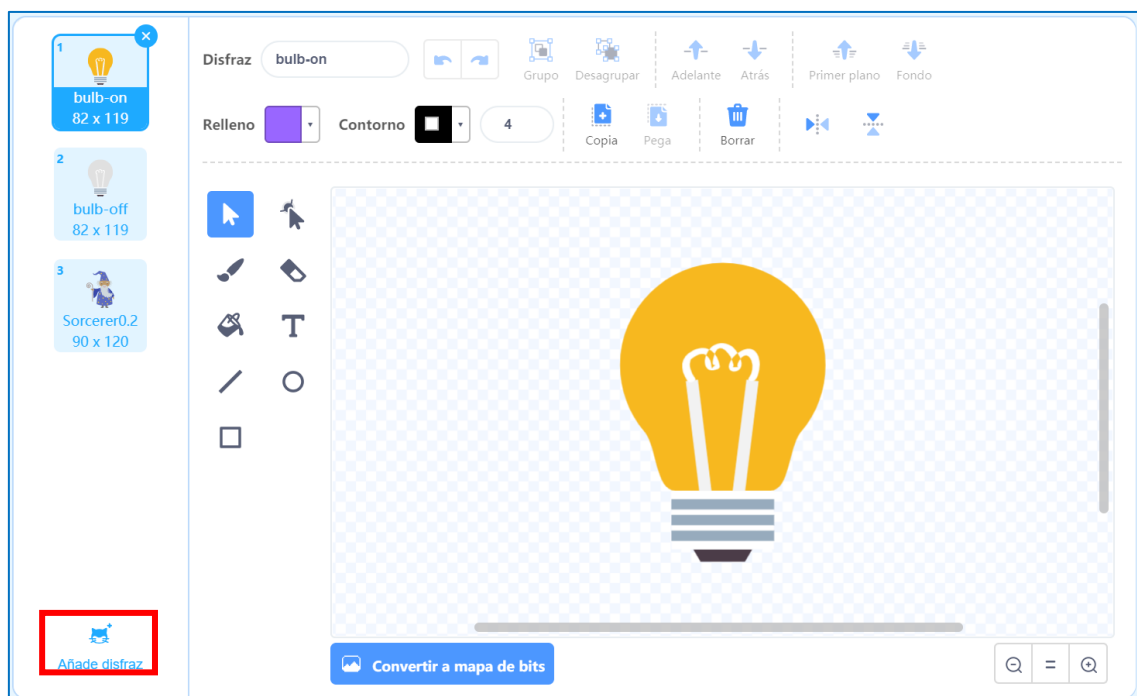
- **Reto 1:** Encender y apagar el led interno de la placa Arduino Uno usando como accionador el teclado de un PC.

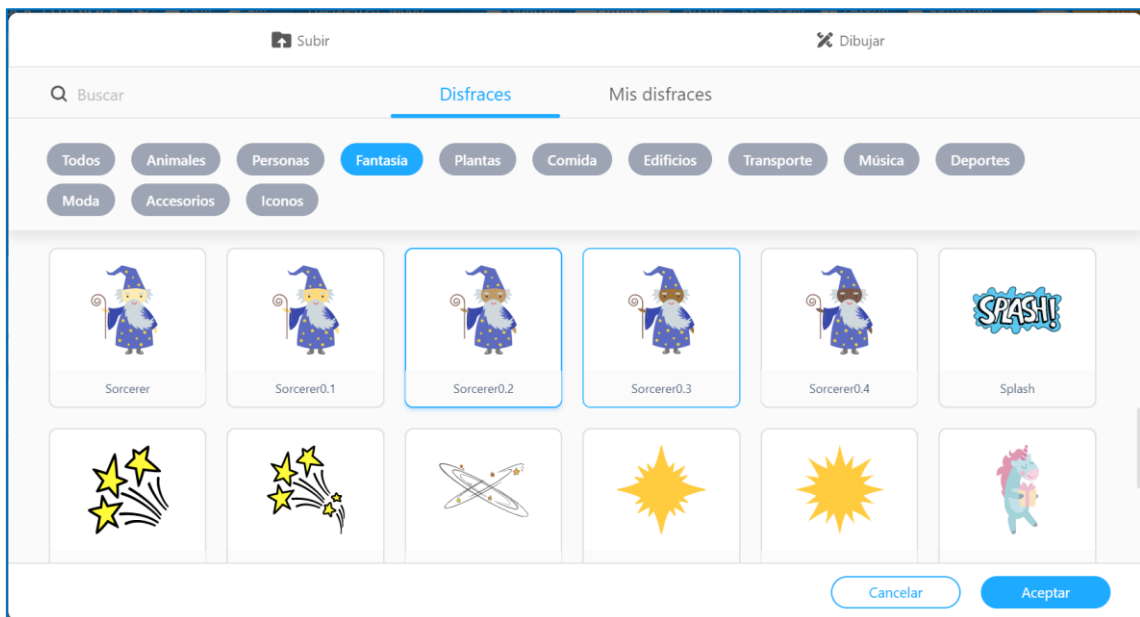
Abrimos un nuevo programa con mBlock5. Explicaremos de forma independiente los objetos, los dispositivos y los fondos.

Objetos: Eliminamos el objeto “Panda” e incluimos el objeto “bulb” que yo he renombrado como “Bombilla”:



Este objeto se compone de dos disfraces que también he renombrado como “bulb on” y “bulb off” y a mayores he añadido otro disfraz del bloque *Fantasia*.





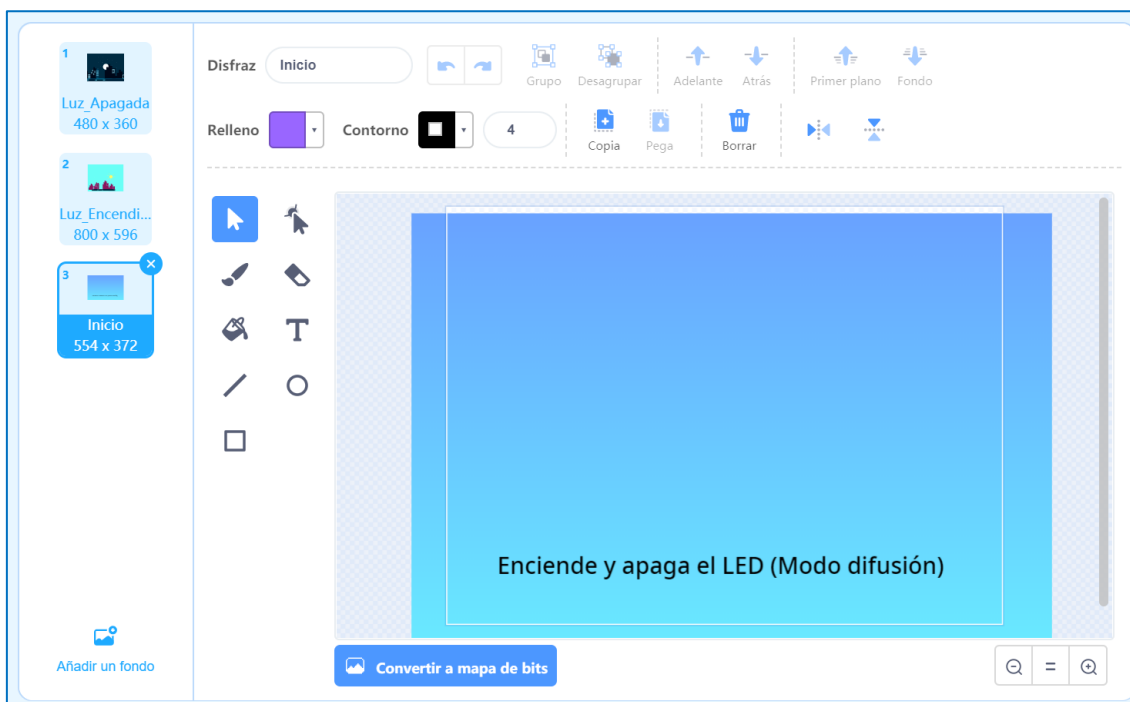
Pantalla del disfraz añadido

En ese objeto cargamos la extensión **“Cargar en modo difusión”**.

Dispositivos: Escogemos como dispositivo la placa Arduino Uno y en ella cargamos la extensión **“Mensajes entre dispositivos”**. Debemos fijarnos que siempre ha de estar en modo “carga”.



Fondos: Se compone de tres fondos que son “Inicio” (para el principio del programa) y los dos fondos que se intercambiarán cuando encendamos o apaguemos el LED con el teclado y que los llamo “Luz_Encendida” y “Luz_Apagada”:



Comenzamos la programación por el objeto “Bombilla”:

Al hacer clic en la bandera verde debe salirnos el mago para contarnos cómo podemos interactuar con el programa.

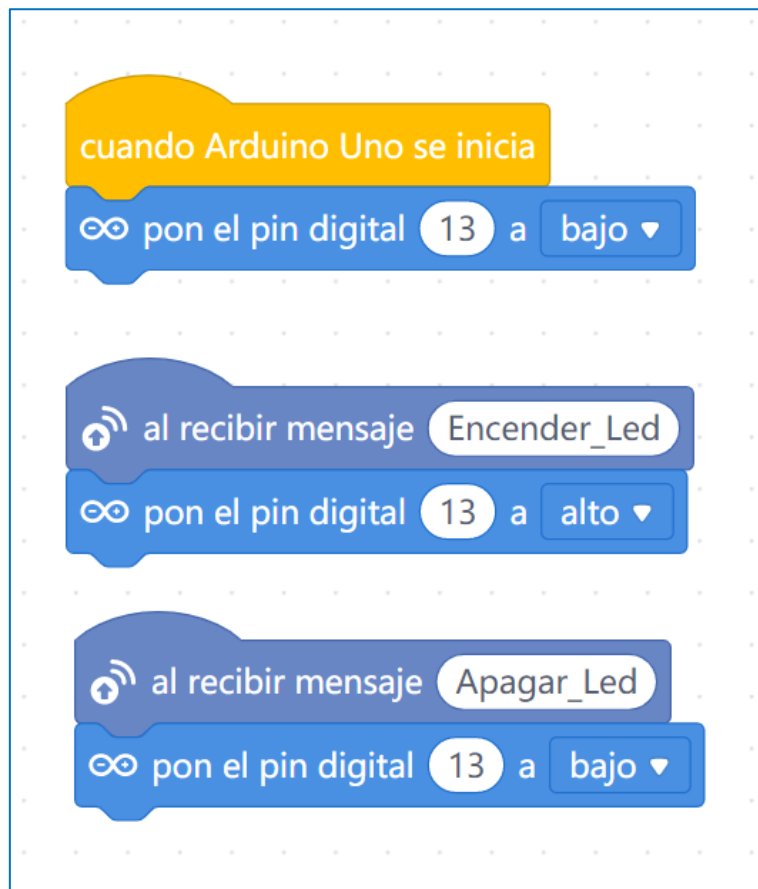
Si pulsamos la letra “e” se debe encender el LED y por ese motivo se envía el mensaje en carga “Encender_Led”, pero si pulsamos la tecla “a” se debe apagar y esto se consigue enviando el mensaje en carga “Apagar_Led”.

Todo envío presupone una recepción de alguien. En nuestro caso, la placa Arduino Uno.



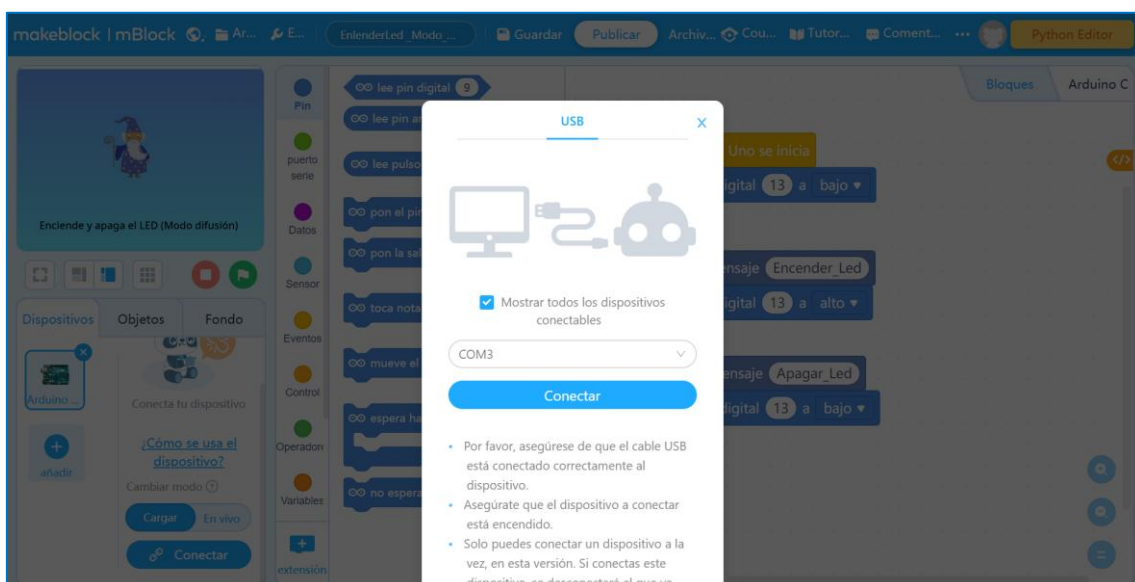
Programación del objeto Bombilla

El dispositivo Arduino Uno recibirá los mensajes y procederá a encender y apagar su LED interno:



Programación del dispositivo Arduino Uno

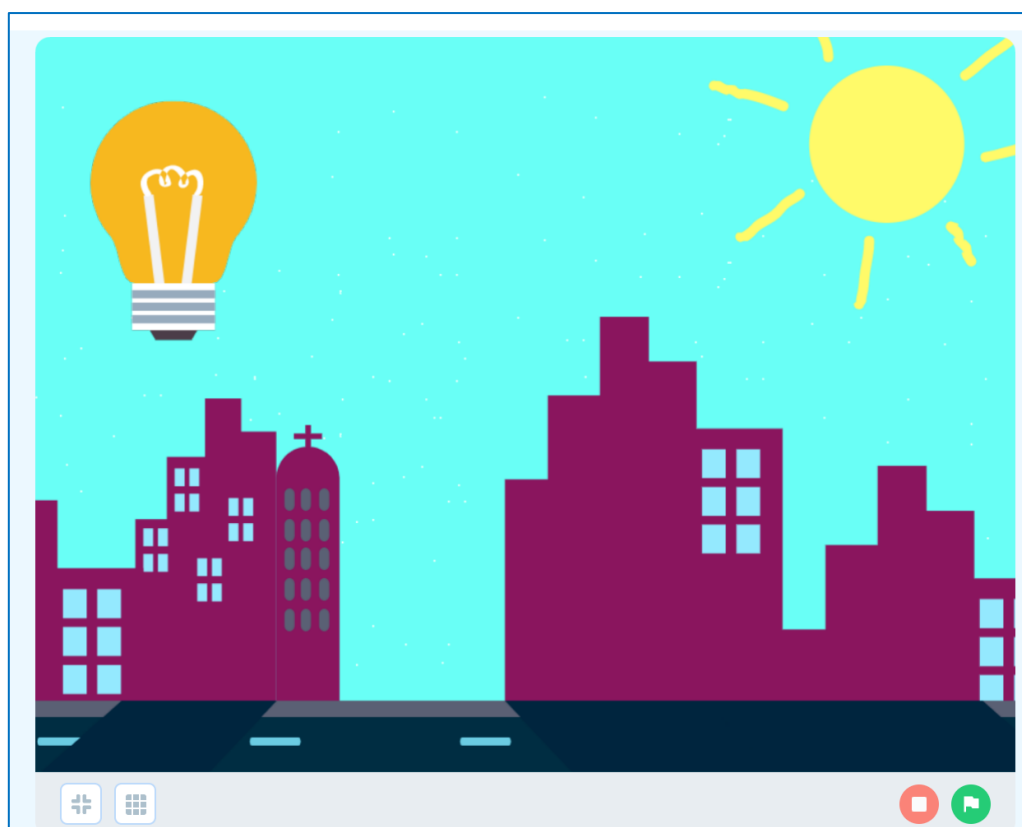
Conectamos la placa Arduino Uno a mBlock5 y subimos el programa:



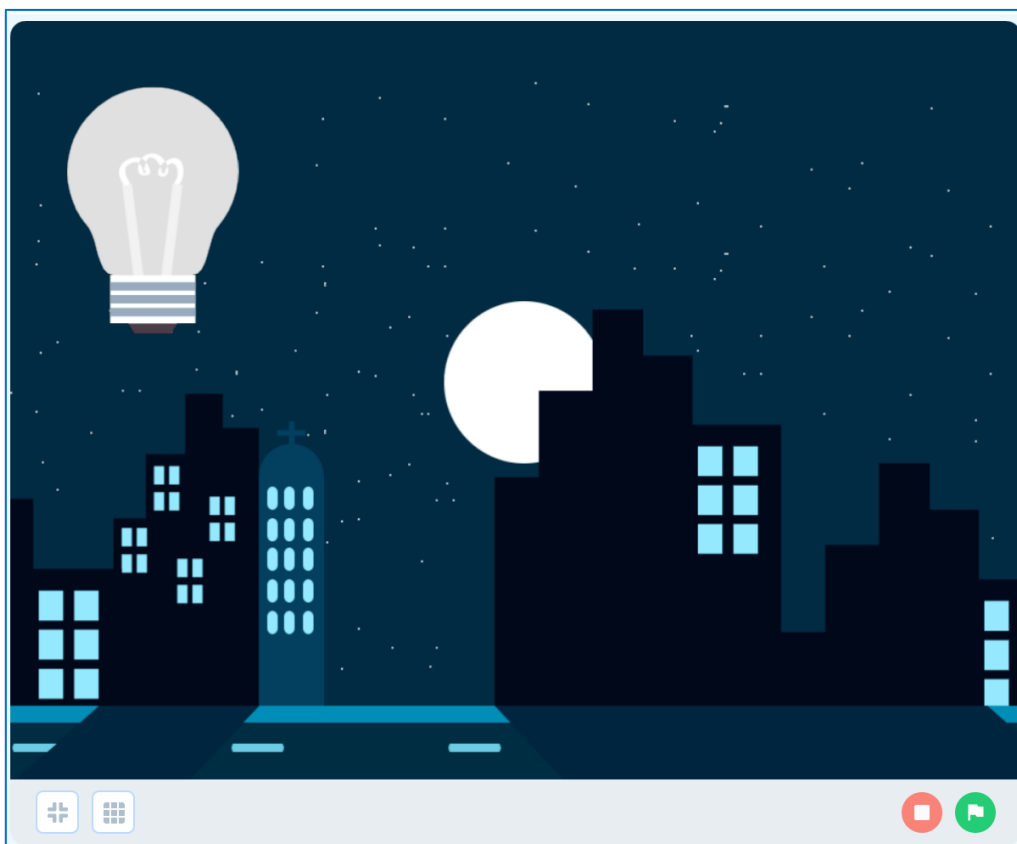
Tras hacer clic en la bandera verde comienza a interactuar el “mago”:



Tras pulsar la letra “e” del teclado se enciende el LED interno de la placa Arduino Uno y en el escenario de mBlock5 observamos que el fondo ha cambiado al igual que el disfraz del objeto “Bombilla”. Se simula el encendido:

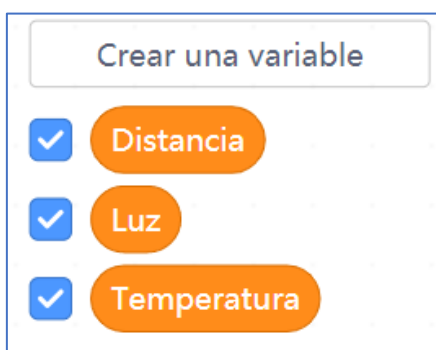


Tras pulsar la letra “a” observamos que el LED interno se apaga y el escenario y disfraz de la “Bombilla” simula la oscuridad:

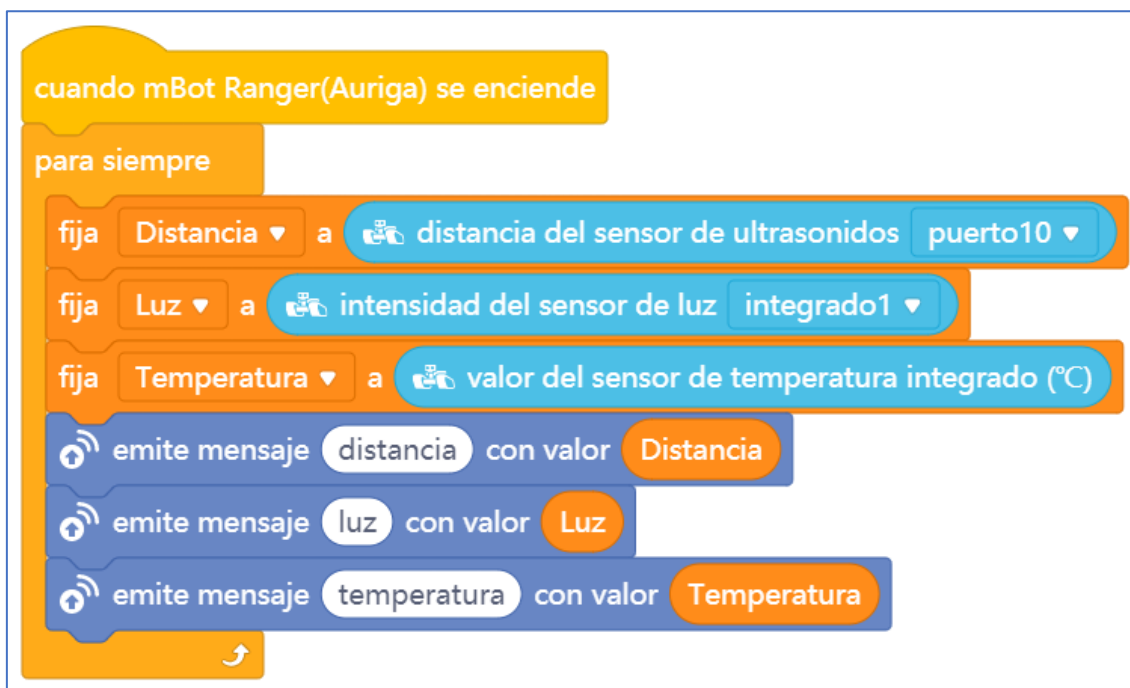


- **Reto 2:** Usaremos como dispositivo el Ranger ya que todos/as lo tenemos en el taller. Vamos a enviar al PC los valores de diferentes sensores (ultrasonidos, sensor de luz y temperatura ambiente).

Dispositivo: Comenzamos creando las variables necesarias para asociarlas con su correspondiente sensor y así enviar mensajes con estos valores. En nuestro ejemplo serán “Distancia”, “Luz” y “Temperatura”. Cargamos la extensión “Mensajes entre dispositivos”.



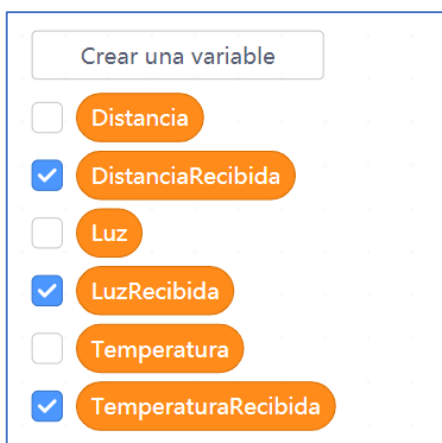
Fijamos los valores de las variables con los sensores correspondientes dentro de un bucle “para siempre”. Finalmente, enviamos los mensajes con estos valores:



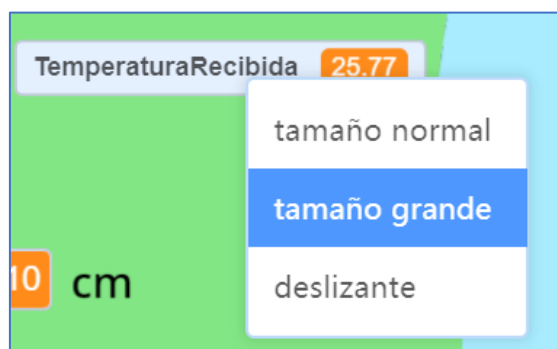
Programa para el dispositivo Ranger

Debemos tener claro que, con el programa anterior el Ranger está continuamente enviando mensajes y no podrá nunca recibir. Si quisiéramos que también lo hiciera, debemos modificar el programa de robot Ranger y, obviamente, el del objeto. Lo veremos en el ejemplo 3.

Objeto: He diseñado un objeto que llamo “Recepción” y en él creamos y situamos 3 nuevas variables para recibir las mediciones y que llamo “DistanciaRecibida”, “LuzRecibida” y “TemperaturaRecibida”. También escondemos las variables que hemos creado para el dispositivo:



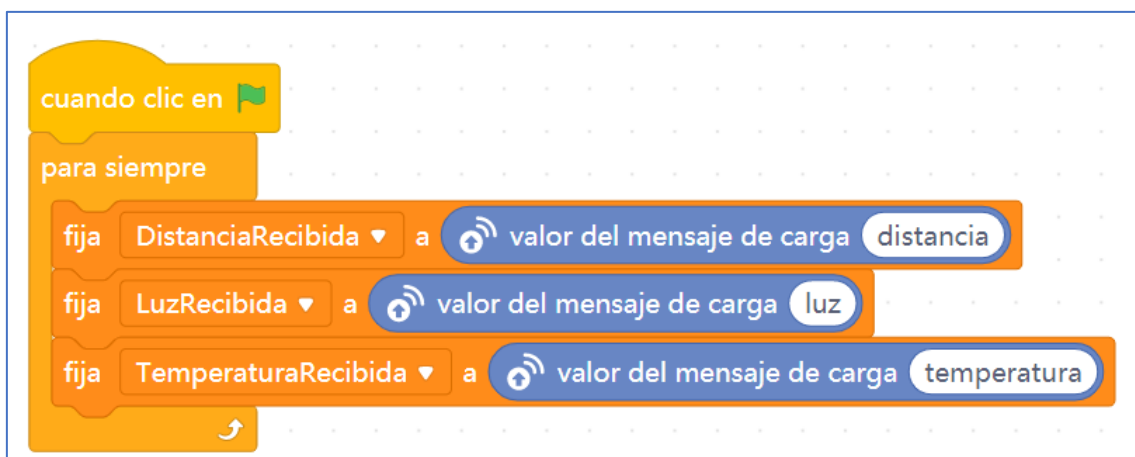
Recordar que al haciendo clic en la variable con el botón derecho del ratón, podemos convertir las variables en otros formatos. Usaremos el tamaño grande para el objeto “Recepción”:



El resultado del montaje en el objeto es el siguiente:

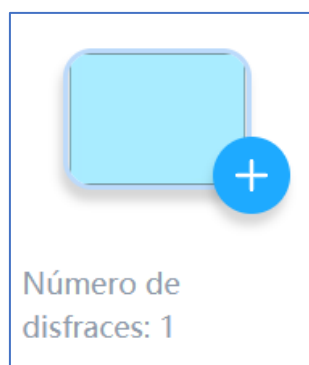


Relativo al programa del objeto sólo debemos dar valor a las variables de recepción. Lo hacemos con el siguiente programa:



Programa para el objeto Recepción

Fondo: Lo cubrimos con un recuadro azul.



Sólo queda probarlo: Conectamos el Ranger y subimos el programa. Tras hacer clic en la “Bandera verde” comenzamos a recibir los datos:



- **Reto 3:** Vamos a aprovechar el programa anterior y modificarlo de modo que el Ranger no sólo envíe datos, sino que también los reciba. La idea es que, además de enviar los datos de los tres sensores del ejemplo 2, reciba la orden de poner los leds de la corona de leds en un color determinado.

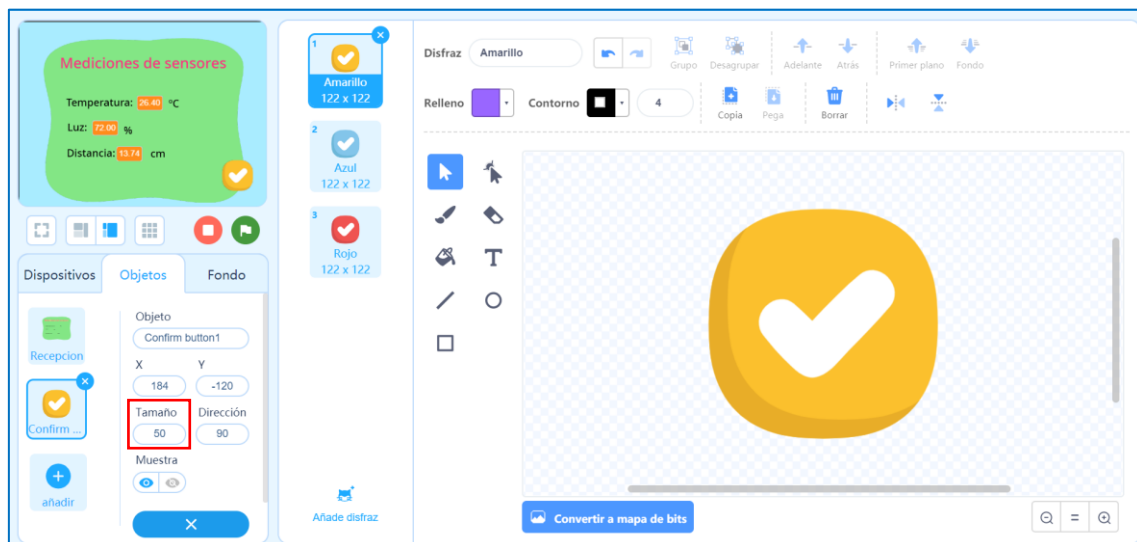
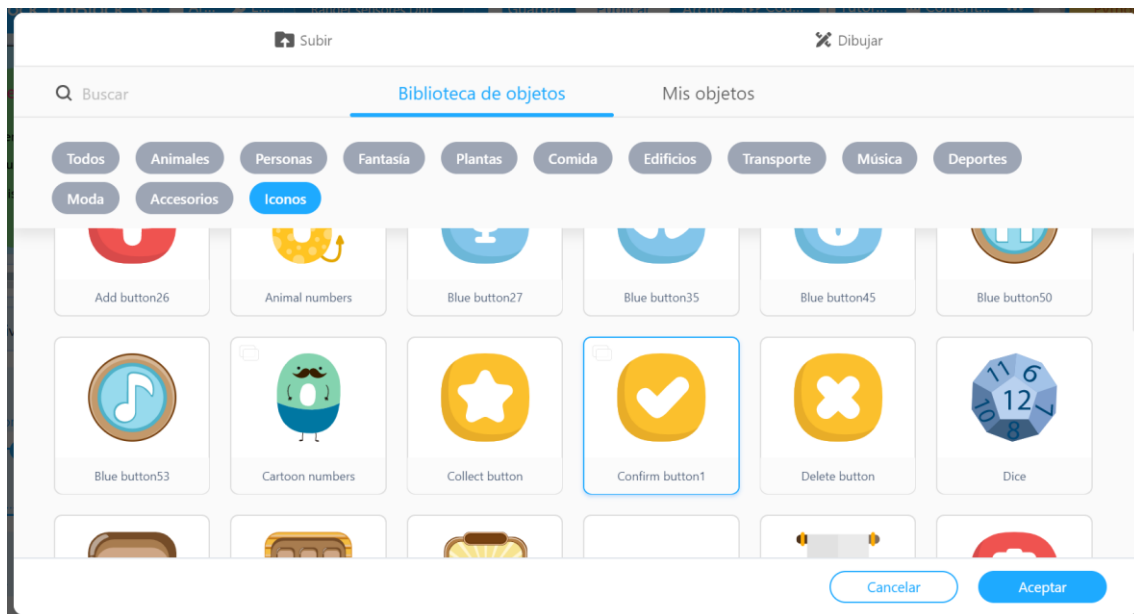
Dispositivo: Modificamos el programa anterior del Ranger para que pueda enviar y recibir. Hemos sacado el comando “Por siempre” en el envío porque si siempre está enviando, no podrá recibir. En su lugar hemos usado el comando “al recibir mensaje sensores” que, obviamente, alguien debe enviarlo y en nuestro caso serán los objetos.



Programa para el dispositivo Ranger

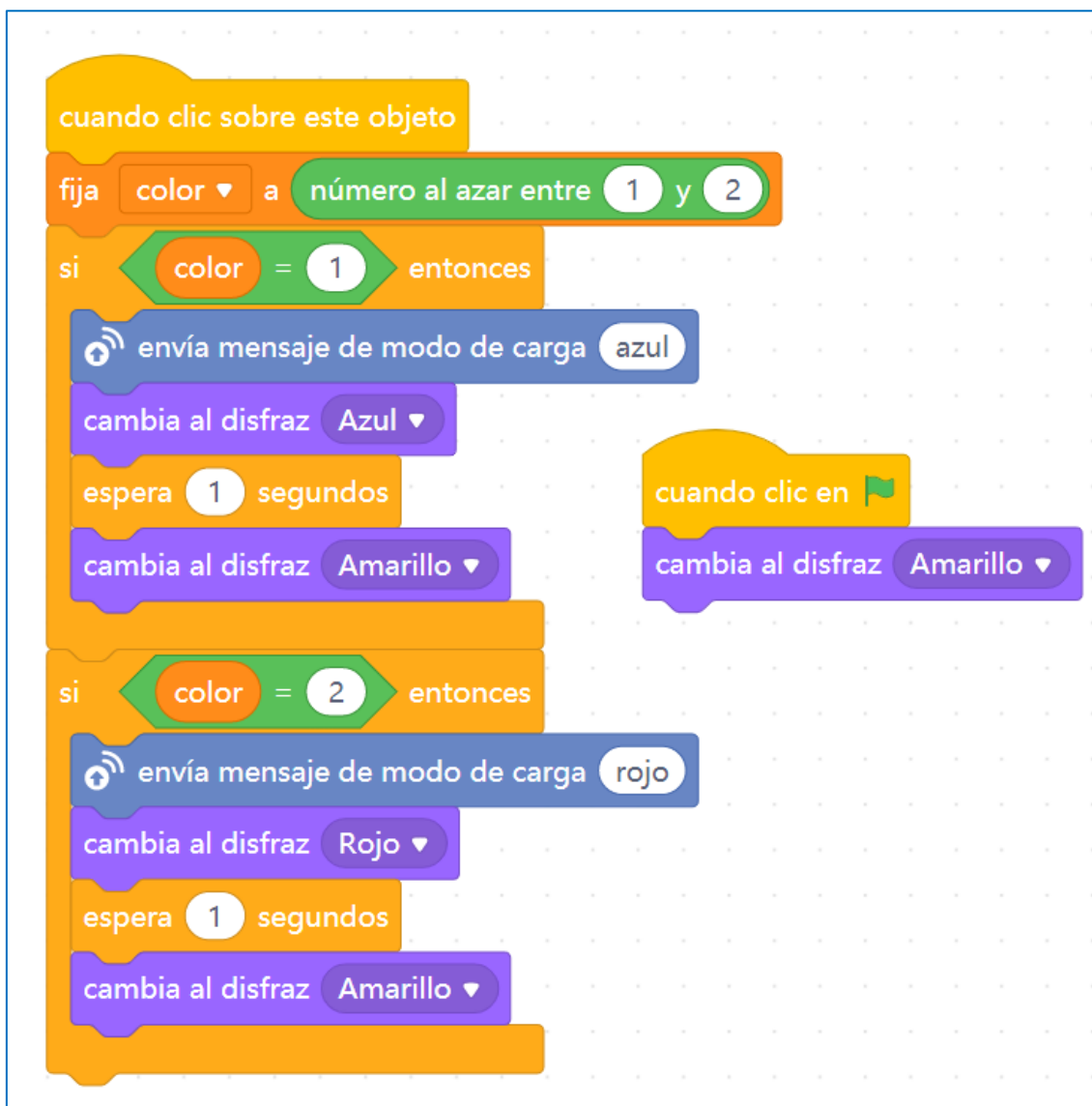
Objetos: Modificamos el programa del objeto “Recepción” e incluimos un nuevo objeto (un simple botón que disponga de varios disfraces) para interactuar con los leds de la corona de leds RGB.

Introducimos el objeto que mBlock llama “Confirm button 1”. Cambiamos su tamaño al 50% y renombramos los disfraces a “Amarillo”, “Rojo” y “Azul”.



Este objeto comenzará con el disfraz amarillo y enviará aleatoriamente los mensajes de color “rojo” y “azul” cuando el usuario haga clic con el ratón en el objeto: (estos mensajes los recibirá el dispositivo que actuará iluminando durante 1 segundo los leds de la corona de LEDs RGB en ese color que se envía):





Programa para el objeto botón

El control de las mediciones lo realiza el objeto “Recepción”. Este objeto se encarga de decirle al Ranger que testee sus sensores cada 0.5 segundos. Por lo tanto, el Ranger cuando no envía los datos de sus sensores, puede recibir órdenes. En nuestro caso, las relativas a los LEDs RGB.

Cuando el Ranger reciba el mensaje “sensores” envía los datos en los mensajes “distancia”, “temperatura” y “luz”. Estos datos los recibe el objeto “Recepción” gracias a la extensión “*Cargar en modo difusión*”:



Programa del objeto Recepción

Anexos

1. [Vídeo de introducción](#)
2. [Extensión Lápiz](#)
3. [Extensión música](#)
4. [Introducción a la AI](#)