Reto zoológico en micro:bit

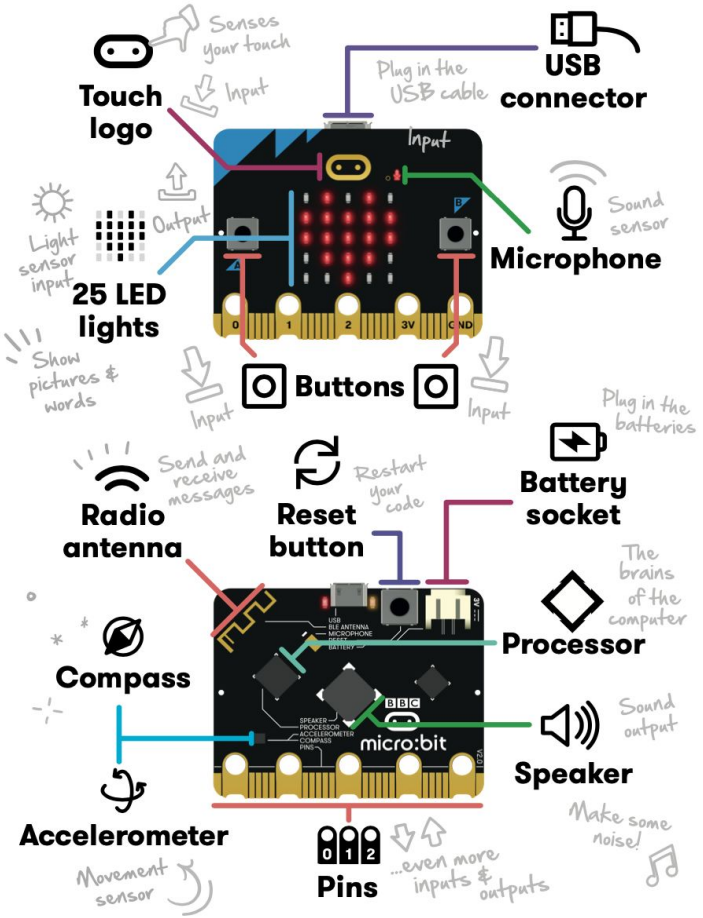
ZER Moianès Llevant 2023

PRESENTACIÓN

La placa **micro:bit** permite crear letras, números y dibujos de una forma sencilla.

En este **RETO** se trata de conseguir crear un **zoológico** con varias tarjetas **micro:bit**

Veamos cómo hacerlo en el simulador de Makecode y cómo subirlo al **micro:bit**



EL RETO

Queremos crear un **zoológico** con **diversos animales** utilizando varias tarjetas micro:bit. Cada persona participante debe elegir en su tarjeta un animal y **compartirlo** con las demás. El objetivo es conseguir que el animal elegido por cada participante sea diferente de los demás.

El programa será idéntico en todas las tarjetas. Se dibujarán nueve animales diferentes, creados por los participantes, que se podrán seleccionar pulsando un botón. Con otro botón se enviará el animal elegido al resto de tarjetas. Si no hay ningún animal repetido se considerará conseguido el **RETO de crear un zoológico diverso**. En caso contrario se deberá seguir probando con otros animales.

AYUDA

Para facilitar la resolución del RETO se pueden establecer algunas ayudas.

Se puede crear un juego de pistas que ayude a solucionar el reto de una forma más fácil.

Quizás se puede dar pistas sobre el tamaño de los animales, alguna de sus características, el país de origen, u otro tipo de pistas.

Quizás con un botón se pueda obtener una pista en la propia **micro:bit**

GRADO DE DIFICULTAD

La resolución de este reto es abordable en primaria si se dispone de bastante tiempo, debido a la necesidad de explicar con detalle cada fase del mismo.

Hemos dividido las explicaciones en dos partes:

- hasta la página 39 se trabajan todos los conceptos de creación y manipulación de imágenes y el envío de datos por radio
- en la página 40 se inicia la segunda fase, que implica trabajar el concepto de listas, fácil de comprender con una hoja de cálculo y difícil de comprender al realizar la programación de las mismas

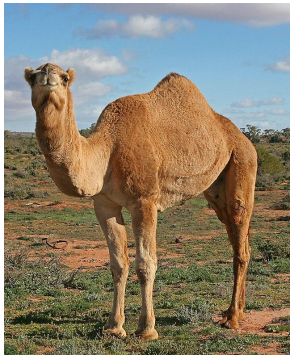
La segunda fase se puede evitar si entregamos el programa finalizado al alumnado, para que puedan probar el resultado final esperado en el reto

QUÉ NECESITAMOS

Hemos de crear varias listas:

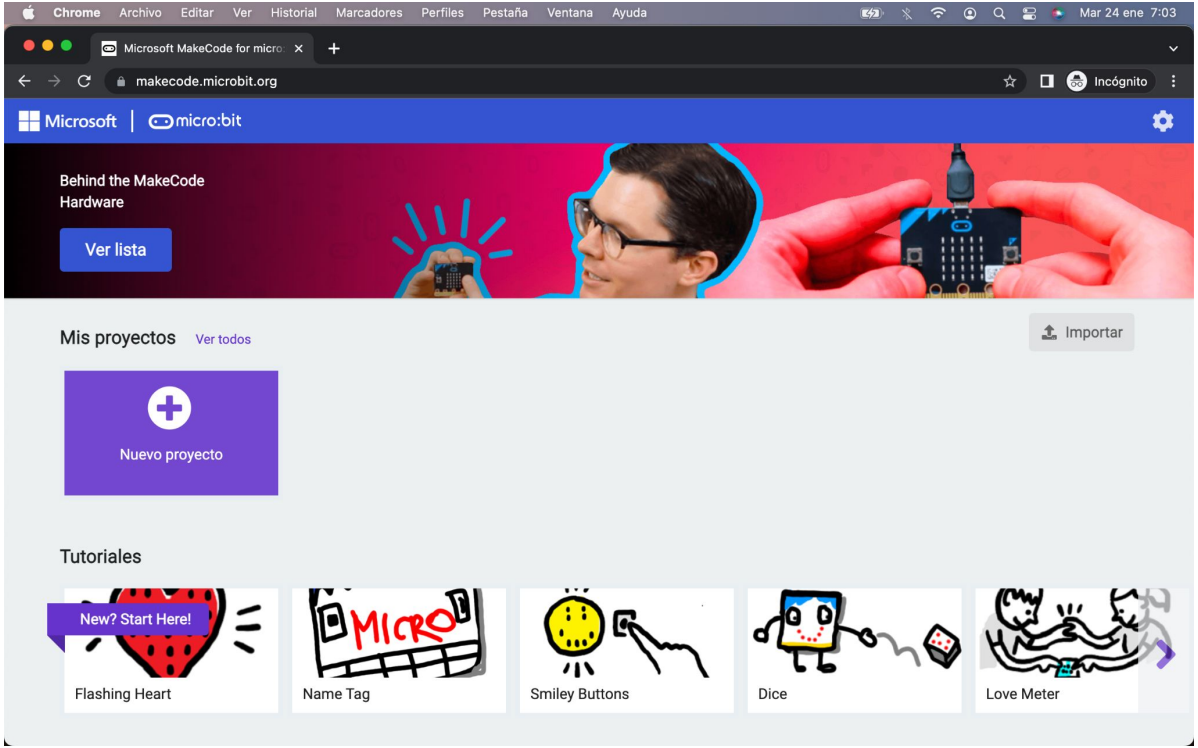
- Lista de imágenes de animales
- Lista de participantes
- Lista de animales elegidos

Vamos a empezar nuestro programa !!



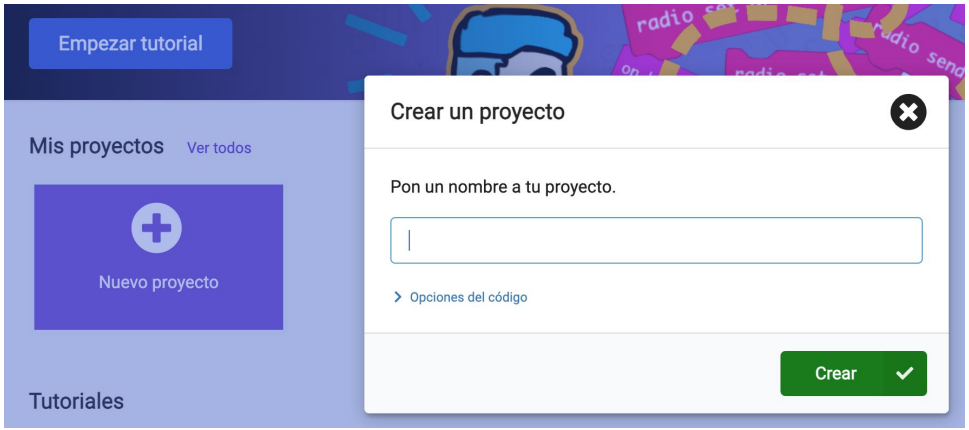

Empezando con MakeCode

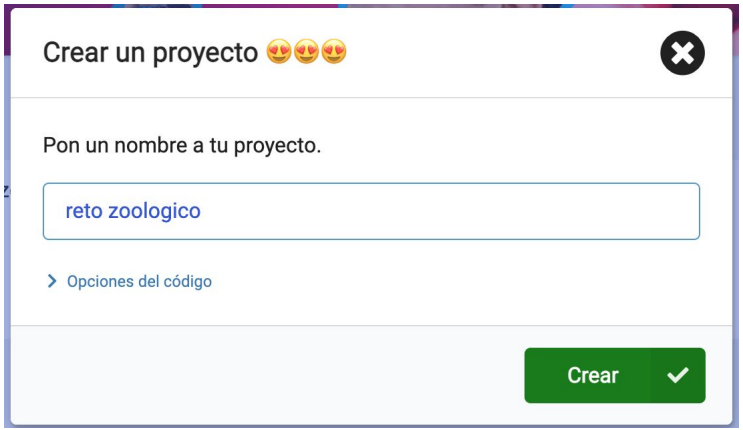
Escribimos en el navegador **makecode.microbit.org**



Nuevo proyecto MakeCode

Si pulsamos en “**Nuevo proyecto**” podremos dar un nombre a nuestro proyecto





Panel de trabajo y simulador de MakeCode

Opciones

Simulador

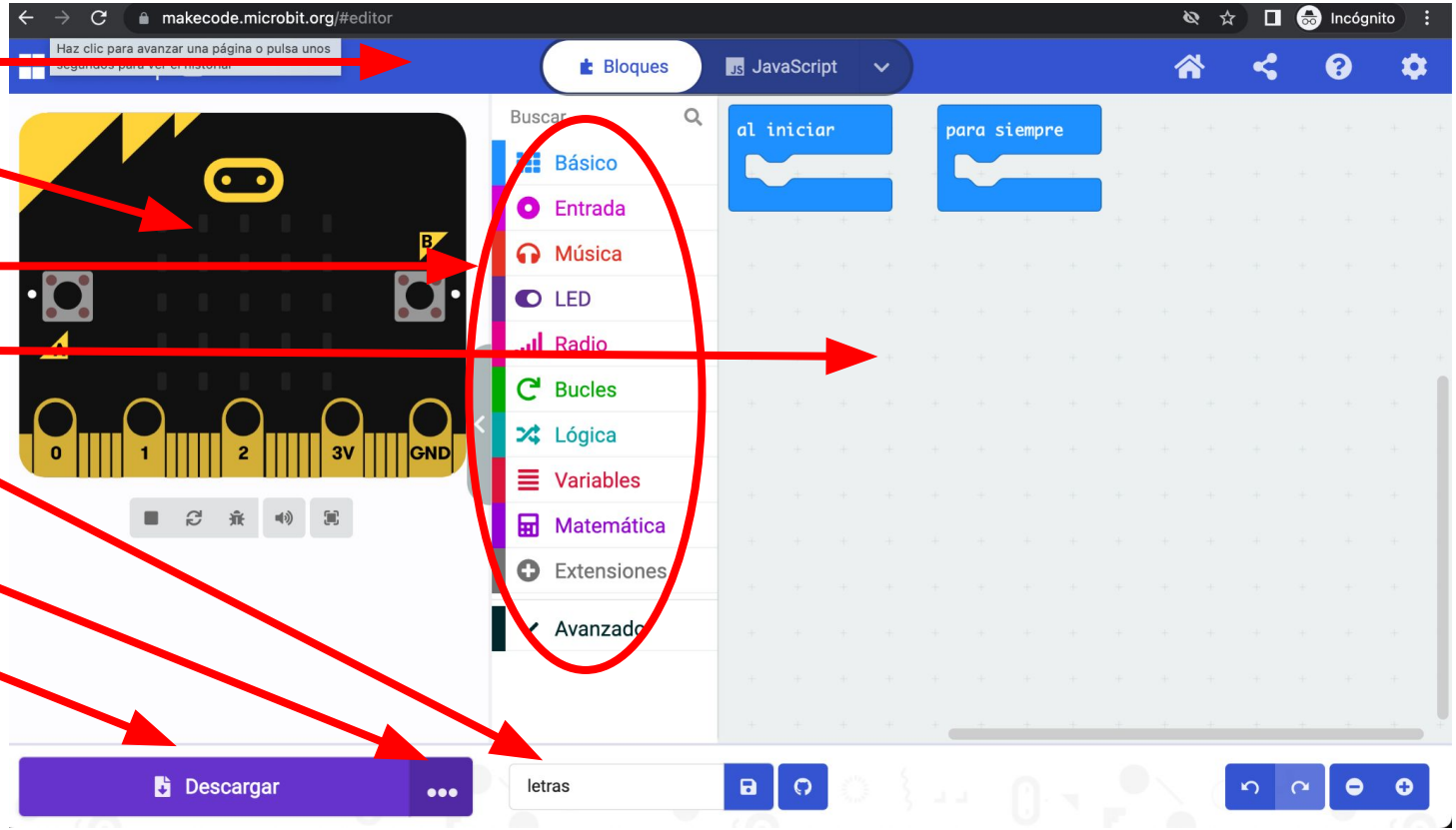
Bloques

Código

Nombre

Conexión

Descarga

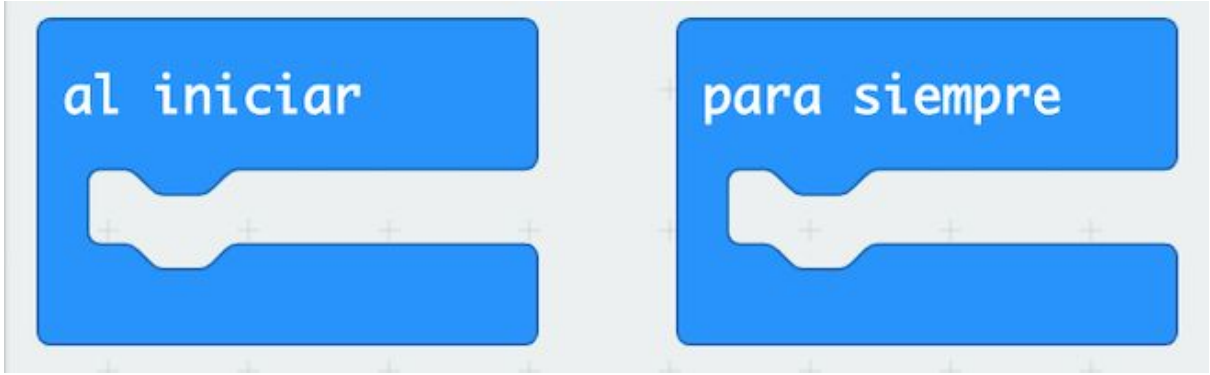


Código por defecto

El código de muestra tiene dos partes:

- Al iniciar - aquí incluiremos el código que se ejecuta una vez
- Para siempre - aquí tenemos el código que se ejecuta repetidamente

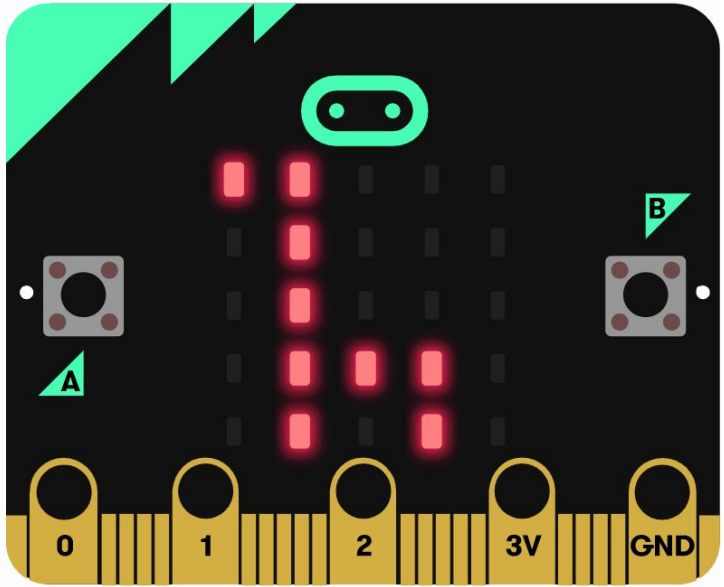
El simulador de la izquierda se activará cuando escribamos un programa



Pensemos un poco

¿Podemos dibujar un **animal** con **micro:bit**?

Vamos a probar uno. ¿qué ves?



Girafa

¿Qué programa necesitamos?

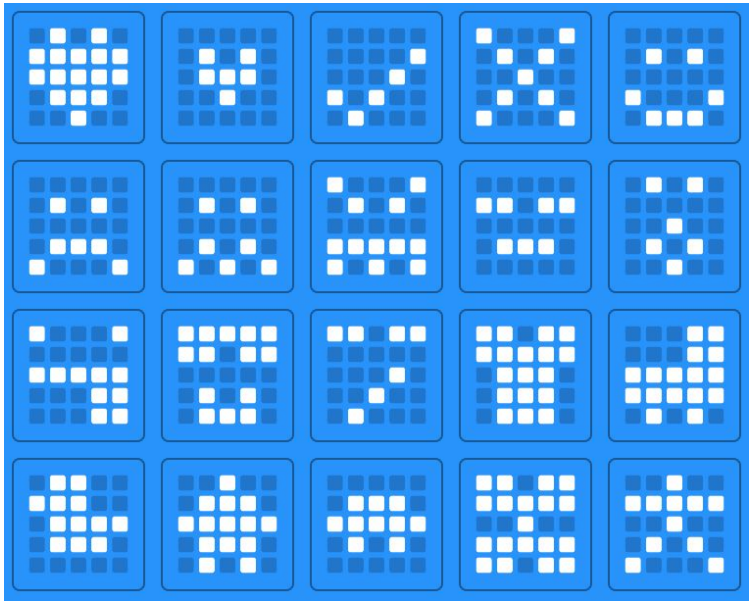
Uno realmente simple

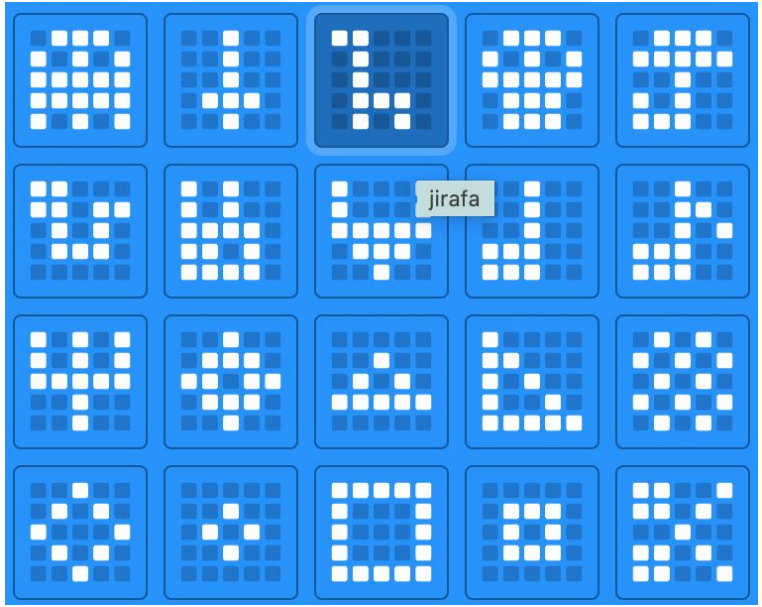
micro:bit tiene algunos dibujos disponibles en la instrucción **mostrar icono**

Iconos

Imágenes prediseñadas o iconos.

¿identificamos los dibujos?

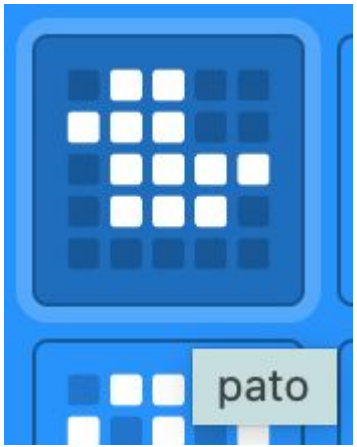


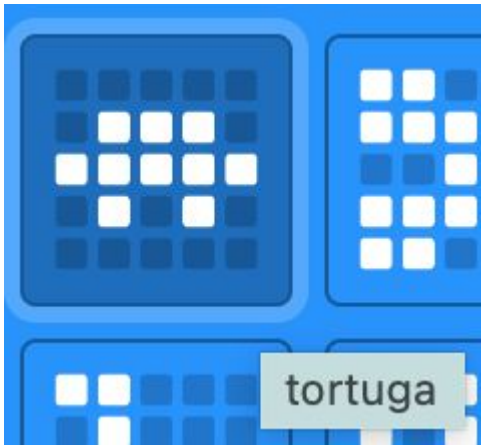


Observemos

Ves pasando el cursor sobre los iconos y van apareciendo los nombre de los dibujos.

Algunos son evidentes, otros complicados y algunos quizás no sabremos verlos





¿Podemos dibujar?

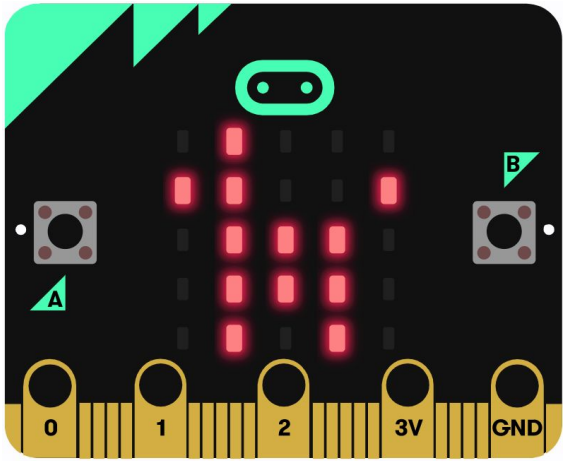
¿Conocemos otra forma de hacer un dibujo con leds?

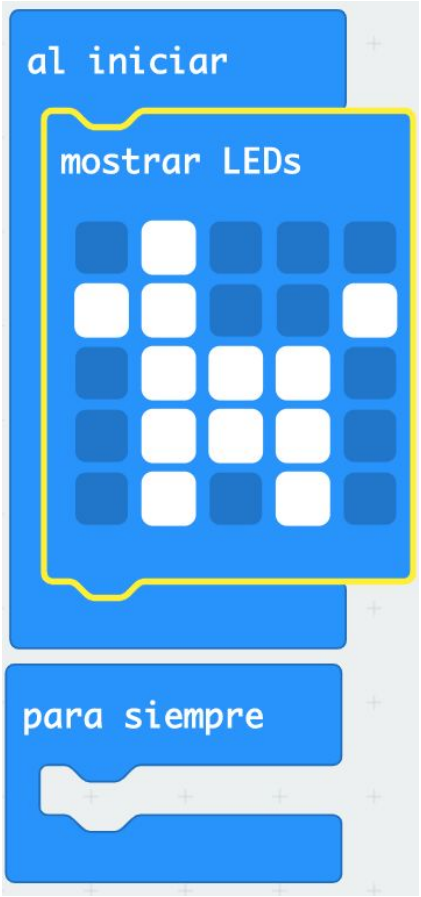
Seguro que sí, lo hemos usado en otros programas

Dibujos

El bloque **mostrar LEDs** nos permite hacer dibujos en pantalla

Huy, ¿si parece un animal?





Prueba de hacer otros dibujos de animales

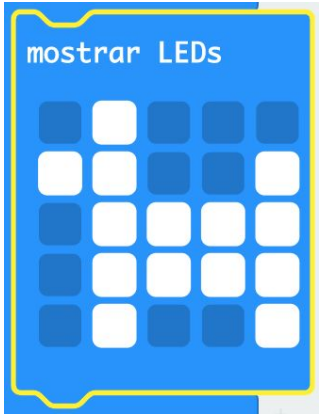
Algunos animales

dromedario

elefante

conejo

perro pequeño



perro grande

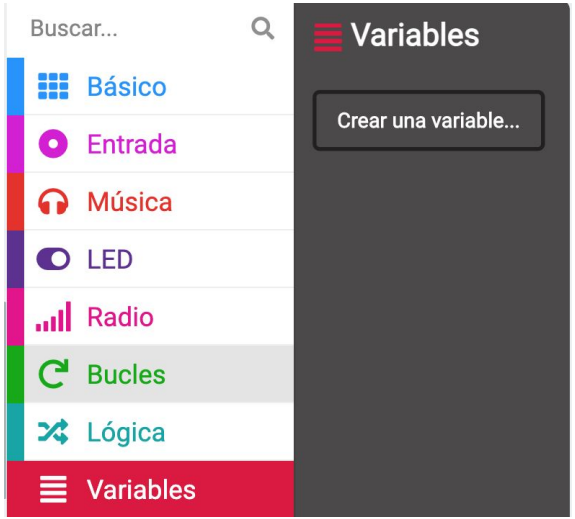
Seguro que has encontrado otros dibujos. Compártelos

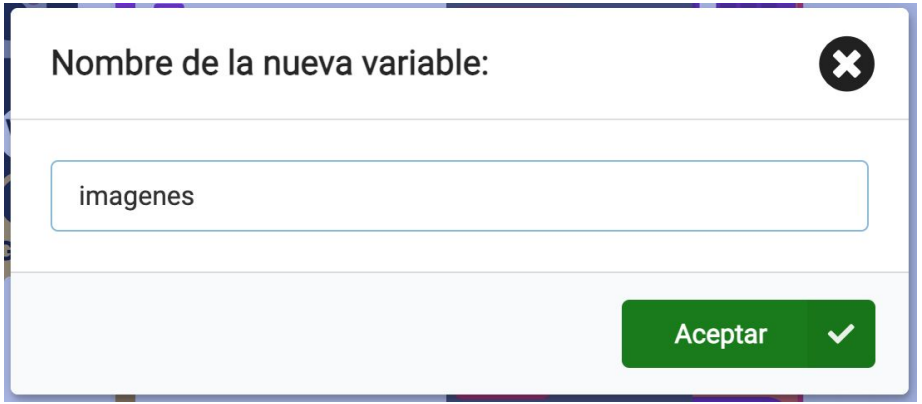
Lista de animales

Vamos a utilizar una **lista** de animales para que sea más fácil identificarlos en el programa.

La **lista** se guarda en una **Variable**

Empezamos creando la variable **imagenes**





Listas (mal traducido por Arreglos)

En los bloques de **Avanzado** encontramos los que corresponden a **Arreglos** (**Listas**)

La primera opción nos permite crear una lista de valores

Crear imágenes

Queremos crear una **lista de imágenes**.

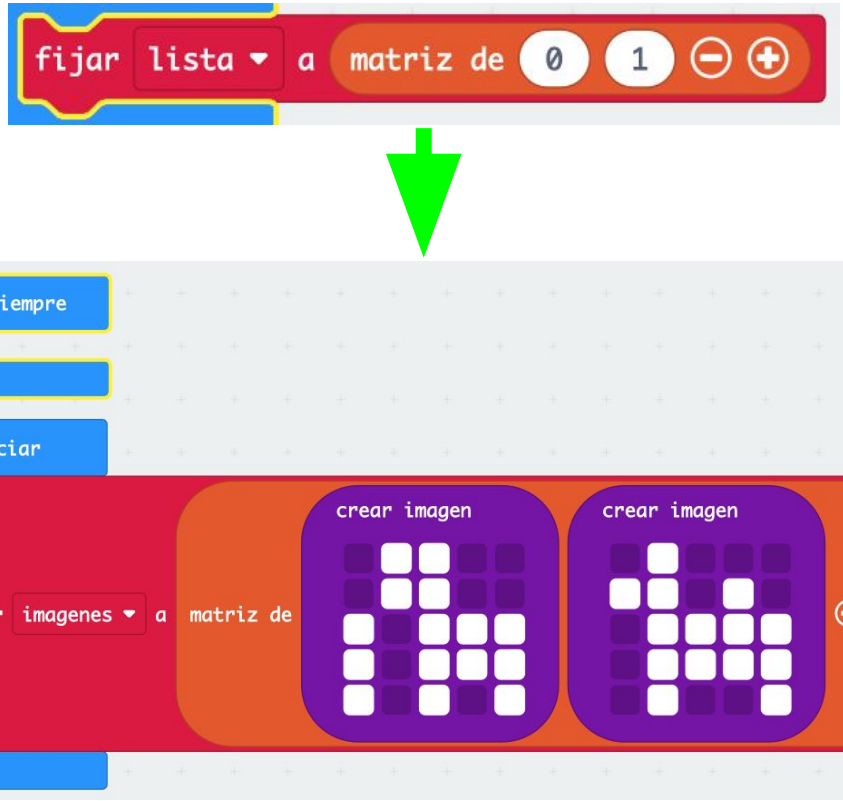
Encontramos un bloque de **Imágenes** que nos permite **crear imagen**

Vamos a utilizarlo para dibujar los diferentes animales del juego y ponerlos en la lista

Lista de imágenes

Cambiamos el nombre de la variable y ponemos **imagenes**

En cada valor ponemos una imagen con el bloque de **crear imagen**

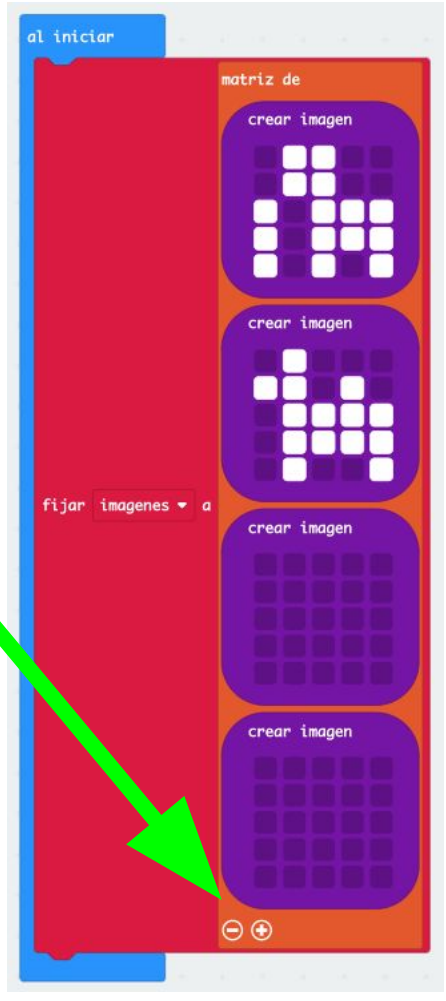


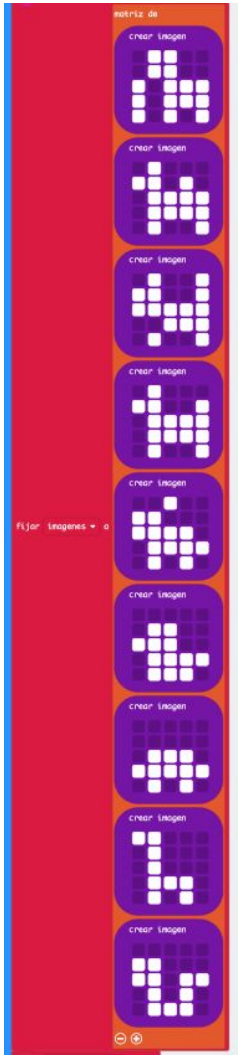
Lista de imágenes (2)

Pulsando el botón **+** podemos ampliar la lista y nos aparecen directamente los espacios para crear las nuevas imágenes

Vamos dibujando los animales que hayamos elegido

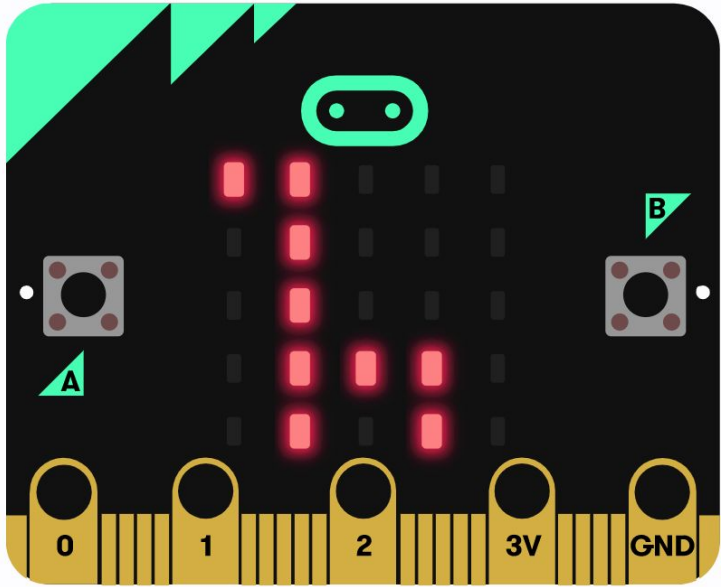
Marcamos con ✓ el trozo de programa que ya es definitivo





Pensemos un poco

Y ahora, ¿cómo conseguimos que aparezcan los animales en la pantalla?



Visualizar las imágenes

Al principio hemos dicho que queremos poder elegir una imágen de animal.

- Necesitamos una **variable** para guardar el **código del animal**
- El código empieza en cero (lo habitual en programación)
- Pulsaremos un botón para cambiar el animal (incrementaremos el código de animal)
- El código no puede ser mayor que el número de imágenes disponibles
- Visualizaremos el nuevo animal al pulsar el botón

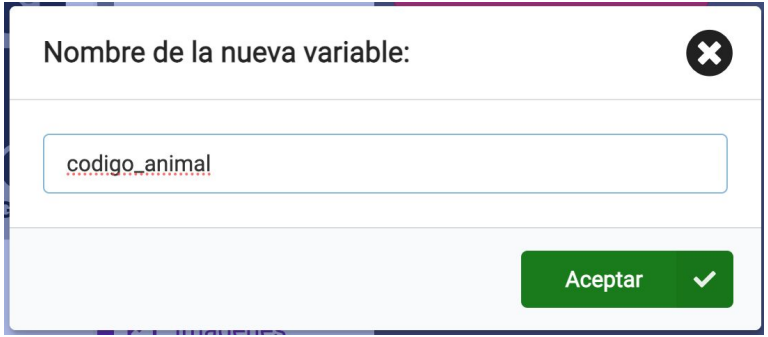
Veamos cómo se programa todo lo anterior !!

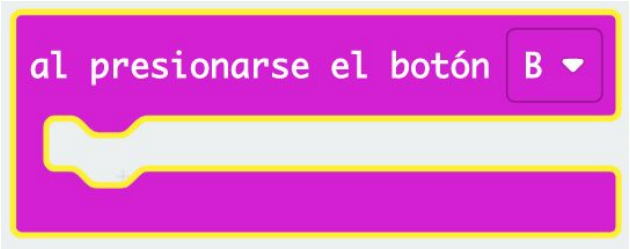
Elementos para visualizar

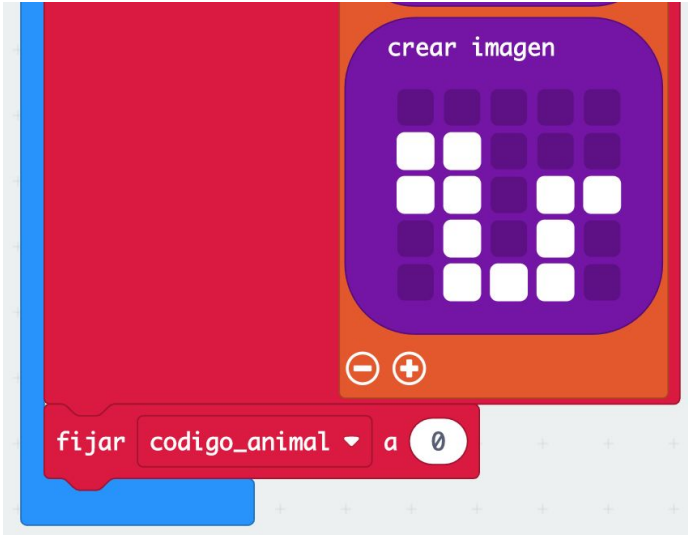
Empezamos creando la **variable** para guardar el código con el nombre **codigo_animal**

Definimos el valor 0 para la variable **codigo_animal** al final del bucle **al iniciar**

Elegimos el **botón B** para cambiar de animal







Cambio de animal

Para **elegir el animal** debemos aumentar el valor de **codigo_animal**

El código no puede ser mayor que el número de imágenes que tenemos. ¿podemos saber cuántas imágenes hay sin contarlas?

Tenemos un bloque que nos da la **longitud** de la **lista de imagenes**

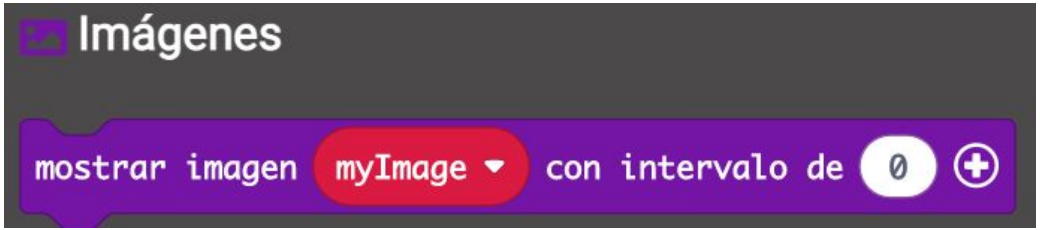
Control de límites

Ahora nos falta comparar el **codigo_animal** con la longitud de la lista de **imagenes** y poner el **codigo_animal** a cero si hemos llegado al límite.

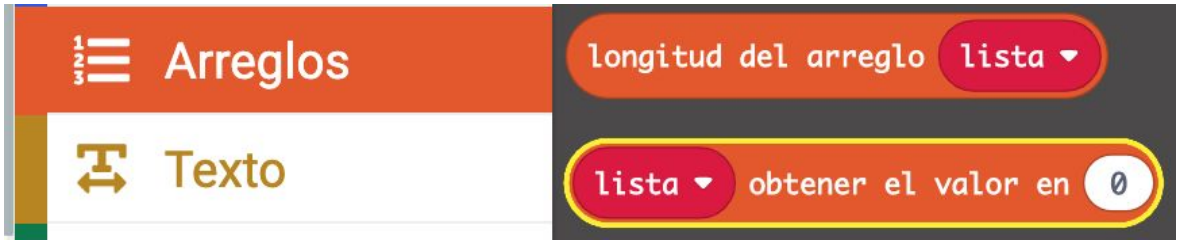
Recordemos que si tenemos **9 animales** el **código irá de 0 a 8** y por tanto debemos comparar con la opción **mayor o igual** al **número de imágenes**

Visualizar el animal

Sólo nos queda visualizar el animal elegido. Encontramos dentro de **Imágenes** el bloque adecuado



y en **Arreglos** (Listas) la posibilidad de elegir una imagen de la lista de imágenes indicando su **posición** (**obtener el valor en 0)**



Botón B finalizado

Este es el código completo para el botón B. Al pulsar el botón B las imágenes irán desfilando por la pantalla del **micro:bit**

Observemos

Miremos ahora la simulación

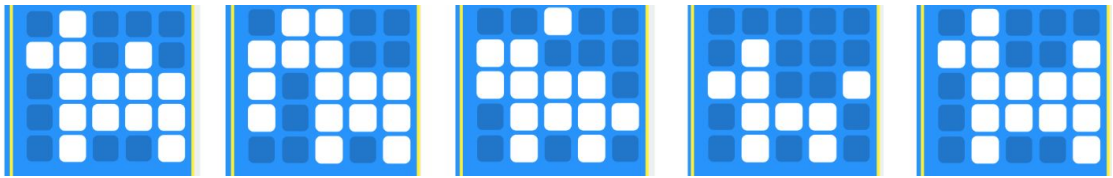
Cada vez que pulsamos el botón B aparece una nueva imagen de animal por la pantalla

El animal que elijamos es el que formará parte de nuestro zoológico

Pensemos un poco

Y ahora, ¿cómo hacemos para saber qué animal ha elegido cada participante?

Debemos comunicar las tarjetas **micro:bit** entre ellas, y lo podemos hacer por **radio**

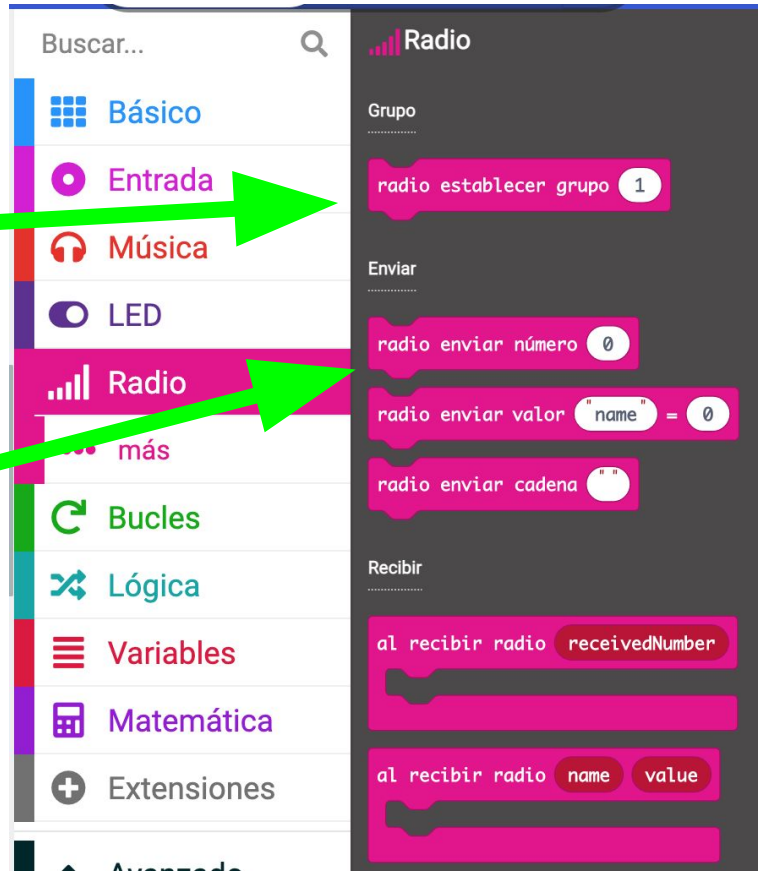


Radio

Tenemos unos bloques dedicados a la **Radio**

El primer bloque nos permite crear un **grupo de radio** que será el que utilizaremos para comunicarnos. Es como elegir una clase donde poder hablar sin molestar al resto de alumnas y alumnos de la escuela

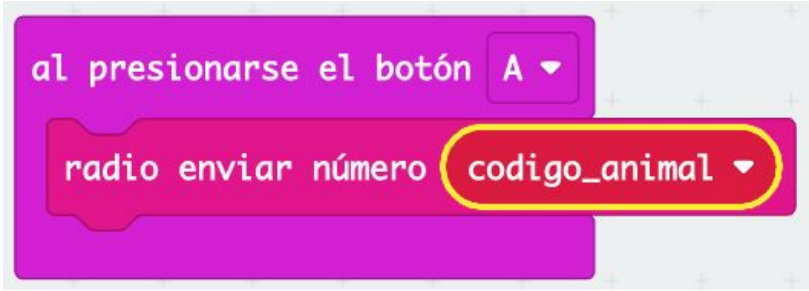
El segundo bloque nos permite **enviar un número**. Podemos enviar el **código del animal** al resto de participantes



Probando la radio

En el bucle **al iniciar** creamos nuestro **grupo de radio** y le ponemos el número **13**

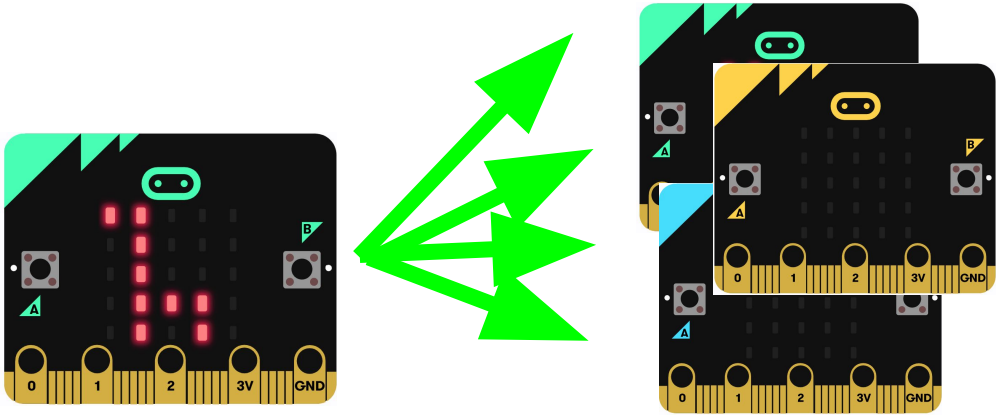
Elegimos el **botón A** para enviar nuestro código de animal al resto de participantes



Pensemos un poco

Ya sabemos cómo enviar un código.

¿alguien lo recibe? ¿cómo lo ve?



Recibir por radio

Tenemos unos bloques dedicados a **recibir** por **Radio**

Podemos recibir un número de forma sencilla y hacer algo con lo que recibamos, como por ejemplo visualizar el número en la pantalla

Observemos

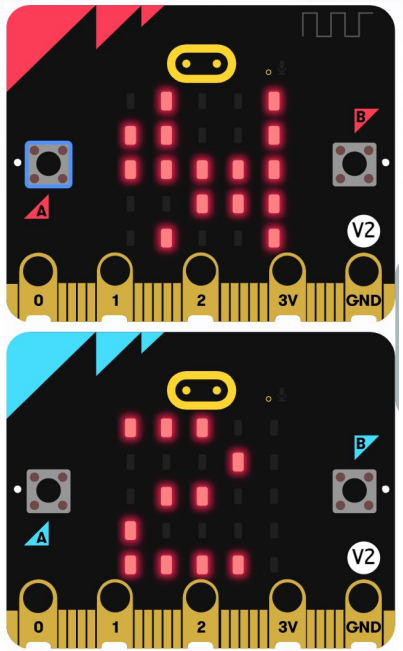
Miremos ahora la simulación

Pulsamos el botón B para elegir la imagen del animal en la pantalla

Pulsamos el botón A y aparece un segundo **micro:bit.** Volvemos a pulsar **el botón A** y veremos en la segunda pantalla el código de nuestro animal.

¿qué ocurre al pulsar uno de los botones **B**?

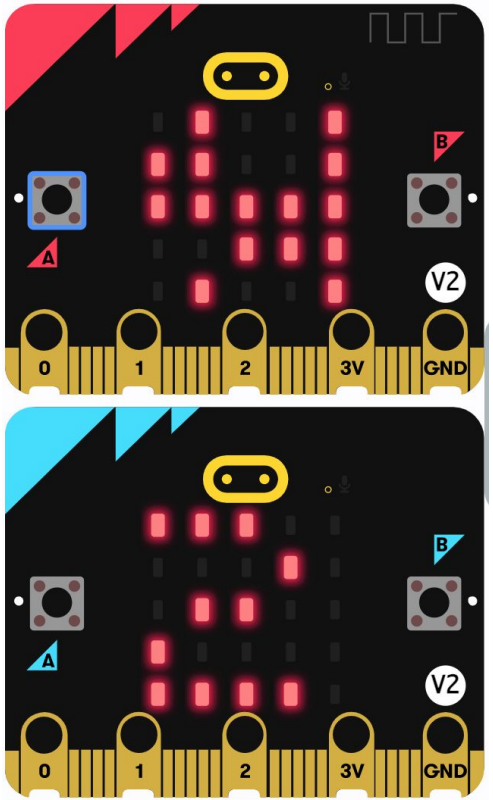
¿y al pulsar uno de los botones **A**?



RESUMEN

Tenemos un programa que nos permite visualizar un animal y enviar su código por radio a otras tarjetas.

Ahora es el momento de copiarlo a nuestra tarjeta **micro:bit** y probarlo de verdad.



Recuerda los pasos:

1. Conectar el micro:bit al ordenador
2. Emparejar el **micro:bit** (**Connect device**)
3. Descargar el código al **micro:bit**

Pensemos un poco

¿podríamos visualizar el animal de otro participante en lugar de su código?

Como que la lista de animales está en las dos tarjetas, podríamos cambiar la acción al recibir un valor por radio y veríamos el animal

RESUMEN

Podemos probar el programa sólo en el simulador o bien copiarlo a nuestra tarjeta **micro:bit** y probarlo de verdad.

Recuerda los pasos:

1. Conectar el micro:bit al ordenador
2. Emparejar el **micro:bit** (**Connect device**)
3. Descargar el código al **micro:bit**

Pensemos un poco

Bueno, ya hemos visto que es posible comunicar varias tarjetas **micro:bit** y también habremos observado que:

- no sabemos quién envía cada animal
- no sabemos cómo contar los animales
- no sabemos si son iguales o diferentes

Es decir, **nos falta información**

¿podemos acumular estos datos?

¿Dónde guardamos los datos?

Necesitamos guardar los datos, y para ello emplearemos una **lista de códigos de participante** y otra **lista con los animales** que ha enviado cada participante

Los participantes pueden cambiar el animal elegido en cualquier momento. Deberemos actualizar la lista de animales cada vez que se reciba un animal

Además necesitaremos una **lista con la suma** de cada tipo de animal que está seleccionado en cada momento, para poder decidir si ya tenemos el **zoológico completo**

¿Qué queremos conocer?

Queremos saber:

1. quien envía un animal (código del participante)
2. qué animal envía (código del animal)

Con estos datos podremos hacer una **lista de animales** y podremos saber si son **iguales o diferentes** y cuantos tenemos de cada uno

Tratemos de hacer una tabla animales-participantes

Tabla de animales elegidos

Supongamos que tenemos esta situación. Es fácil contar los animales y saber si tenemos alguno repetido (suma > 1)

	A	B	C	D	E	F	G	H	I
1			PARTICIPANTE						
2			P1	P2	P3	P4	P5		
3	ANIMAL	CODIGO						TOTAL	
4	ELEFANTE	0						0	
5	DROMEDARIO	1	X					1	
6	GATO	2						0	
7	PERRO	3		X		X		2	Repetido
8	CONEJO	4			X			1	
9	PATO	5						0	
10	TORTUGA	6						0	
11	GIRAFA	7					X	1	
12	SERPIENTE	8							

Listas necesarias

Para guardar los datos anteriores necesitamos tres listas:

1. **codigos_participantes** para guardar el código de cada participante
   se guardan en orden a medida que se reciben por radio
   hay tantos valores como participantes (son 5 en este ejemplo)
2. **animal_del_participante** para guardar los animales
   se guardan en el mismo orden que los anteriores
   hay tantos valores como participantes (son 5 en este ejemplo)
3. **suma_de_animales** para poder contar cuantos animales hay de cada
   se guardan en el mismo orden que las **imagenes** que tenemos
   hay tantos valores como imágenes (son 9 en este ejemplo)

¿Cómo lo hacemos?

Queremos saber:

1. quien envía un animal (**código del participante**)
2. qué animal envía (**código del animal**)

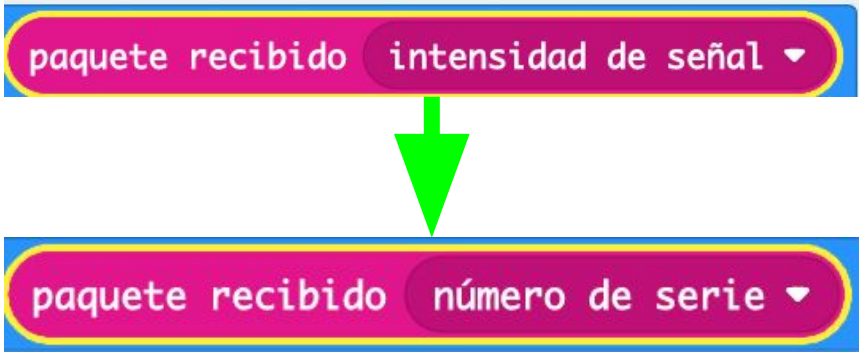
Para poder conocer el **código del participante** debemos **activar** una opción de **Radio**, en el apartado **más**

El **código del animal** ya sabemos cómo enviarlo

Código de radio

Activamos el **código del participante** al principio del bucle **al iniciar,** justo después de definir el **grupo de radio**

El **código del participante** lo obtendremos con el bloque que nos da el **número de serie** (elegimos **paquete recibido** en **Radio** y cambiamos a **número de serie**)



Nuevas variables

Vamos a crear ahora algunas variables que necesitamos para hacer el programa. Las iremos explicando con detalle cuando las utilicemos

La variable **numero_de_animales** deberá tener el **número de imágenes de animales** que tenemos en nuestra **lista de imagenes** (en este programa son 9)

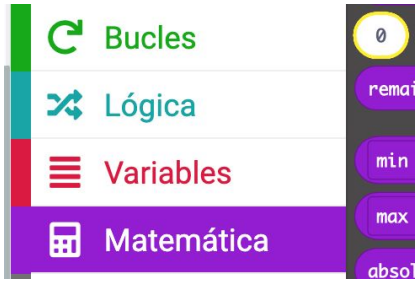
Espera !!, lo pondremos más adelante

Variables con listas

Para poner una **lista en una variable** debemos ir a **Arreglos (listas)** y elegir

La matriz de **suma_de_animales** empieza vacía mientras que **codigos_participantes** y **animal_del_participante** empezaran con un valor cero, para reservar un espacio para guardar nuestros datos como participante

Pulsamos el botón **+** y en **Matemática** encontramos el valor **0**



Valor de las variables

Al final del bucle **al iniciar** definimos el valor de las variables. Las que son listas las creamos con el bloque **matriz** que encontramos en **Arreglos**

Lista completa de variables

Vamos a crear algunas variables más y ya tendremos todas las que necesitaremos en nuestro programa

Las variables **animal_enviado** y **animal_recibido** facilitan la identificación de los animales

La variable **codigo_recibido** guardará el número de serie recibido por radio que identifica al participante

Las variables **indice** y **indice_animal** facilitarán la posición en las tablas de animales y participantes

Y por fin una variable **repetido** para controlar el juego

Botón A finalizado

Modificaremos el código del **botón A** con un nuevo bloque, para guardar en una variable el código del **animal_enviado** por radio

De esta forma, aunque vayamos cambiando el animal que tenemos en la pantalla, para poder enviar uno nuevo, siempre sabremos el código del animal que hemos enviado por radio al resto de participantes

Estado del programa

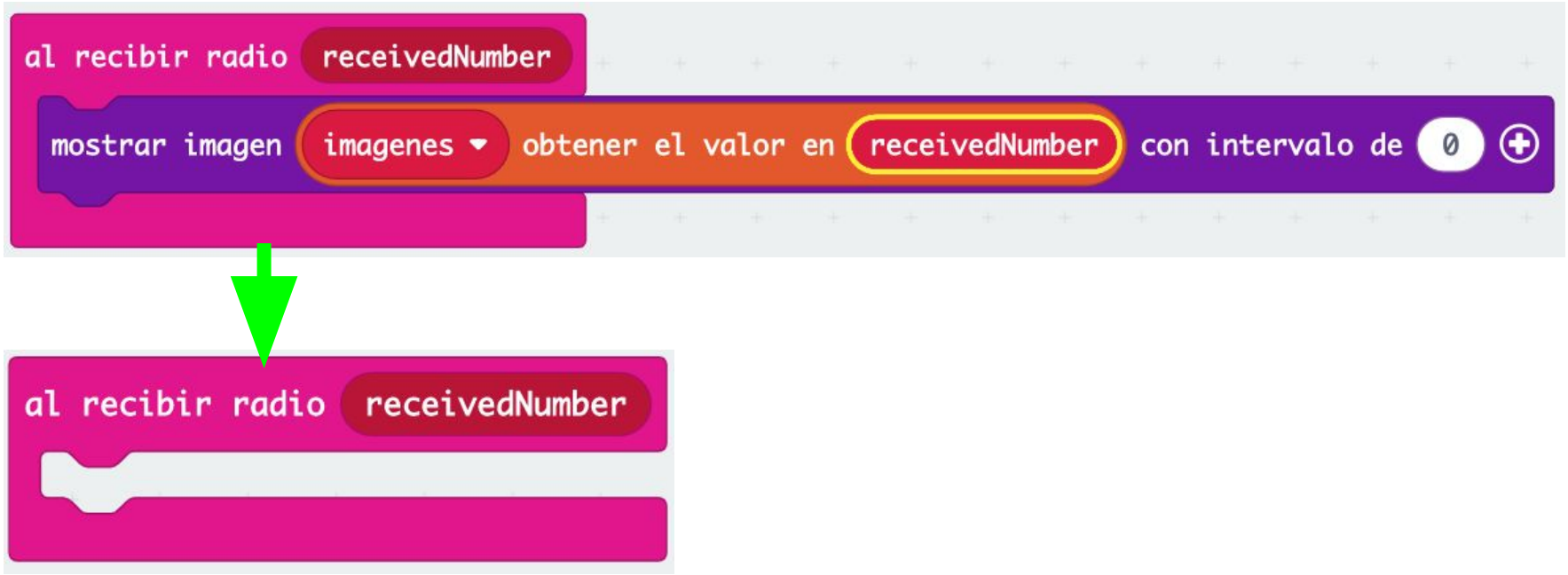
Ya tenemos completo el programa de:

- **botón A**
- **botón B**
- **al iniciar**

Vamos ahora a ver las acciones que se deben realizar al recibir un código por **radio**.

Quitamos un bloque

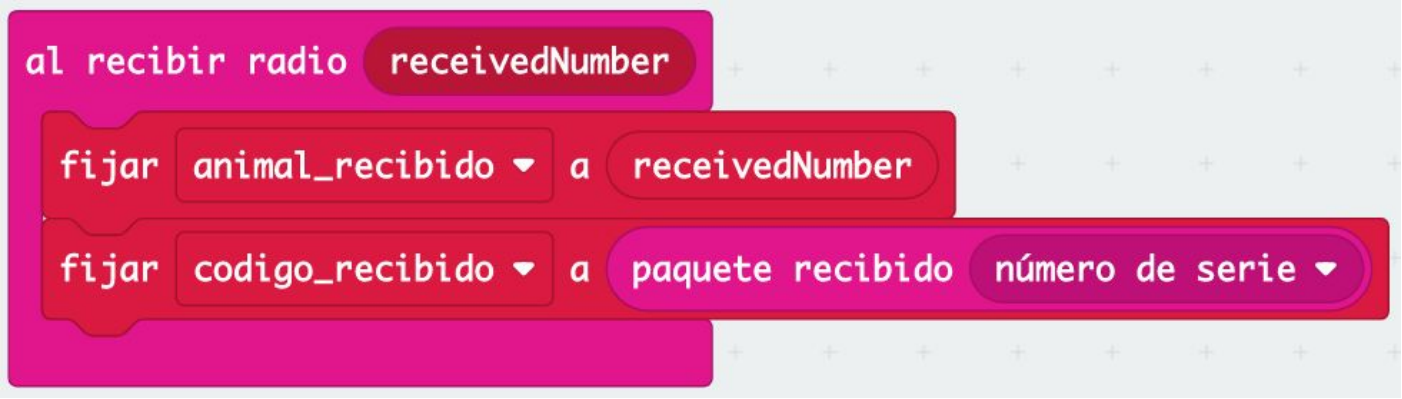
Vamos ahora a modificar las acciones de **radio** Quitamos la instrucción de **mostrar imagen**



Recibir código animal y participante

Lo primero que necesitamos guardar es el **código del participante** (ya hemos comentado antes que es el número de serie) y el **código del animal** que nos han enviado por radio (pulsando el botón A en otro **micro:bit**)

Los guardamos como en la imagen



Buscar el participante

Debemos añadir el **código del participante** a nuestra lista de participantes, pero sólo si no existe

¿cómo buscamos un valor en una lista?

En **Arreglos** encontramos el bloque

que nos dice en qué posición de la lista está el valor que buscamos. Si existe nos dice en qué posición está el valor (0, 1, 2, …) y si no existe nos devuelve **-1**

Indice del participante

Buscamos el **código_recibido** en nuestra lista de **codigos_participantes** y lo guardamos en **indice**

Si no existe (valor -1) lo deberemos añadir al final de la lista

Indice real del participante

Volvemos a buscar el **código_recibido** que ahora ya existe en la lista

¿dónde guardar el animal recibido?

Ya sabemos en qué posición (0, 1, 2, ..) de la lista de **codigos_participantes** tenemos al participante

Ahora debemos guardar el animal que hemos recibido (**animal_recibido**) en la lista de animales **animal_del_participante**

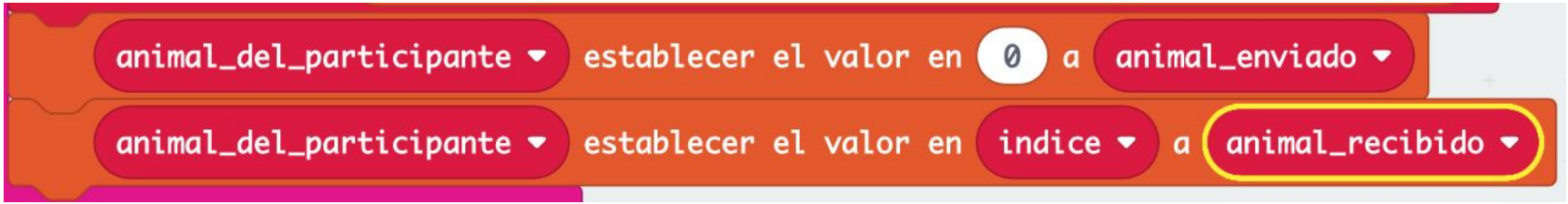
OJO ! Debemos guardarlo en la **misma posición** en que tenemos el participante para saber de quién es cada animal y poder cambiarlo si el participante envía un nuevo animal

Guardar animales

Vamos a guardar el **animal_enviado** en la **posición 0** de la lista **animal_del_participante** (para poder sumarlo con facilidad)

El animal **animal_recibido** hemos de guardarlo en la misma posición que el código de participante. La posición la tenemos en la variable **indice**

Los bloques necesarios los encontramos en **Arreglos** (**Listas**)



¿cómo se suman los animales?

Tenemos dos listas que nos permiten guardar los **participantes** y los **animales**

Ahora nos falta contar (sumar) cuantos animales tenemos de cada tipo (columna **TOTAL** del ejemplo)

Hemos de usar una lista **suma_de_animales** para agruparlos

La lista tiene tantos valores como imágenes y el índice tendrá **valores de 0** al **numero_de_animales menos 1** (0 a 8) tal como vemos en el **CODIGO** del ejemplo

	A	B	C	D	E	F	G	H	I
1			PARTICIPANTE						
2			P1	P2	P3	P4	P5		
3	ANIMAL	CODIGO						TOTAL	
4	ELEFANTE	0						0	
5	DROMEDARIO	1	X					1	
6	GATO	2						0	
7	PERRO	3		X		X		2	Repetido
8	CONEJO	4			X			1	
9	PATO	5						0	
10	TORTUGA	6						0	
11	GIRAFA	7					X	1	
12	SERPIENTE	8							

Puesta a cero

Empezamos poniendo a cero los contadores o sea las sumas de animales

Aparece aquí una nueva variable de nombre fijo **index**, que utilizaremos solamente dentro de un bucle, y que debe tomar valores desde 0 hasta el **numero_de_animales menos 1** (0, 1, …, 8)

Ponemos a cero cada valor de la lista **suma_de_animales**

Pensemos un poco

¿cómo contamos los animales?

Tenemos una lista **animal_del_participante** que tiene los animales. Bien!!

¿cuantos valores tiene esta lista?

Tantos como participantes, claro

Y podemos saber este número con el bloque **longitud del arreglo** (lista) de **animal_del_participante**

Buscando el animal

Ahora se complica un poco. Buscamos el animal de cada participante que está en la lista

Podemos usar un bucle con todos los valores (0, 1, ??, ??)

Como empieza en cero, el último será uno menos que el número de animales en la lista.

Lo escribiremos así

Buscando el índice del animal

Los códigos de los animales se obtienen de la lista con

y como queremos usar el código del animal como índice, lo guardamos en la variable **indice_animal**

Nuestro bucle queda de la forma siguiente

Contado animales (1)

Para contar animales añadiremos uno cada vez que aparezca. Ya tenemos el código del animal en la variable **indice_animal**

Ahora hemos de aumentar la suma

suma = suma + 1

El valor actual lo encontramos en la lista de suma con

Contado animales (2)

El nuevo valor lo guardaremos en la lista de suma con

Como ya hemos dicho hemos de aumentar en uno la suma

suma = suma + 1

y la expresión completa será pues

Contado animales (3)

El bucle que estábamos escribiendo ya está completo

Ya tenemos una lista con la **suma_de_animales**

Pensemos un poco

Ya tenemos una lista con la **suma_de_animales**

¿cómo sabemos si están repetidos?

Claro, si el valor es **mayor que uno** el animal está **repetido**. Si el valor es cero nadie ha elegido el animal y **si es uno** el animal puede formar parte de nuestro **zoológico**

Lo escribimos así

Verificar repeticiones

Empezamos definiendo la variable **repetido** a **NO** y dentro del bucle verificamos si algún valor es mayor que uno, entonces ponemos un **SI**

Pensemos un poco

Bueno, parece que ya sabemos si algún animal está repetido o no. Pero el resto de participantes no lo sabe. Podemos informar a todas las tarjetas

Este valor se usará en el bloque “para siempre”

Con este bloque ya tenemos completa la acción al recibir un animal por radio

Acción completa al recibir el código del animal de un participante

Recibir dato de animal repetido

Antes hemos enviado el valor de la variable **repetido**

Ahora nos falta hacer una acción al recibir este dato. Simplemente lo guardamos en la variable **repetido** de **todos** los **micro:bit**, y lo usaremos más tarde dentro del bloque “para siempre”

Pensemos un poco

Bien, parece que nuestro programa:

- nos permite elegir un animal
- nos permite visualizarlo
- podemos enviarlo a otros participantes
- sabemos si algún animal está repetido

pero

¿cómo sabemos si ya tenemos el zoológico completo entre todos los participantes?

¿qué hacemos al completar el zoológico?

Si ya no hay animales repetidos podemos hacer sonar una **música** y presentar un **corazón** en pantalla

Como esto puede ocurrir en cualquier momento, por la acción de cualquier participante, conviene poner esta acción en el bucle **para siempre**

Buscamos nuestro bloque actual y borramos la acción

Propuesta

Si no hay animales repetidos suena una música de fondo y a la vez aparece un corazón en pantalla

Pensemos un poco

Genial !! ya sabemos que hemos conseguido el RETO inicial de conseguir un **zoológico**

Pero ahora no vemos los animales que ha elegido cada participante.

Vamos a presentar de forma intermitente los animales que forman el zoológico, cada uno en su propia tarjeta

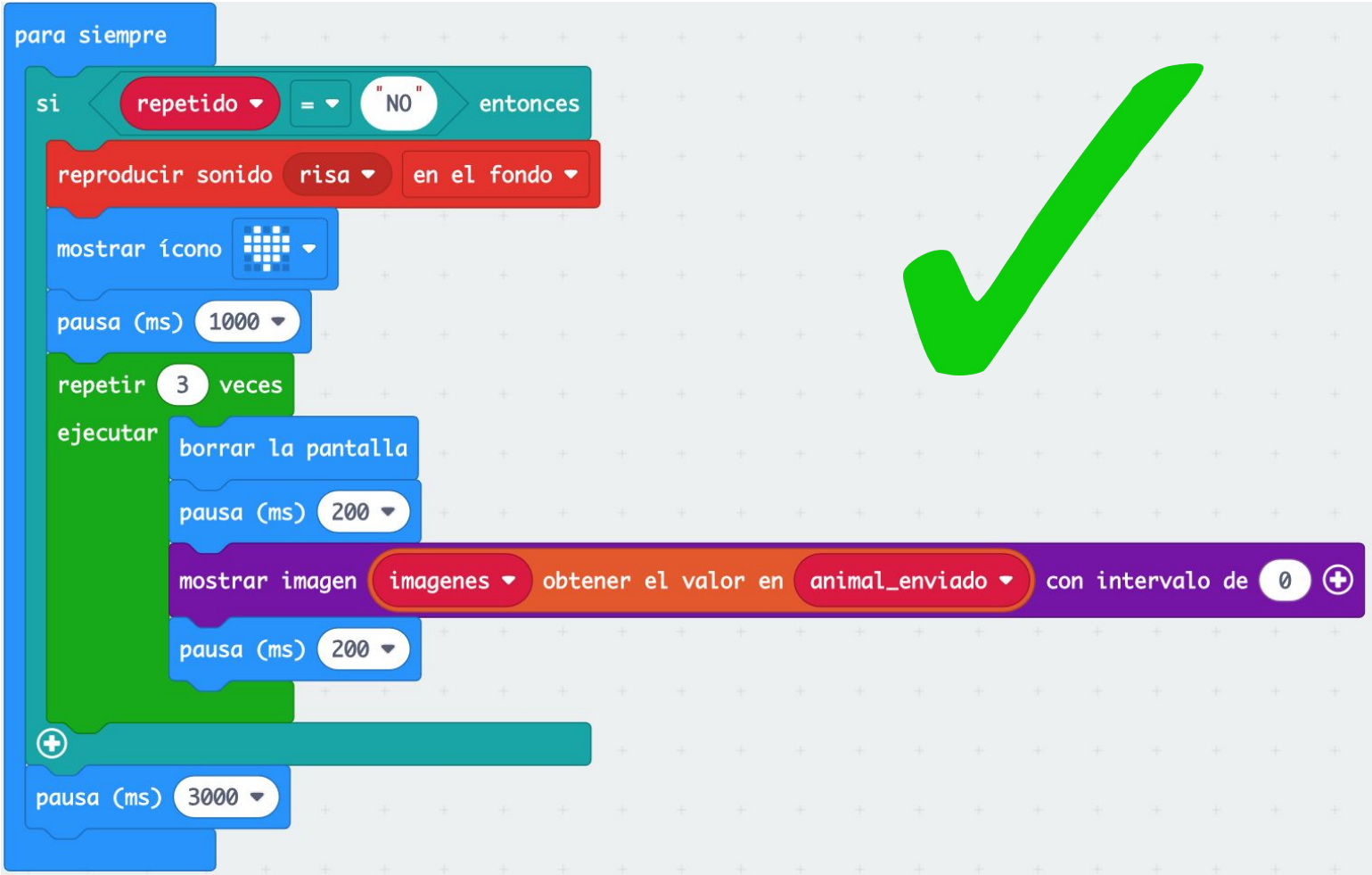
Animales intermitentes

Si **borramos** la pantalla, **esperamos** un tiempo corto, enseñamos el **animal** que hemos enviado, **esperamos** un tiempo y **repetimos** varias veces esta operación veremos el animal de forma **intermitente**

Acabando el programa

Juntamos ahora todas las partes del bloque “para siempre”

Hemos de añadir unas pausas al programa, para poder ver bien las imágenes que se presentan en la pantalla

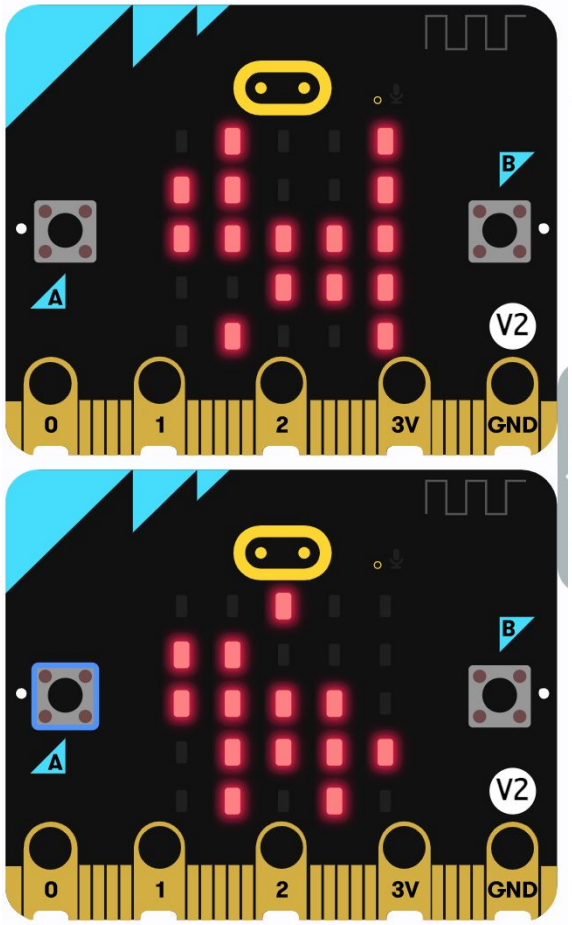


Acción completa de **para siempre**

Observemos

Ya tenemos dos animales en nuestro pequeño **zoológico**

Si copiamos el programa a varias tarjetas **micro:bit** tendremos más animales



RESUMEN

El programa ya está completo.

Ahora es el momento de copiarlo a nuestra tarjeta **micro:bit** y probarlo de verdad.

Recuerda los pasos:

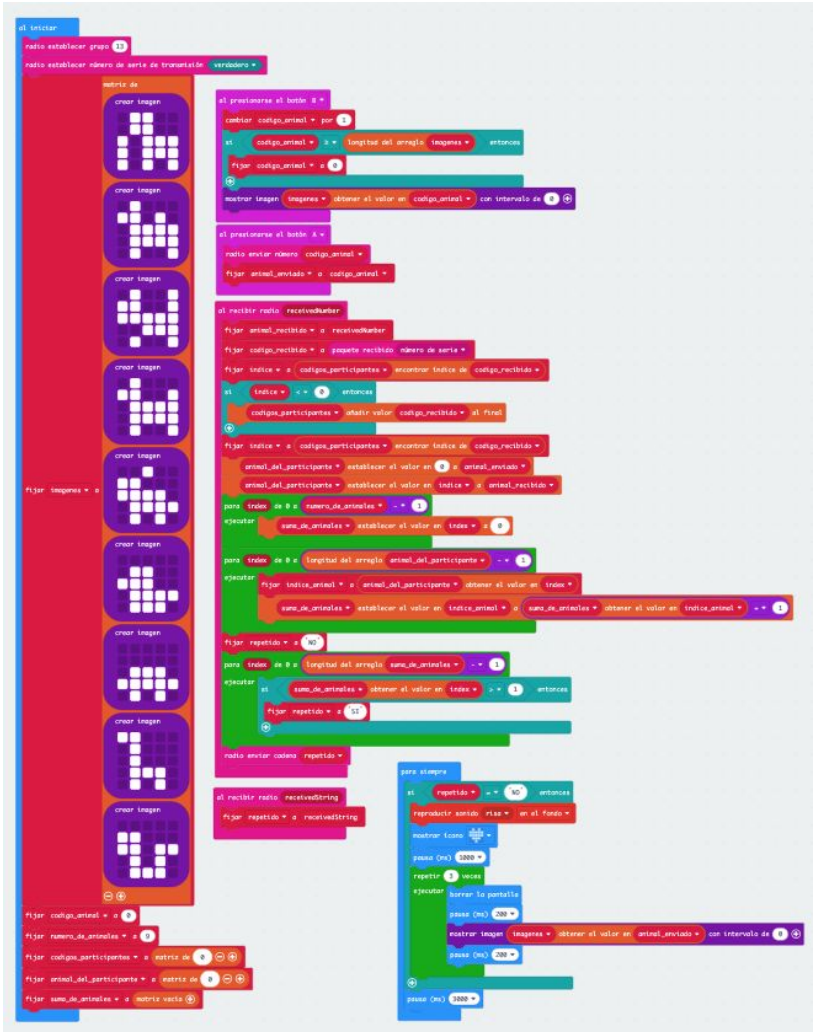
1. Conectar el micro:bit al ordenador
2. Emparejar el **micro:bit** (**Connect device**)
3. Descargar el código al **micro:bit**

Programa completo

Esta es la imagen del programa completo

No se puede leer casi

Puedes encontrar los **bloques de programa** en las **hojas** con la señal



CODIGO COMPLETO

El programa completo se puede ver en los enlaces:

Nombres de variables en castellano

https://makecode.microbit.org/S86240-75813-88279-48918

Nombres de variables en catalán

https://makecode.microbit.org/_aEecTbi2fRsV

o bien usando los códigos QR

Agraïments i Contribucions i Llicència

Han contribuït a la creació d'aquest material

- Joaquin Jimenez Godoy
- Tony Barbosa
- Wouter Molevelt
- Maria Teresa Miras
- Eusebi Calonge

https://steam4all.eu