

# LA MASCOTA INTELIGENTE



## STEMJAM Teaching Guide

Developing make spaces to promote creativity  
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

[www.stemjam.eu](http://www.stemjam.eu)



Co-funded by the  
Erasmus+ Programme  
of the European Union

# LA MASCOTA INTELIGENTE

## RESUMEN

La idea es crear una "mascota mBot" que proporcionará una respuesta a diferentes reacciones humanas y luego recopilará información sobre ellas.

Los estudiantes codificarán un mBot-Pet para usarlo como un sistema de recolección de datos. Al probar diferentes sensores, estudiarán las reacciones más comunes que un ser humano tiene hacia una mascota (como jugar con ella, acariciarla, hablar bien o gritar). El mBot-Pet tendrá una respuesta específica a cualquiera de estas reacciones humanas.

Cuando el robot tiene un cierto número de reacciones almacenadas, el programa se detendrá y los datos recopilados se mostrarán en la pantalla. Esta información será el punto de partida para crear un estudio estadístico en el que los estudiantes podrán poner en práctica algunos conceptos matemáticos (elaboración de códigos de barras, media, mediana, modo, proporción, proporciones...).

La segunda versión de esta actividad, consiste en interactuar con un mBot disfrazado de bebé.

Esta segunda versión consta principalmente de dos secciones.

- ❖ En la primera parte, se ha desarrollado un código en el que el usuario puede personalizar las emociones del bebé, ya que a través del control remoto o mediante el actuador de 4 botones, al presionar un botón, el bebé mostrará una emoción y al tocar otro botón, el bebé volverá a cambiar la expresión. De esta manera, por ejemplo, se puede hacer que a través de teatros, los más pequeños disfruten de esta actividad.
- ❖ La segunda sección es que el bebé reconocerá 3 estados, cuando alguien o algo se aproxime (sensor de ultrasonidos), si hay demasiada luz (sensor de luz) y demasiado sonido (sensor de sonido). Cuando el bebé reconozca esta acción, emitirá una señal acústica y luminosa.

## OBJETIVOS DIDÁCTICOS

### TECNOLOGÍA

- ❖ Introducción al pensamiento computacional.
- ❖ Estudio y uso de diferentes sensores.
- ❖ Asimilación, creación y programación de algoritmos, para descomponer problemas complejos en secuencias ordenadas de instrucciones simples, que lo resuelven.

### MATEMÁTICAS

- ❖ Resolución de estadísticas y ejercicios de probabilidad.
- ❖ Elaboración de códigos de barras.
- ❖ Media, mediana y modo.
- ❖ Desviación estándar y proporciones.



Materia STEM:            Ciencia             Tecnología             Ingeniería             Matemáticas

Nivel educativo:            12-14 años             14-16 años

### PLANTEAMIENTO DEL PROBLEMA

El objetivo de la actividad es aprender sobre diferentes sensores y recopilar información que se utilizará para resolver ejercicios matemáticos (como la elaboración de códigos de barras y el cálculo de la media, la mediana, el modo, la proporción, las proporciones, etc.) mediante la aplicación de competencias tecnológicas.

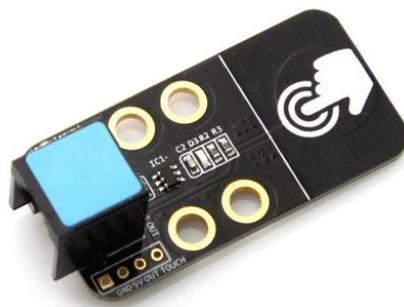
Este objetivo se logrará, ya que los estudiantes deberán diseñar los bloques de programación relacionados con la interacción entre el mBot-Pet y el humano, así como los diferentes efectos de sonido y baile que resultan de las respuestas del robot.

### LISTADO DE MATERIALES

❖ mBot:



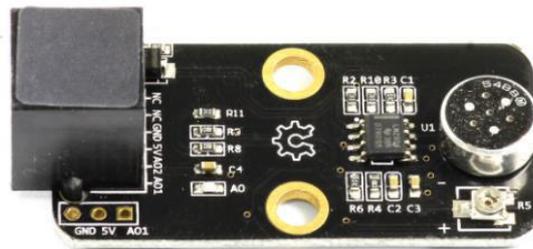
❖ Me Sensor Táctil:



❖ Matriz LED:



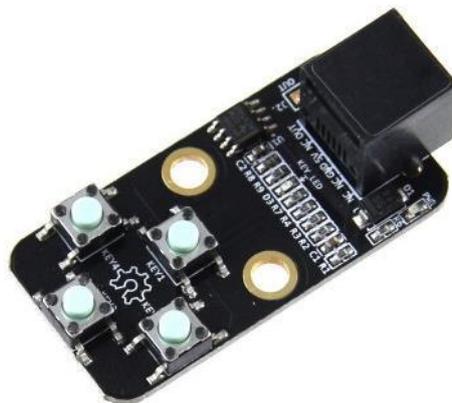
❖ Me Sensor de Sonido:



❖ Me Sensor de Ultrasonido:



❖ Me Actuador 4 Botones:



❖ Disfraz:



❖ Traje de mascota (papel de color y algodón).

❖ Resto de Attrezzo (no esencial).

En la siguiente tabla puede ver a qué puertos hemos conectado cada componente. Los códigos han sido programados siguiendo estas conexiones.

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl	W*	W*
Mbot Robot 2'4G			1																
Motor 1	W*																W*		
Motor 2	W*																	W*	
Ultrasonic sensor	Y	1	1	Y															
Touch sensor	B	1	1			B													
Sound sensor	Bl	1	1							Bl									
Matriz de LEDs	B	1	1										B						
RJ25 cables			4																
Structures																			
	Support P1		3																
	Cut-out beam		4																
	Plate 45°		1																
Laptops			1																
Atrezzo (not essential)			X																

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl	W*	W*
Mbot Robot 2'4G			1																
Motor 1	W*																W*		
Motor 2	W*																W*		
Me RJ 25 adapter	Y																		
	B																		
	Bl																		
Mini Pan-Tilt kit																			
It has 2 servos.																			
We have to connect the servo to a RJ25 adapter																			
Mini Gripper																			
We have to connect the servo to a RJ25 adapter																			
Me 7-Segment serial display	B																		
Me Led Matrix 8x16	B	(1)	1							B									
Me Ultrasonic sensor	Y	(1)	1			Y													
Me Temperature Sensor - Waterproof	Y																		
Me Line Follower	B																		
Me Flame sensor	Bl																		
Me PIR Motion sensor	B																		
Me Sound sensor	Bl	(1)	1												Bl				
Me Touch sensor	B																		
Mini Fan Pack	B																		
Me Temperature and Humidity sensor	Y																		
Me 130 Motor Fan Pack	B																		
RJ25 cables			3																
Structures and beams																			
Laptops			1																
Attrezzo (not essential)																			

## DESCRIPCIÓN DE LA ACTIVIDAD

### Primera versión

El *mBot-Pet* se queda quieto y espera a que aparezca un humano. Una vez que detecta al humano, va corriendo hacia él para interactuar con esta persona. A partir de este momento, el programa depende del comportamiento humano.:

- ❖ Si el humano toca el *mBot-Pet*, muestra su amor con corazones, los sonidos felices y las luces de color rosa intermitentes y la variable "Caress" agrega un punto.
- ❖ Si el humano juega con el *mBot-Pet*, reproduce y la variable "Playing " agrega un punto.
- ❖ Si el humano habla con el *mBot-Pet* normalmente, se comunica al mostrar un corazón y la variable "Nice words" agrega un punto.
- ❖ Si el humano habla con el *mBot-Pet* en voz alta, se comunica con un mensaje inesperado y sorprendente y la variable "Bad words" agrega un punto.

El robot interactuará con el humano hasta que tenga un número específico de respuestas y luego el programa finalizará mostrando los datos recopilados. Se han resuelto un número de 10 respuestas, pero este valor puede modificarse.

Después de descargar el software mBlock, se emparejará con el mBot mediante el uso del puerto serie inalámbrico de 2.4G. Una vez que se realiza el emparejamiento, el siguiente paso será comenzar con las tareas de PROGRAMACIÓN.

### 1. DEFINICIÓN DE SALUDO

El mBot permanece quieto hasta que detecta una presencia a una distancia de 100 cm. Luego, llega a la presencia detectada y se detiene 20 cm antes de encontrarse, mientras muestra el mensaje "¡Hola!" Y enciende las luces rosas a bordo.

Estas dos distancias pueden ser editables, dependiendo de las especificaciones necesarias.



```

define Greeting performance
  show drawing Port4 x: 0 y: 0 draw: ...
  set Caress to 0
  set Playing to 0
  set Nice words to 0
  set Bad words to 0
  set Noise to 0
  repeat until ultrasonic sensor Port1 distance < 20
    if ultrasonic sensor Port1 distance < 100 then
      show drawing Port4 x: 0 y: 0 draw: Hi!!
      set led on board all red 60 green 0 blue 60
      set motor M1 speed 255
      set motor M2 speed 255
    else
      show drawing Port4 x: 0 y: 0 draw: ...
  set motor M1 speed 0
  set motor M2 speed 0
  wait 2 secs
  show drawing Port4 x: 0 y: 0 draw: 
  set led on board all red 0 green 0 blue 0

```

## 2. DEFINICIÓN DE LA CARICIA:

El desempeño de la caricia consiste en agregar un punto a la variable "Caricia" y luego repetir tres veces un mensaje intermitente combinado sobre corazones y luces de color rosa a bordo.

```

define Caress performance
  change Caress by 1
  repeat 3
    show drawing Port4 x: 0 y: 0 draw: 
    set led on board all red 150 green 0 blue 25
    wait 0.5 secs
    show drawing Port4 x: 0 y: 0 draw: 
    set led on board all red 0 green 0 blue 0
    wait 0.5 secs

```

### 3. DEFINICIÓN DE JUEGO:

La interpretación en ejecución consiste en agregar un punto a la variable "Reproducción" y luego mostrar los siguientes efectos:

- Mensaje "si!!"
- Melodía corta.
- Leds y series de movimiento repetidas dos veces.  
Borrar todo lo anterior.



```
define Playing performance
  change Playing by 1
  show drawing Port4 x: 0 y: 0 draw: !si!
  play tone on note G5 beat Eighth
  play tone on note G6 beat Quarter
  play tone on note G6 beat Eighth
  play tone on note G7 beat Quarter
  repeat 2
    set led on board all red 0 green 20 blue 60
    set motor M1 speed 100
    set motor M2 speed -100
    wait 0.5 secs
    set led on board all red 0 green 60 blue 20
    set motor M1 speed -100
    set motor M2 speed 100
    wait 0.5 secs
  set motor M1 speed 0
  set motor M2 speed 0
  set led on board all red 0 green 0 blue 0
  show drawing Port4 x: 0 y: 0 draw: 
```

#### 4. DEFINICIÓN DE COMUNICACIÓN:

El rendimiento de la comunicación se ha diseñado teniendo en cuenta la modulación de voz de un ser humano. De acuerdo con esto, cuando un ser humano está enojado, generalmente habla con un tono de voz más alto y comunica mensajes cortos y directos.

Para elaborar este código, se necesitan dos fases:

- ❖ **Fase 1:** "Se borra la lista de "modulación de voz". El sensor de sonido recopila tres valores de sonido diferentes en tres segundos, y se almacenan en la lista de "Modulación de voz". La variable "Max. Noise" se establece en un valor que se considera más alto de lo normal (300 en nuestro caso, pero este valor puede cambiar según las condiciones físicas del entorno).
- ❖ **Fase 2:** Si uno de los tres valores recopilados es mayor que el valor establecido como "Max. Noise", el mBot percibe gritos y reacciona mostrando un mensaje confuso en la pantalla Led, luces rojas a bordo y tonos bajos de música. Entonces, el valor de "Bad words" agrega un punto.

Por el contrario, si todos los valores recopilados están por debajo del valor establecido como "Max. Noise", el mBot percibe la comunicación con el ser humano y reacciona mostrando un mensaje amoroso de corazones y luces de color rosa a bordo. Entonces, el valor de "Nice words" agrega un punto.

```
define Communication performance
  delete all of Voice modulation
  repeat 3
    show drawing Port3 x: 0 y: 0 draw: [ ]
    set Max. noise to 300
    set Noise to Voice modulation
    add sound sensor Port3 to Voice modulation
    wait 0.5 secs
    show drawing Port4 x: 0 y: 0 draw: [ ]
    wait 0.5 secs
  if item 1 of Voice modulation > Max. noise or item 2 of Voice modulation > Max. noise or item 3 of Voice modulation > Max. noise then
    show drawing Port4 x: 0 y: 0 draw: [Bad words]
    set led on board all red 255 green 0 blue 0
    play tone on note C4 beat Half
    play tone on note C4 beat Half
    change Bad words by 1
    wait 1 secs
    show drawing Port4 x: 0 y: 0 draw: [ ]
    set led on board all red 0 green 0 blue 0
  else
    show drawing Port4 x: 0 y: 0 draw: [Nice words]
    set led on board all red 255 green 0 blue 60
    play tone on note G5 beat Eighth
    play tone on note G7 beat Quarter
    change Nice words by 1
    wait 1 secs
    show drawing Port4 x: 0 y: 0 draw: [ ]
    set led on board all red 0 green 0 blue 0
  set led on board all red 0 green 0 blue 0
  set motor M1 speed 0
  set motor M2 speed 0
```

## 5. DEFINICIÓN DE DESPEDIRSE

La función de despedirse consiste en agregar un punto a la variable "Caress" y luego repetir dos veces un mensaje intermitente combinado sobre la iluminación completa de la pantalla LED y mostrar luces en el tablero mientras se reproduce un tono.

Luego, mBot comunica que los resultados obtenidos de la actividad están disponibles en la pantalla de la computadora portátil, en el programa mBlock.

Finalmente, dice un mensaje de "¡Adiós!" Mientras muestra las luces en el tablero y reproduce un tono, y todo se borra.



```
define Goodbye performance
repeat 2
  show drawing Port4 x: 0 y: 0 draw: [Blue]
  set led on board all red 255 green 0 blue 255
  play tone on note C5 beat Quarter
  show drawing Port4 x: 0 y: 0 draw: [White]
  set led on board all red 0 green 0 blue 0
  wait 0.5 secs
set text to 20
repeat until text = -150
  show face Port4 x: text y: 0 characters: Results are on the screen
  change text by -1
wait 1 secs
show drawing Port4 x: 0 y: 0 draw: [Bye!]
set led on board all red 0 green 255 blue 60
play tone on note G5 beat Half
wait 2 secs
show drawing Port4 x: 0 y: 0 draw: [White]
set led on board all red 0 green 0 blue 0
```

La interfaz de la pantalla mBlock mostrará cómo aumentan las variables en función de la interacción con el humano, al igual que la imagen de abajo.



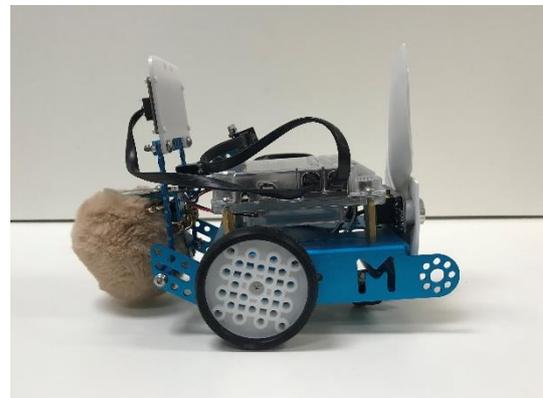
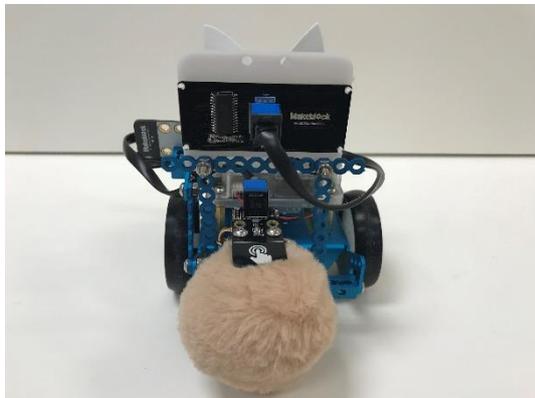
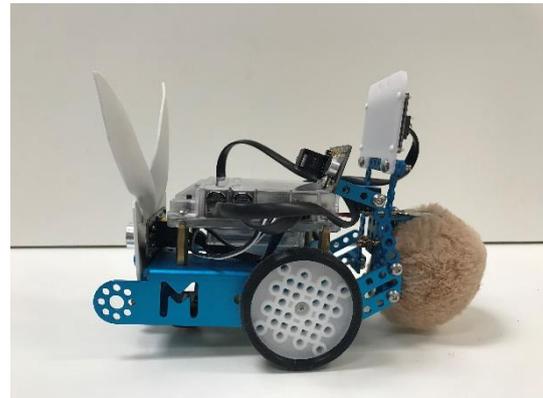
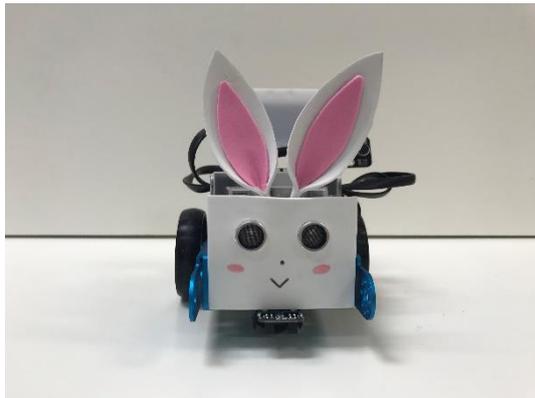
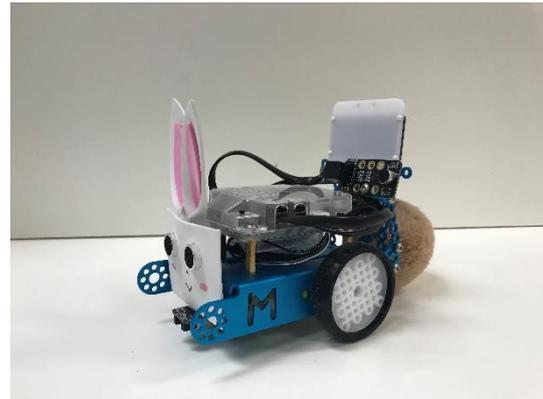
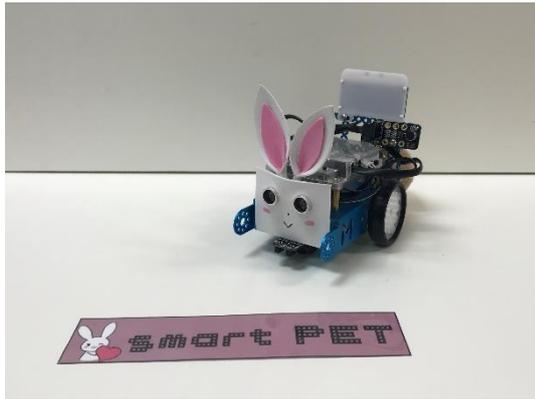
#### 6. PROGRAMA PRINCIPAL: la secuencia de la actividad.

El programa principal de la actividad se muestra en la siguiente imagen:

- ❖ En primer lugar, se llama a la función "Greeting performance".
- ❖ Entonces, el mBot puede seguir tres caminos diferentes dependiendo de la interacción con el humano:
  1. Si la persona lo toca, se completará la función "Caress performance".
  2. Si la persona detecta a alguien a una distancia cercana (10 cm), se completará "Playing performance".
  3. Si la persona comienza a hablar con un tono de voz normal (detectado como el valor "200"), completará el "Communication performance".
- ❖ Cuando haya recopilado un número específico de respuestas (en nuestro caso, 10 respuestas, pero esta cantidad es editable), se realizará el "Goodbye performance" y el programa finalizará. Los resultados se mostrarán en la pantalla..

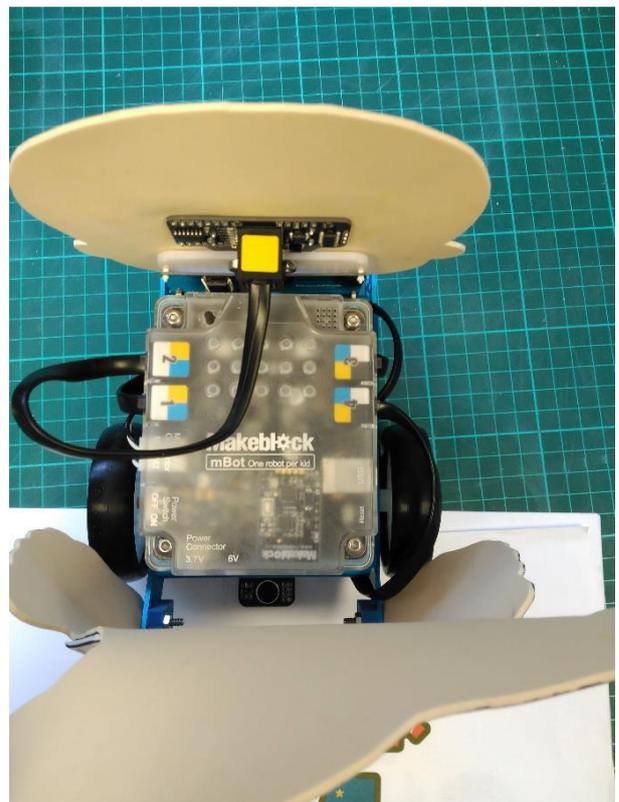
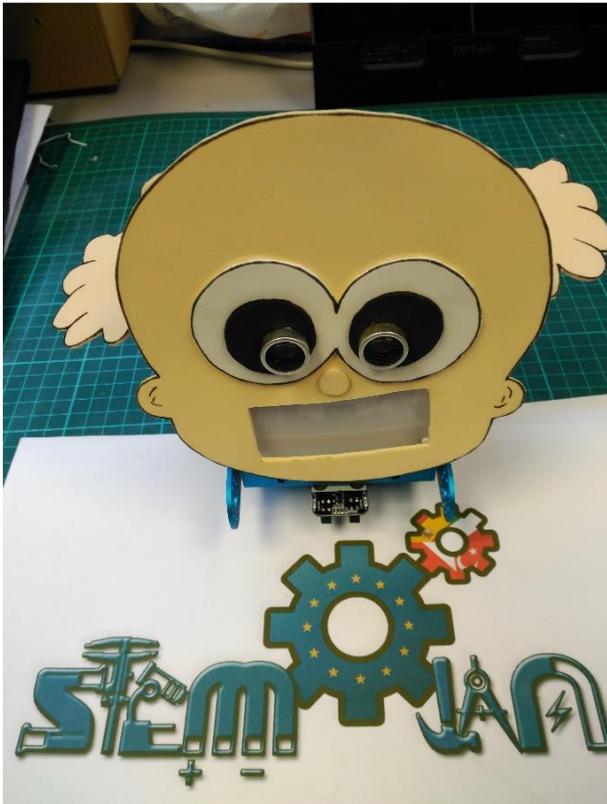
```
when flag clicked
  Greeting performance
  wait 2 secs
  repeat until Caress + Playing + Nice words + Bad words = 10
    if touch sensor Port2 = 1 then
      Caress performance
      wait 2 secs
    if ultrasonic sensor Port1 distance < 10 then
      Playing performance
      wait 2 secs
    if sound sensor Port3 > 200 then
      Communication performance
      wait 2 secs
  Goodbye performance
```

Una vez que se termina la programación, comenzamos con LA CONSTRUCCIÓN DE LA ESTRUCTURA donde se establecerán todos los elementos mecánicos, al igual que los elementos electrónicos.



## Segunda versión

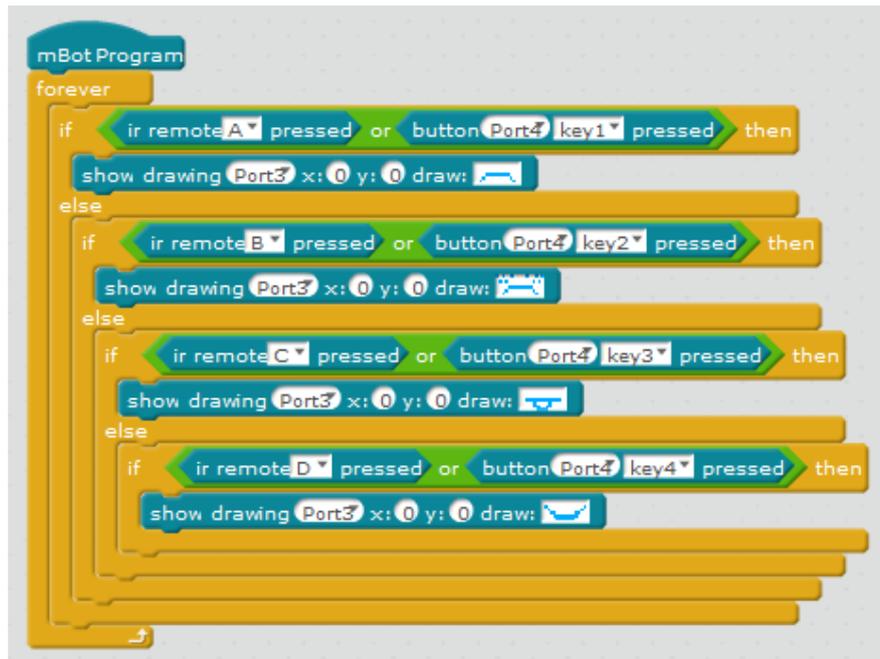
Para el desarrollo de la actividad, tendremos que adaptar un disfraz a nuestro mBot, que representará el animal o ser con el que queremos jugar.



## 1. Personaliza las emociones del bebé:

En la primera parte, se ha desarrollado un código en el que el usuario puede personalizar las emociones del bebé, ya que a través del control remoto o mediante el actuador de 4 botones, al presionar un botón, el bebé mostrará una emoción y al tocar otro botón, el bebé volverá a cambiar la expresión. De esta manera, por ejemplo, se puede hacer que a través de teatros, los más pequeños disfruten de esta actividad.

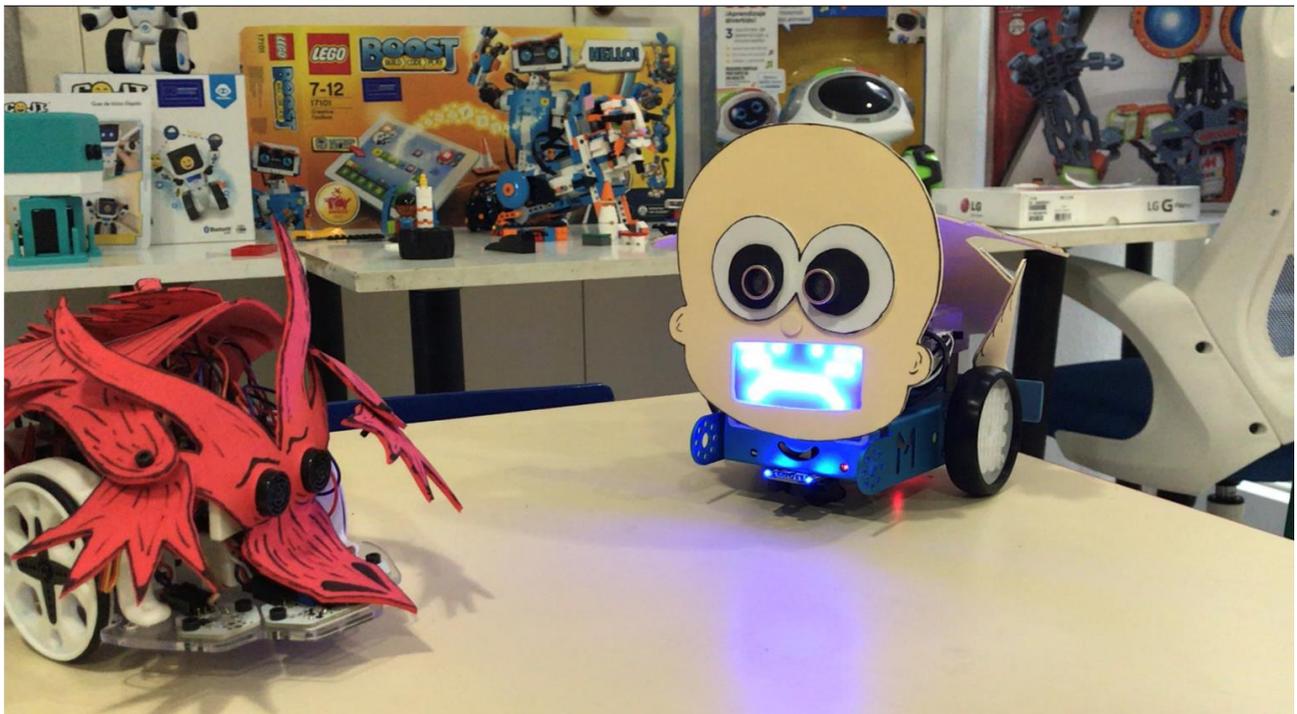
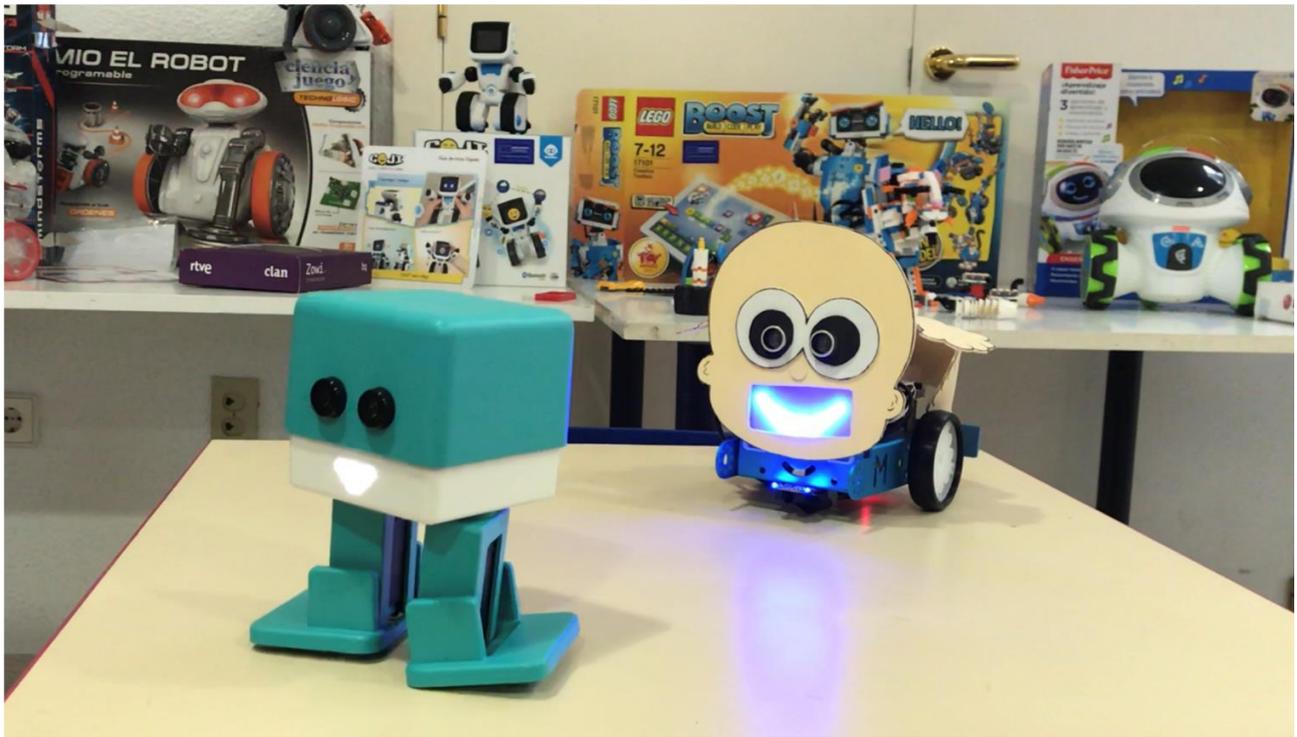
Una vez que tenemos nuestro atrezzo, comenzamos la programación para el mBot:



```
mBot Program
forever
  if ir remote A pressed or button Port4 key1 pressed then
    show drawing Port3 x: 0 y: 0 draw: [Smiling Face]
  else
    if ir remote B pressed or button Port4 key2 pressed then
      show drawing Port3 x: 0 y: 0 draw: [Neutral Face]
    else
      if ir remote C pressed or button Port4 key3 pressed then
        show drawing Port3 x: 0 y: 0 draw: [Sad Face]
      else
        if ir remote D pressed or button Port4 key4 pressed then
          show drawing Port3 x: 0 y: 0 draw: [Angry Face]
```

El código es muy simple, donde siempre se repite el bucle, y cuando el usuario presiona el botón del control remoto o uno de los botones del "Actuador de 4 botones", la Matriz LED mostrará la expresión que se ha determinado.

Para explicar mejor esta primera sección, hemos utilizado diferentes robots, que aparecerán frente al Baby STEMJAM y esto mostrará sus sentimientos, que serán seleccionados por el usuario a través del control remoto.





## 2. No despiertes al bebé:

La segunda sección es que el bebé reconocerá 3 estados, cuando alguien o algo se acerca (sensor ultrasónico), si hay demasiada luz (sensor de luz) y demasiado sonido (sensor de sonido). Cuando el bebé reconozca esta acción, emitirá una señal acústica y luminosa.

El objetivo es convertir mBot en un bebé real, que se adaptará a cualquier circunstancia, como veremos en el código del programa.

En caso de cualquier anomalía, el mBot emitirá una alarma y mostrará en la Matriz de LED la causa por la cual se emitió. Una vez activada, la alarma se puede desactivar presionando el botón que está integrado en la placa del mBot.

Para lograr el objetivo, utilizaremos un sensor de ultrasonido, el sensor de luz y el sensor de sonido.

Ahora, detallaremos el código:

El código ha sido desarrollado en Bitbloq (<http://bitbloq.bq.com>)

## 1. Declaración de variables:

### – Global variables, functions and classes

The image shows a Scratch-style variable declaration block for a project named "STEMJAM BABY". The block contains the following declarations:

- Comment // STEMJAM BABY
- Declare variable DistanceIni = 0
- Declare variable DistanceAct = 0
- Declare variable DistanceVar = 0
- Declare variable LightIni = 0
- Declare variable LightAct = 0
- Declare variable LightVar = 0
- Declare variable SoundIni = 0
- Declare variable SoundAct = 0
- Declare variable SoundVar = 0
- Declare variable ReasonAlarm with type text =
- Declare variable Alarm = 0
- Declare variable I = 0

Al iniciar el programa, el mBot tomará como referencia los valores actuales de distancia, luz y sonido, por lo que cualquier variación que exceda un margen determinado por el usuario en el programa provocará que la alarma se active, de modo que se emita una alarma. Ese efecto produce un mayor realismo a la actividad, ya que nuestro mBot podrá adaptarse a cualquier circunstancia.

Las variables que vemos en la imagen son el valor inicial de la distancia, el valor actual de la distancia y su variación, lo que determinará si el margen se ha excedido o no. Lo mismo para sonido y luz.

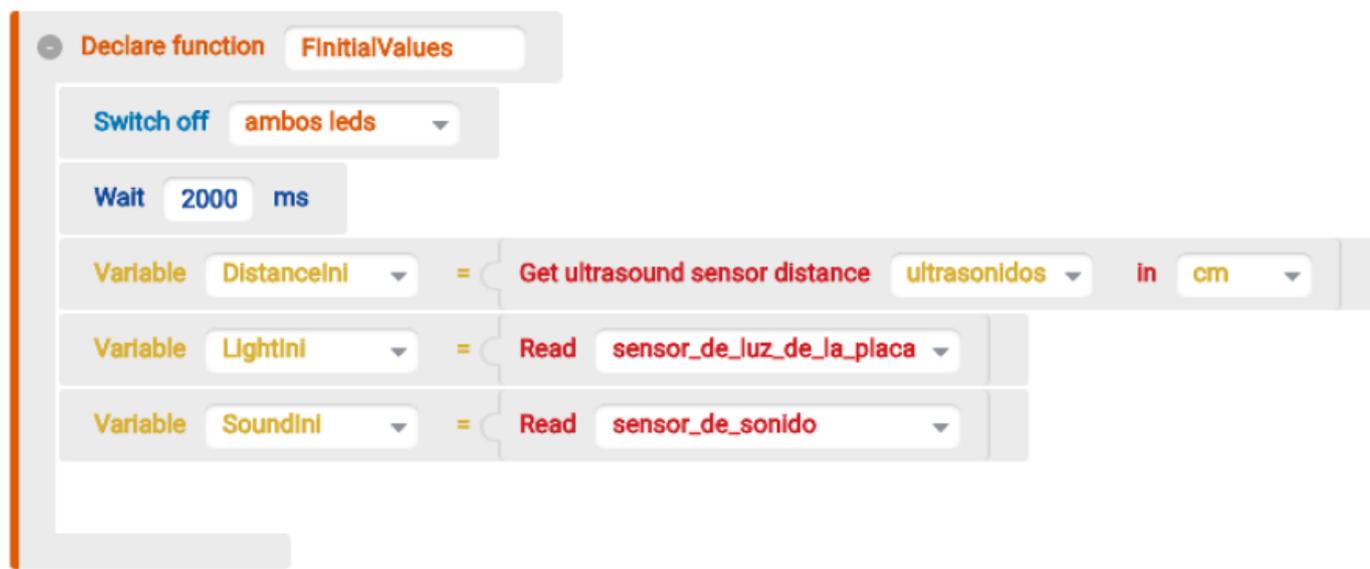
También tenemos una variable que mostrará la razón por la que la alarma ha sonado y si la alarma está activa o no.

## 2. Función de “espera”:

```
Declare function FWait
  Variable I = 3
  While Variable I >= 0 do:
    Display the number Variable I on the matriz_de_leds
    Sound the buzzer with the note Si for 50 ms
    Switch on ambos leds in green
    Wait 450 ms
    Switch off ambos leds
    Wait 500 ms
    Variable I = Variable I - 1
  End While
  Sound the buzzer with the note Si for 500 ms
```

La función de espera cuenta hacia atrás. En el momento en que se presiona el botón de la placa, empieza una cuenta regresiva de 3 segundos, y cuando llega a 0, el bebé ya estará expuesto a cualquier variación del entorno.

### 3. Función de valores iniciales:

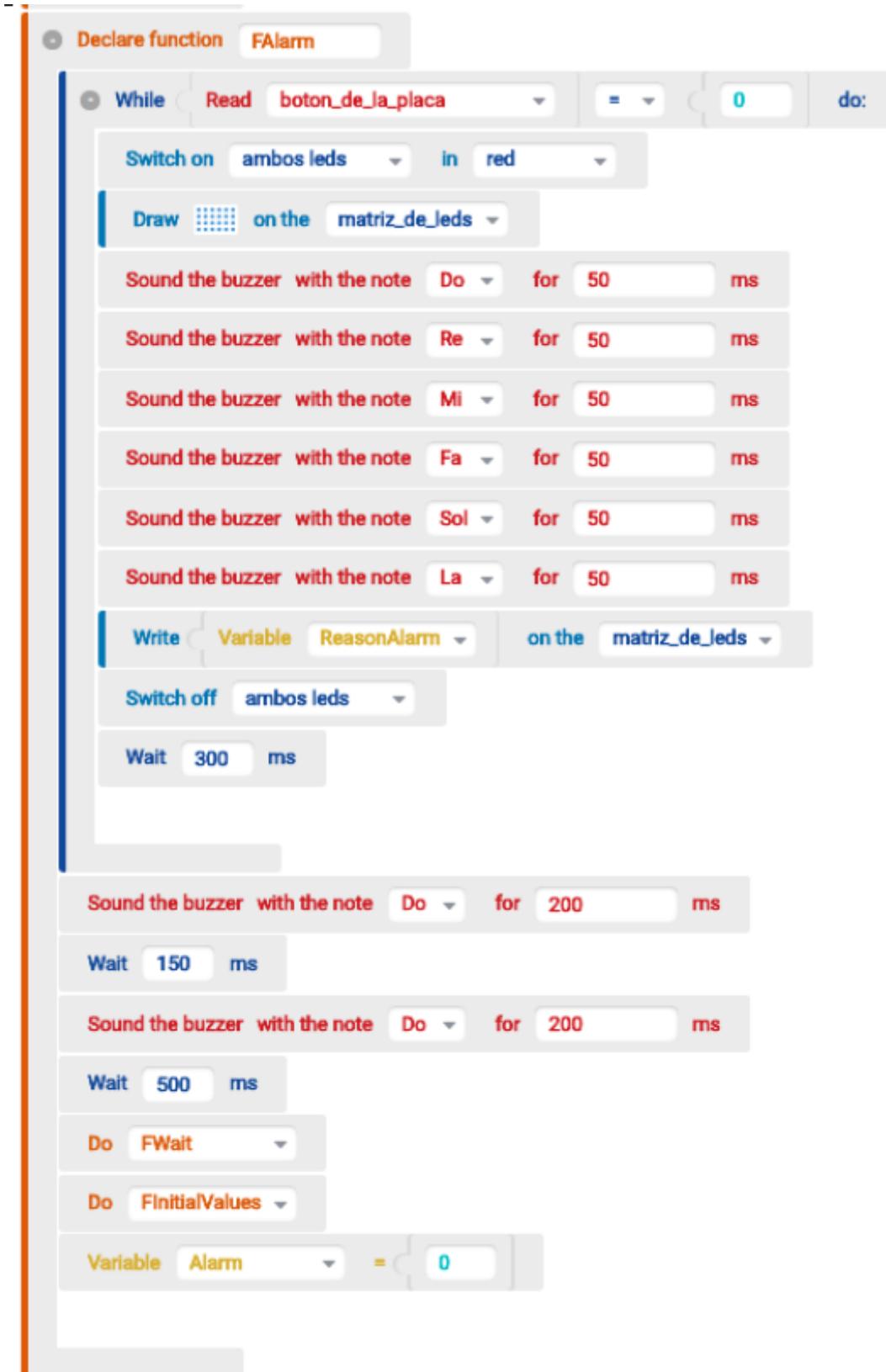


```
Declare function FinitialValues
  Switch off ambos leds
  Wait 2000 ms
  Variable DistanceIni = Get ultrasound sensor distance ultrasonidos in cm
  Variable LightIni = Read sensor_de_luz_de_la_placa
  Variable SoundIni = Read sensor_de_sonido
```

Esta función establece los valores iniciales de cada sensor.

#### 4. Función "Alarma":

Es la función que emitirá el sonido de la alarma y que se detendrá cuando se presione el botón de la placa.



```
Declare function FAlarm
  While Read boton_de_la_placa = 0 do:
    Switch on ambos leds in red
      Draw [LED] on the matriz_de_leds
      Sound the buzzer with the note Do for 50 ms
      Sound the buzzer with the note Re for 50 ms
      Sound the buzzer with the note Mi for 50 ms
      Sound the buzzer with the note Fa for 50 ms
      Sound the buzzer with the note Sol for 50 ms
      Sound the buzzer with the note La for 50 ms
      Write Variable ReasonAlarm on the matriz_de_leds
      Switch off ambos leds
      Wait 300 ms
    Sound the buzzer with the note Do for 200 ms
    Wait 150 ms
    Sound the buzzer with the note Do for 200 ms
    Wait 500 ms
    Do FWait
    Do FIntialValues
  Variable Alarm = 0
```

The image shows a Scratch script for a function named 'FAlarm'. The script is enclosed in a 'Declare function' block. Inside, there is a 'While' loop that runs as long as the variable 'boton\_de\_la\_placa' is equal to 0. The loop contains several blocks: a 'Switch on' block for 'ambos leds' in 'red' mode, which triggers a 'Draw' block on the 'matriz\_de\_leds' variable. This is followed by seven 'Sound the buzzer' blocks, each with a different note (Do, Re, Mi, Fa, Sol, La) and a duration of 50 ms. After the sounds, there is a 'Write' block for the variable 'ReasonAlarm' on the 'matriz\_de\_leds' variable, followed by a 'Switch off' block for 'ambos leds' and a 'Wait' block for 300 ms. Below the loop, there are three more 'Sound the buzzer' blocks (all with note 'Do' and 200 ms duration), a 'Wait' block for 150 ms, another 'Sound the buzzer' block (note 'Do', 200 ms), a 'Wait' block for 500 ms, and two 'Do' blocks for 'FWait' and 'FIntialValues'. Finally, there is a 'Variable' block that sets 'Alarm' to 0.

## 5. Bucle:

```
Loop
  Variable DistanceAct = Get ultrasound sensor distance ultrasonidos in cm
  Variable DistanceVar = Variable DistanceIni - Variable DistanceAct
  Variable DistanceVar = Absolute value Variable DistanceVar
  Variable LightAct = Read sensor_de_luz_de_la_placa
  Variable LightVar = Variable LightIni - Variable LightAct
  Variable LightVar = Absolute value Variable LightVar
  Variable SoundAct = Read sensor_de_sonido
  Variable SoundVar = Variable SoundIni - Variable SoundAct
  Variable SoundVar = Absolute value Variable SoundVar
  If Variable Alarm != 1 do:
    Write ZZZ on the matriz_de_leds
    Wait 2000 ms
    Write on the matriz_de_leds
    Wait 2000 ms
```

En esta parte del código, se consulta la variable "DistanceAct", si la distancia actual ha cambiado mucho con respecto a la "DistanceIni", la alarma se activará debido a que hay demasiada variación. (Para el sonido y la luz es el mismo proceso).

```
if (Variable DistanceVar > 10) do:  
  Variable Alarm = 1  
  Variable ReasonAlarm = 'Near'  
  
if (Variable LightVar > 80) do:  
  Variable Alarm = 1  
  Variable ReasonAlarm = 'Light'  
  
if (Variable SoundVar > 150) do:  
  Variable Alarm = 1  
  Variable ReasonAlarm = 'Sound'  
  
if (Variable Alarm = 1) do:  
  Do FAlarm
```

Cuando las variables "xxxVar" son mayores que el valor predeterminado por el programador, la alarma se activará y la variable "ReasonAlarm" mostrará porque la alarma se ha ejecutado.

A continuación, se muestran algunas imágenes de la actividad:

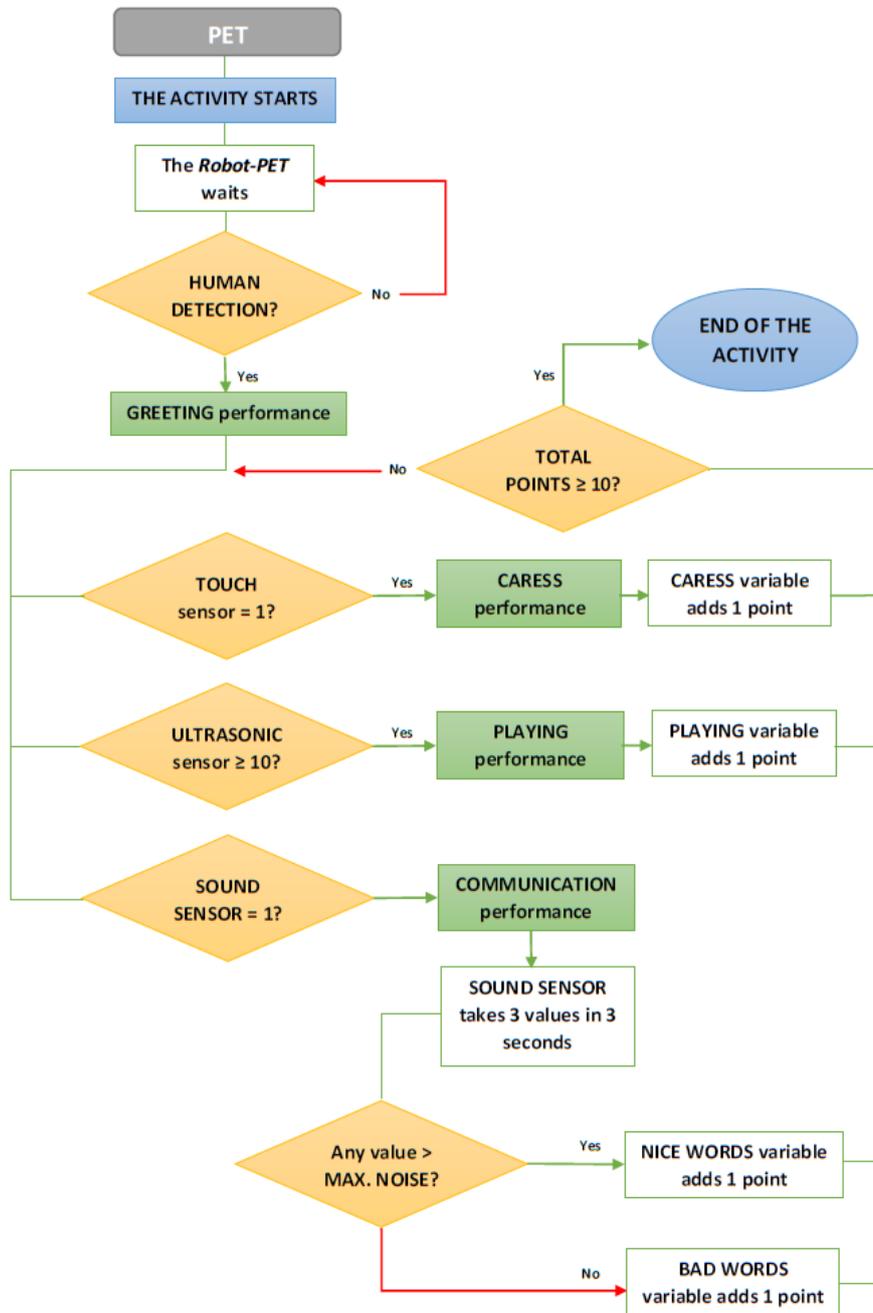




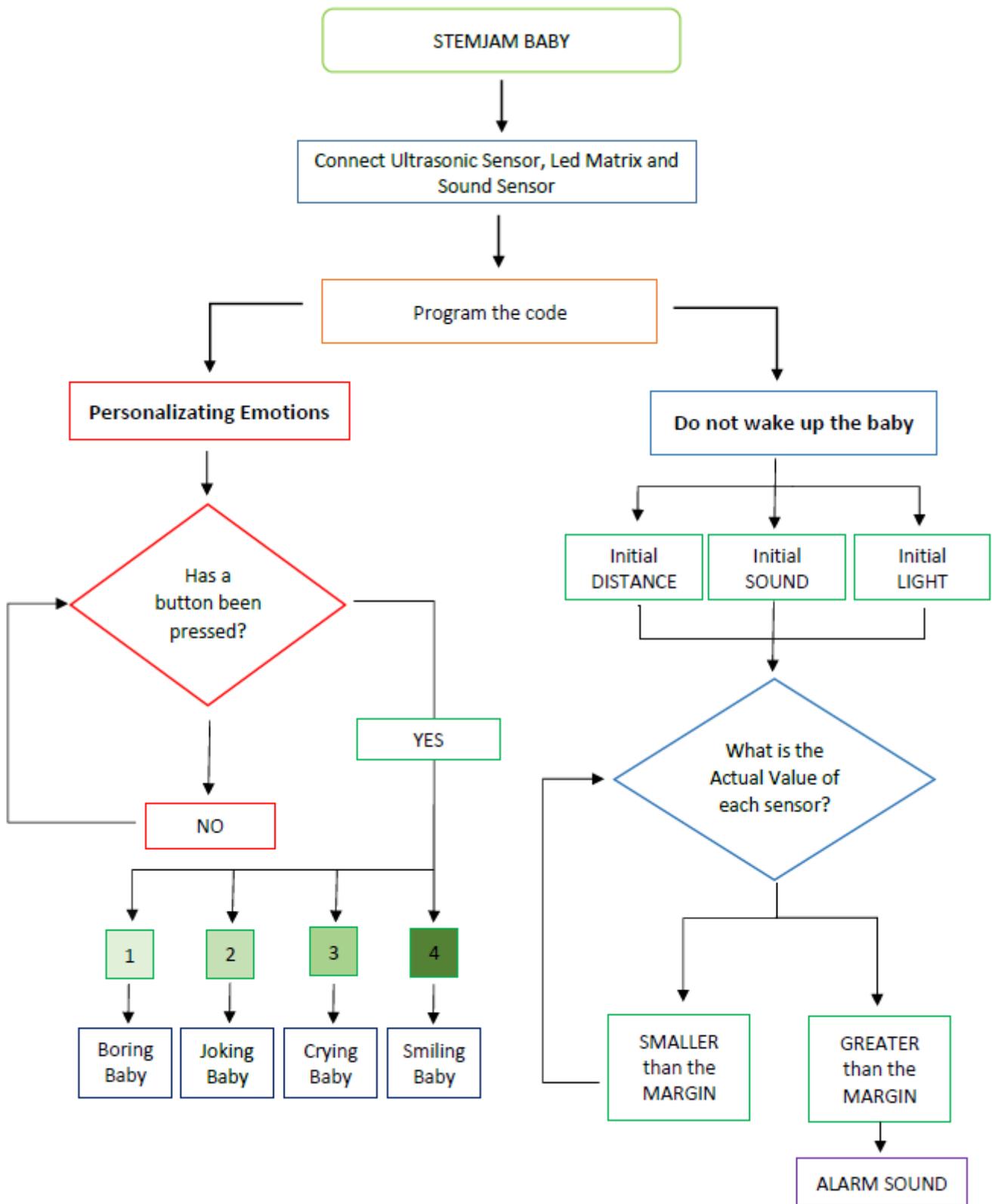


# DIAGRAMA DE FLUJO

## Primera versión



Segunda versión



## EVALUACIÓN DE LOS ESTUDIANTES

Para la evaluación de los estudiantes en esta actividad, use la rúbrica de evaluación diseñada para este proyecto.

### BIBLIOGRAFÍA

“Jugando con MBlock”. Makeblock España

“Divirtiéndome con MBot”. Susana Oubiña

“Prácticas mBot”. Javier Fernández Panadero

Comunidad de Makeblock en español. (<http://www.makeblock.es/foro/>)

### ESCALABILIDAD

La escalabilidad de esta actividad puede ser tan difícil como quieras, siempre basada en hacer que el mBot sea lo más real posible para el ser que imitamos.

### MÁS INFORMACIÓN

#### DIFICULTADES:

- ❖ **DETERMINACIÓN DEL TONO DE VOZ HUMANA:** fue muy difícil establecer un patrón para diferenciar entre hablar agradable y desagradable. Por eso, después de leer artículos médicos sobre modulación de voz y artículos musicales sobre tonos, se decidió que había dos variables involucradas (tiempo y nivel de sonido), y combinándolos fue posible aproximar esta colección de valores.
- ❖ **PROGRAMACIÓN DEL SENSOR TÁCTIL:** debido a la falta de conocimiento sobre el funcionamiento del sensor táctil, fue difícil hacer que el sensor funcionara. El led que incorpora da signos de actividad, pero el sensor simplemente no puede funcionar si no se ha igualado a un valor binario ("1").

