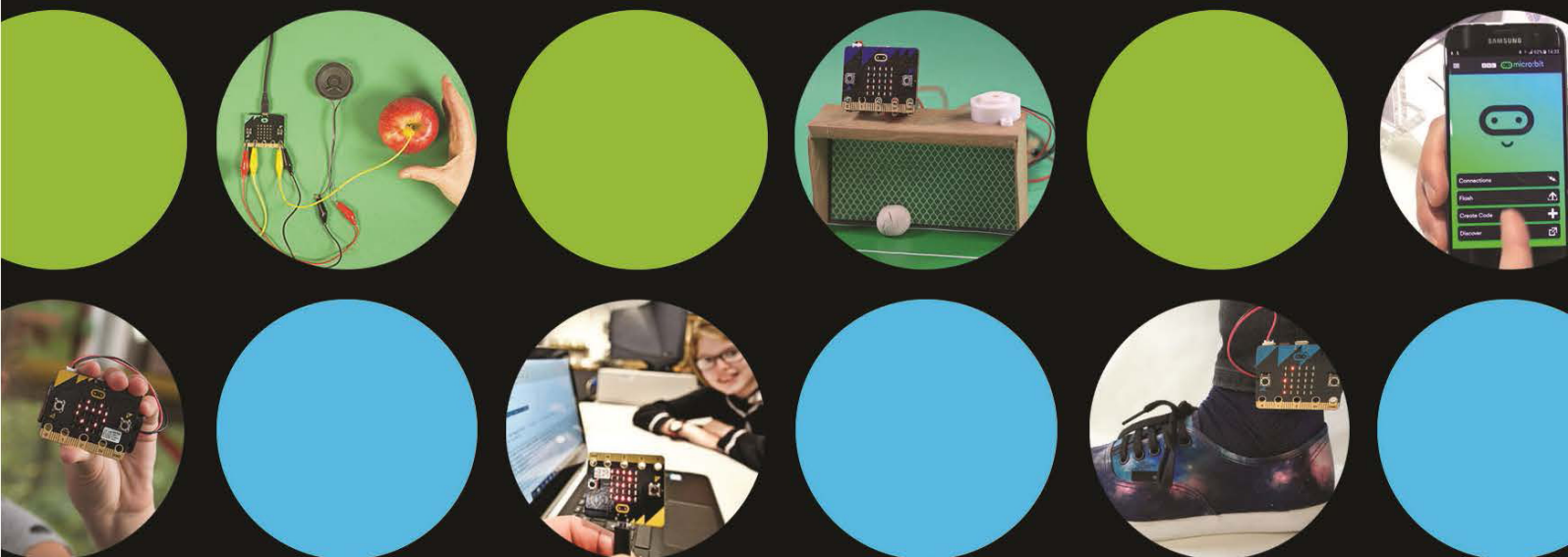




Manual de Programación micro:bit



Manual de Programación micro:bit

Título: Manual de Programación micro:bit

Autor: José Francisco Muñoz (<https://almeribot.com/>)

Editor: Microes.org

1ª Edición: Xullo, 2018 - 2ª Edición: Xaneiro, 2019

Este manual é un resumo online doutras guías de manexo impresas:

Manual micro:bit & Basic kit

Manual de Programación micro:bit & Octopusbit



Estas guías poden adquirirse ao comprar os packs de ampliación dispoñibles:

<http://microes.org/formacion/packs-educativos-microbit>

“Manual de Programación micro:bit” foi publicado baixo a licencia Creative Commons:



micro:es

Comunidade micro: bit en España

Dúvidas? ¿Queres colaborar coa comunidade?. Contáctanos:

e-mail: info@microes.org
teléfono: 945 29 80 50 (de 8:00 a 14:00)

Podes mercar a tarxeta micro:bit e infindade de complementos compatibles en:

robotopia

<http://robotopia.es/>

Índice de contidos

INTRODUCCIÓN Á MICRO:BIT

PROGRAMANDO A MICRO:BIT

Panel LED

Reto 1. Ola Mundo

Reto 2. Animando os iconos

Reto 3. Deseñando iconos

Sensor de temperatura

Reto 4. Termómetro dixital

Reto 5. Aviso de placas de xeo

Reto 6. Temperatura óptima dunha neveira

Sensor de luminosidade

Reto 7. Amosar o valor do sensor de luz

Reto 8. Alarma de exposición ao sol

Premedores

Reto 9. Que botón premeches?

Reto 10. Medir a intensidade da luz ambiente

Reto 11. Modificar o brillo dos LEDs

Acelerómetro

Reto 12. Amosar o valor do acelerómetro

Reto 13. Número ao chou entre 0 e 100

Brújula

Reto 14. Mostrar el valor de la brújula

Reto 15. Aviso sonoro de orientación Norte

Radio

Reto 16. Que botón se premeu? Versión inalámbrica

Reto 17. Chivato a distancia

Pins de entrada e saída

Reto 18. Hai continuidade?

Reto 19. Movendo un servomotor

~ Introducción á micro:BIT ~

1.- Que é a micro:BIT?

A micro:BIT é unha pequena tarxeta programable, cun custo asequible a calquera peto. Aínda cando o seu tamaño é moi reducido, incorpora gran cantidade de sensores e actuadores o que unido a que usa un software Open Source, fan da micro:BIT unha plataforma ideal para introducirse no mundo da programación de robots.

2.- Como se programa a micro:BIT?

Hai varias plataformas que permiten codificar a micro:BIT, entre elas destaca MakeCode, tanto na súa versión online como offline.

A versión online é accesible dende este enlace:

<https://makecode.microbit.org/>

A versión offline pódese descargar dende este enlace: <http://kittenbot.cc/bbs/topic/3/microbit-makecode-offline-version>

Ademais a micro:Bit pódese programar con JavaScript, Python, Scratch (engadindo unha extensión) e Tickle (aplicación para iPad).

3.- Características

A micro:BIT incorpora:

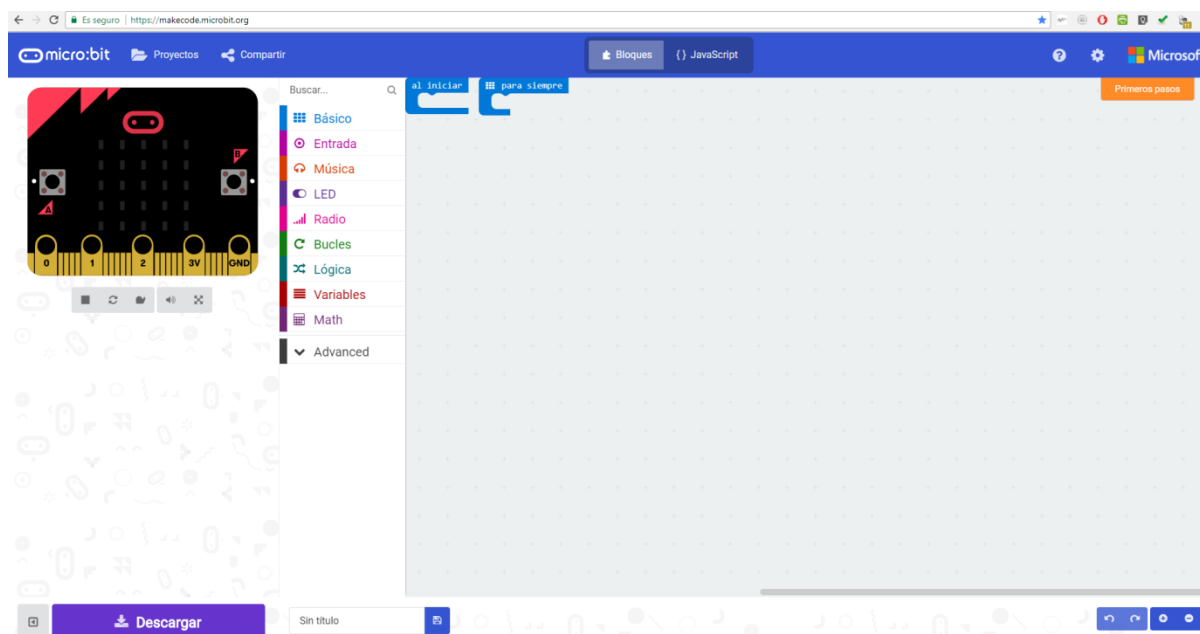
- 25 LEDs. Pódense programar de forma independente e permiten amosar números, letras e imaxes. Se o texto ou a cifra non caben no display desprázanse de forma automática.
- Sensor de Luz. Os LEDs tamén teñen a posibilidade de ser usados como sensor de luz ambiente.
- Premedores. Existen 2 botóns, etiquetados como A e B. Pódese detectar a pulsación independente de cada un deles así como a pulsación simultánea de ámbolos dous.
- Conectores. Situados na parte inferior da placa, dispón de 25 conexións que permiten conectar outros sensores e actuadores. 5 das conexións (0,1,2 3v. y GND) atópanse sobredimensionadas, para facilitar a conexión mediante pinzas de crocodilo.
- Sensor de temperatura. Permite coñecer á micro:BIT a temperatura ambiente. As unidades son os graos Celsius.

- Acelerómetro. Activada cando se move a placa, permite coñecer aceleracións e xiros aos que se somete a placa.
- Brújula dixital. Permite coñecer a desviación respecto o Norte Magnético. Tamén permite detectar a presenza de campos magnéticos próximos. Ao iniciar o seu uso entra en modo de calibración.
- Radio. Permite conectarse inalámbricamente con outras micro:BITs.
- Bluetooth. Ideal para conectarse e intercambiar datos inalámbricamente con outros dispositivos (móviles, tablets, ordenadores, etc) que dispoñan deste tipo de conexión.
- USB. Usado para descargar os programas á memoria da tarxeta e para alimentar eléctricamente a micro:BIT.
- Conector de batería. Permite subministrar electricidade mediante dúas pilas AAA ou unha batería. A tarxeta carece de interruptor, polo que cando se conecta a fonte de alimentación execútase de forma automática o código que haxa en memoria.

4.- MakeCode. Primeiros pasos

Como se veu anteriormente MakeCode pódese executar online, se está dispoñible conexión a Internet ou offline.

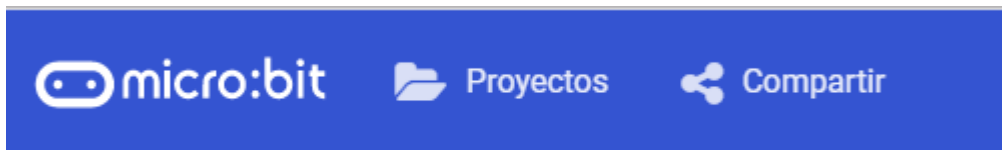
Para acceder directamente á versión online de MakeCode para micro:BIT débese introducir no navegador a seguinte dirección: <https://makecode.microbit.org/>



A) Descripción Xeral

1. **Simulador.** Situado no lado superior esquerdo da pantalla, amósanos unha micro:BIT, onde se pode ver a execución do programa. Esta sección é útil para depurar o código antes de volcalo á memoria da micro:BIT.

2. **Caixa de Ferramentas.** Situado nunha columna vertical, ao lado do simulador, alí ubícanse os bloques de programación organizados por categorías. Ao premer sobre o nome dunha categoría ábrese, á dereita, un despregable onde aparecen os bloques máis usados da sección. É importante fixarse en que xusto debaixo do nome da categoría, a maioría de veces, aparece a opción **... Máis**, se se preme amósanse o resto de bloques da categoría. Unha vez seleccionado o bloque a usar débse arrastrar ao área de programación.
3. **Área de programación.** Situado á dereita da Caixa de Ferramentas, é a zona onde se arrastran os bloques para crear o programa.
4. **Barras de Ferramentas.** Situadas na parte superior e inferior da pantalla, ofrece atallos ás diferentes funcionalidades. Na parte superior:

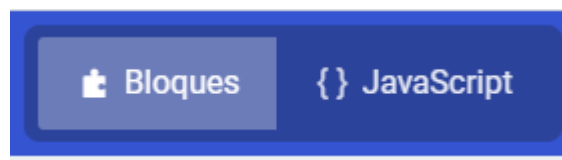


4.1 Ao premer sobre **micro:bit** leva á páxina <https://microbit.org/code/> onde atopar exemplos de codificación en diferentes linguaxes de programación.

4.2 A sección **Proyectos** permite elixir entre:

- a) As miñas Cousas. Aquí pódese crear un novo proxecto ou importar un arquivo previamente gardado, ademais de aparecer un listado dos programas máis recentes, ordenados por data.
- b) Proxectos. Nesta sección atópanse diferentes tutoriais, paso a paso, onde aprender a codificar a micro:BIT usando MakeCode.
- c) Exemplos. Ofrécense varios códigos de exemplo.

4.3 **Compartir.** Permite ao usuario compartir os códigos creados. En primeiro lugar solicítase “Publicar o Proxecto”, tras o que aparece unha nova ventá, que ofrece un enlace e diferentes opcións de código para poder inserir nunha páxina web.



4.4 Permite intercambiar entre a visualización mediante bloques ou ver o código escrito en JavaScript.



4.5 No extremo dereito da Barra de Ferramentas atópanse as seccións de Axuda, Configuración, acceso á páxina principal de MakeCode e en laranxa a sección primeiros pasos, que ofrece un tutorial de uso.

Na barra inferior aparecen os seguintes atallos:

4.6 O primeiro icono da esquerda, permite ocultar o simulador, o que permite ter máis espazo dispoñible no Área de Programación.

4.7 Descargar. Descarga o código á micro:BIT. Usarase un nome por defecto se non se asignou un previamente.

4.8 Premendo sobre o icono do disquete, permite asignar un nome e descargar o programa ao disco duro.



4.9 As frechas permiten desfacer e refacer os pasos de codificación.

4.10 No extremo dereito inferior aparece o icono “+” que aumenta o zoom do área de programación e o icono “-” que diminúe o zoom.

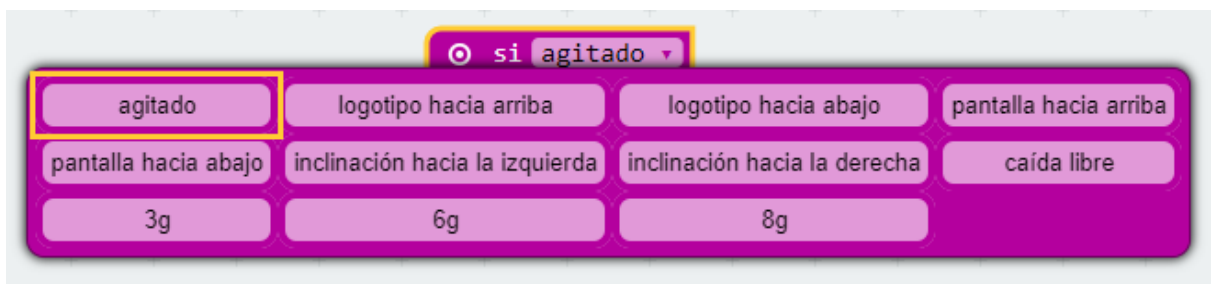


B) Como crear un programa con MakeCode

O primeiro que se necesita é decidir que evento é o que inicia el código.

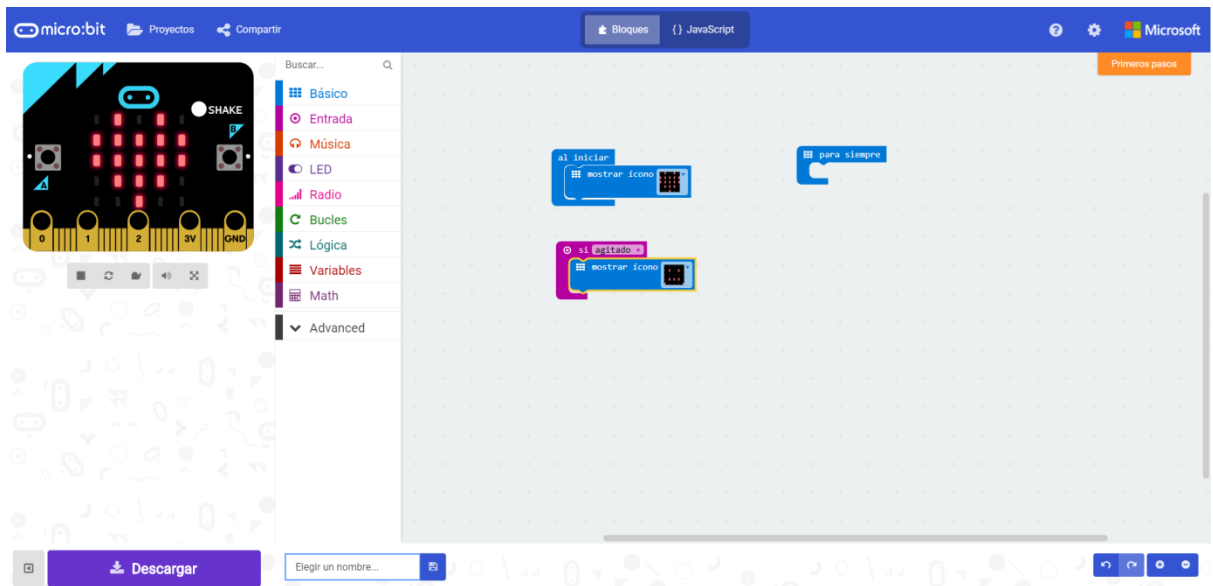
Por defecto, MakeCode, sitúa na área de programación dos eventos cos que iniciar o código, estes son: **ao iniciar** e **para sempre**.

Tamén existen outros eventos que permiten iniciar o programa, situados na categoría **Entrada**, son: **ao premer o botón**, **ao premer pin** ou **se axitado** ademais permiten diferentes opcións.



Para crear o programa, deberase ir arrastrando os diferentes bloques ao área de programación.

Poderase comprobar o funcionamento do programa no simulador situado á esquerda. Ademais segundo os bloques introducidos, aparecerán diferentes opcións que permiten interactuar co simulador.



C) Cómo descargar un código á micro:BIT

Tras conectar a micro:BIT ao ordenador, aparece esta como unha nova unidade. Terminado o programa e comprobado o seu correcto funcionamento, débese premer sobre o icono de descarga, o que copia o código ao ordenador cun nome por defecto. Outra opción é introducir o nome elixido na caixa situada ao lado do icono do disco e premer sobre este para descargar. Xa só queda copiar o ficheiro



con extensión .hex á unidade que apareceu tras conectar a micro:BIT aloordenador.

D) Cómo executar o código

Copiado o código, o programa comeza a executarse de forma automática. Se se quere usar a micro:BIT desconectada do ordenador, débese conectar un pack de pilas ou batería ao conector situado xunto ao conector microUSB e inmediatamente executarse o código descargado.

~ Programando a micro:BIT ~

Neste apartado vanse amosar algúns exemplos de como programar os diferentes sensores e actuadores que compoñen a micro:BIT.

Tal como se explicou no capítulo anterior, os códigos creados poderanse comprobar inicialmente no simulador da páxina de MakeCode. Posteriormente o código pódese descargar á memoria da micro:BIT e executar de forma autónoma.

Panel LED

A tarxeta consta de 25 LEDs de cor vermella, estas pódense programar de forma independente. Permite amosar números, letras e imaxes. O sistema está preparado para desplazarse de forma automática no suposto de que non caiban os caracteres a amosar.

Reto 1. Ola Mundo

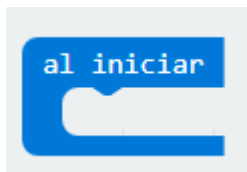
Cando se aprende a programar, chámase “Ola Mundo” ao primeiro programa que se crea. Este código imprime o texto «Ola Mundo!» nun dispositivo de visualización.

Obxectivo.

Ao conectar a micro:BIT, débese amosar o texto «Ola Mundo!!!» no panel LED da tarxeta.

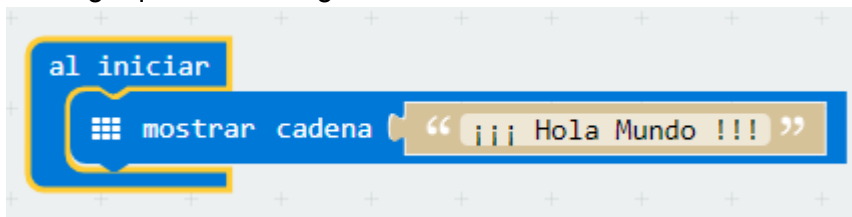
Descrición do código.

Poponse usar o evento **ao iniciar**, situado na categoría **Básico**. Este bloque permite executar o código introducido, cando se inicia a micro:BIT



A continuación usarase o bloque **amosar cadea**, tamén localizado na categoría **Básico** e introducir o texto “Ola Mundo !!!”. Este bloque permite amosar carácter a carácter, o texto introducido.

O código quedará da seguinte maneira.



De forma inmediata comezará a lerse no panel LED a cadea de texto inserida. O texto non cabe na súa totalidade polo que este desprazarase de forma automática.

Proposta.

Modificar o código, para que se amose o nome do programador.

Reto 2. Animando os iconos

A animación é unha técnica que logra crear sensación de movemento a imaxes estáticas. Conséguese mediante una secuencia de imaxes ordenadas, que ao ser amosadas consecutivamente, conseguen xerar a ilusión visual de movemento.

Obxectivo.

Proponse, que ao conectar a micro:BIT, amósese no panel LED un corazón que late.

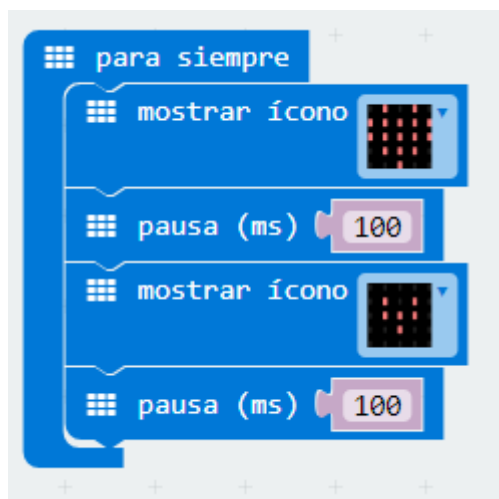
Descrición do código.

Usarase o evento “para sempre”, situado na categoría **Básico**.

Situar no interior do bloque anterior a instrucción **amosar ícono**, usar o chamado corazón.

Engadir o bloque **pausa ms (100)** localizado na categoría **Básico**.

Repetir os dous bloques anteriores, pero seleccionando o ícono “corazón pequeno”.



De forma inmediata visualízase no simulador un corazón que semella latir.

Proposta.

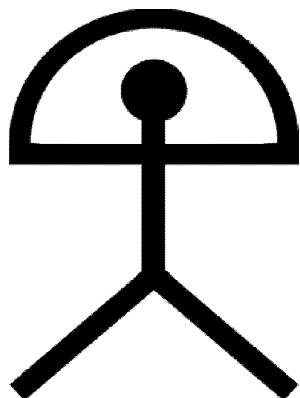
Facer un código que amose a animación dunhas tesoiras cortando.

Reto 3. Deseñando iconos

MakeCode permite deseñar de forma gráfica os iconos a amosar no display LED. Para este mester dispón dun bloque específico situado na categoría **Básico**. O bloque chámase “amosar LEDs” e permite ao usuario deseñar os seus propios iconos marcando de forma independente os LEDs que se queren iluminar.

OBXECTIVO

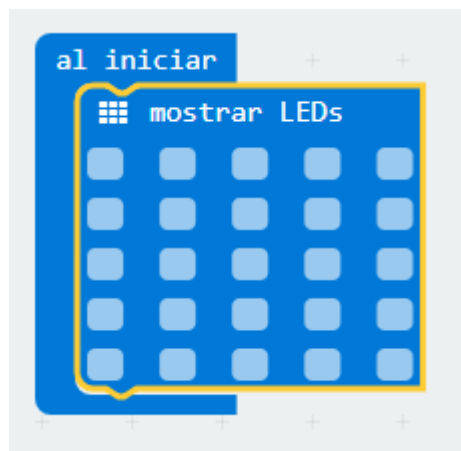
Deseñar un icono con forma de Indalo. Trátase dunha figura rupestre atopada nunha cova da provincia de Almería e que representa unha figura humana cos brazos estendidos e un arco sobre as súas mans. Actualmente considérase un símbolo da provincia de Almería



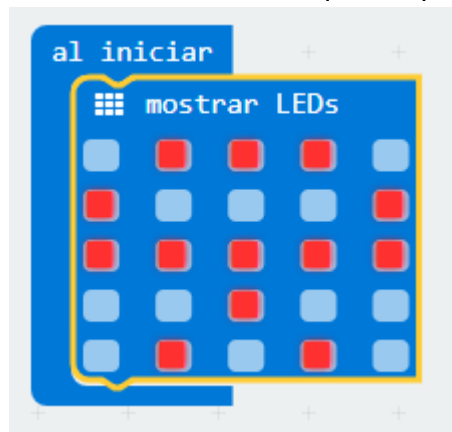
Descrición do código.

Usarase o evento **ao iniciar** situado na categoría **Básico**.

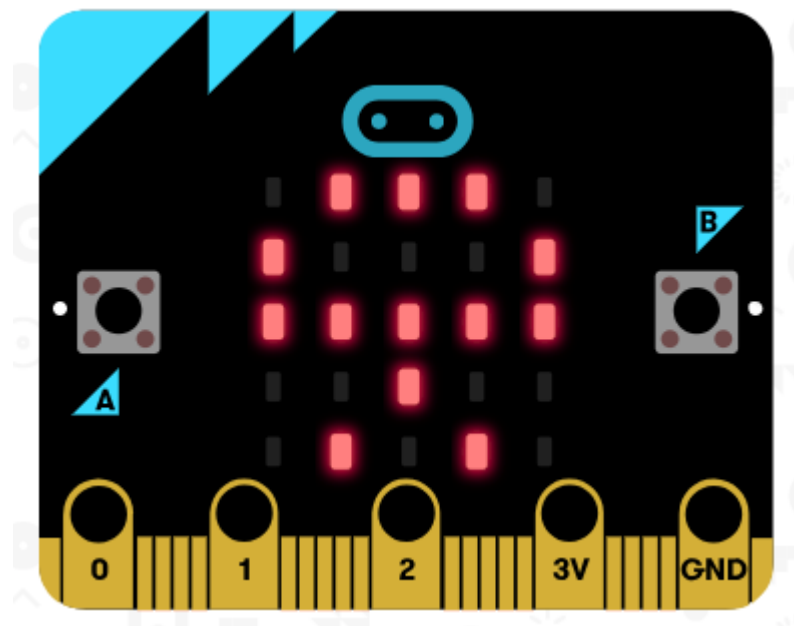
Situar no interior o bloque anterior, a instrución “amosar LEDs”, este localízase na categoría **Básico**. Este bloque permite marcar de forma independente os LEDs que se queren prender.



Premer sobre os LEDs que se queren iluminar ata conseguir o icono desexado.



Deberá aparecer no simulador o icono coa forma deseñada.



Proposta.

Deseñar un código que amose un logotipo deseñado polo programador.

Sensor de temperatura

O sensor de temperatura integrado na placa detecta a temperatura ambiente en graos Celsius.

Reto 4. Termómetro dixital

Un termómetro é un instrumento que serve para medir a temperatura; o máis habitual consiste nun tubo capilar de vidro fechado e terminado nun pequeno depósito que contén unha certa cantidade de mercurio ou alcohol, o cal dilátase ao aumentar a temperatura ou contráese ao diminuír e cuxas variacións de volume se len nunha escala graduada.

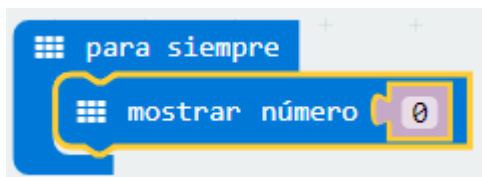
Os termómetros dixitais son aqueles que, valéndose de dispositivos [transdutores](#), utilizan circuitos electrónicos para [converter](#) en números as pequenas variacións de tensión obtidas, amosando finalmente a temperatura nun [visualizador](#).

Obxectivo.

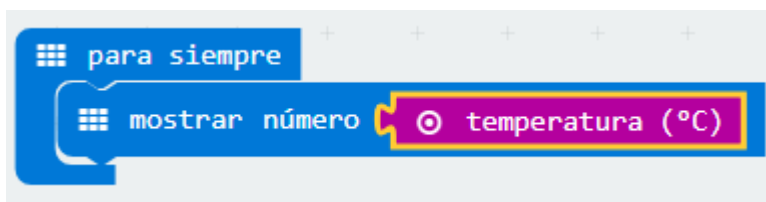
Amosar no panel LED a temperatura detectada polo sensor de temperatura integrado na micro:BIT.

Descrición do código.

Propónse iniciar o programa usando o evento “para sempre”. Engadir o bloque, localizado na categoría [Básico](#), [amosar número](#). Este bloque amosa na pantalla LED o número introducido, desprazándose se é maior de 1 cifra.



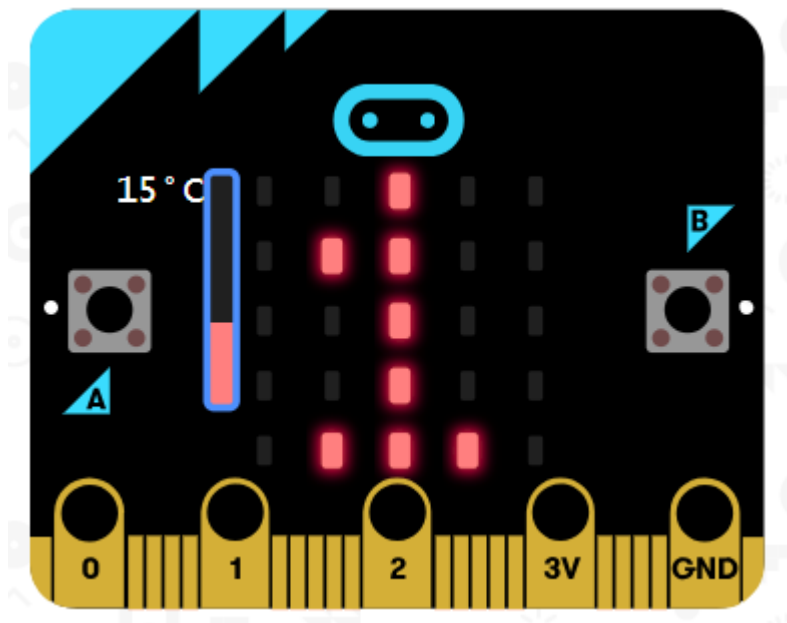
Substituír o número “0” polo valor obtido do sensor de temperatura. Este bloque localízase na categoría [Entrada](#).



A continuación bórrase a pantalla e insírese unha pausa de 1000 ms. Quedando o programa da seguinte forma:



No! simulador aparecerá una barra vertical que permite modificar a temperatura, sendo a temperatura marcada a amosada no panel LED.



Proposta.

Facer un programa que ao premer o botón A amosa a temperatura en graos celsius e ao premer o botón B a temperatura amosada sexa en graos Fahrenheit.

Reto 5. Aviso de placas de xeo

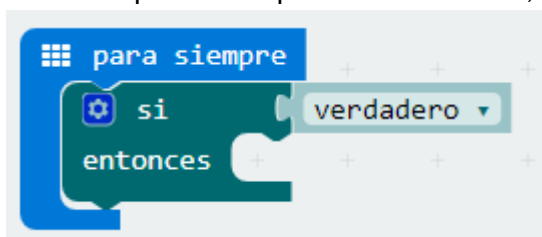
A maioría de coches incorporan no salpicadeiro, xunto á pantalla que marca a temperatura exterior, unha testemuña con forma de folerpa de nieve, que avisa da posibilidade de que haxa placas de xeo na estrada. Esta testemuña sóese iluminar cando a temperatura baixa de 3 graos centígrados.

Obxectivo.

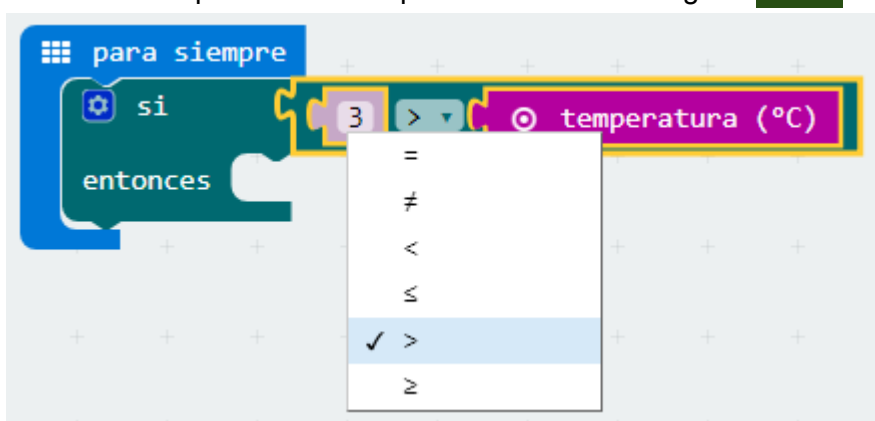
Amosar no panel LED da micor:BIT un icono con forma de folerpa de neve cando a temperatura estea por debaixo de 3 graos.

Descrición do código.

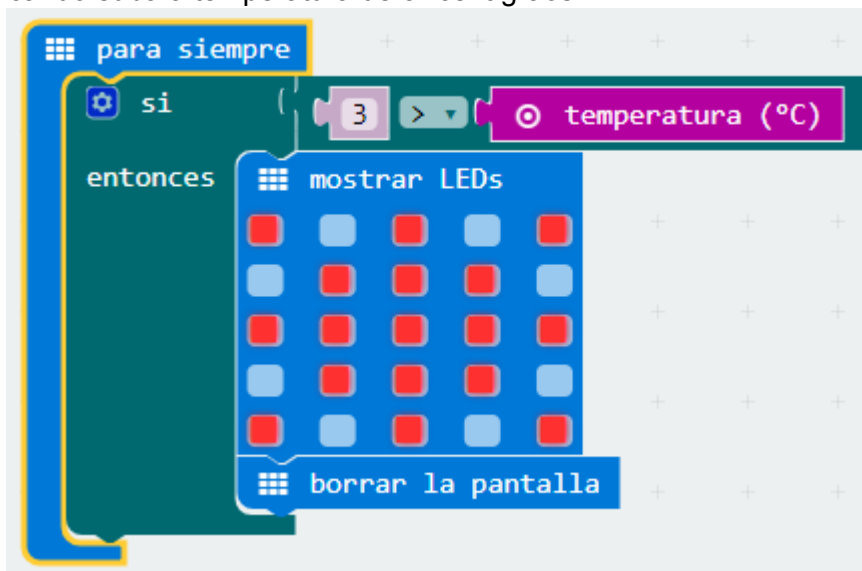
Para iniciar o programa, propónse usar o evento **para sempre**. Posteriormente débese engadir o operador lóxico **se ... entón** situado na sección **Lóxica**. Este bloque verifica se é verdade que se cumpre unha condición, se é así, executa as instrucións introducidas.



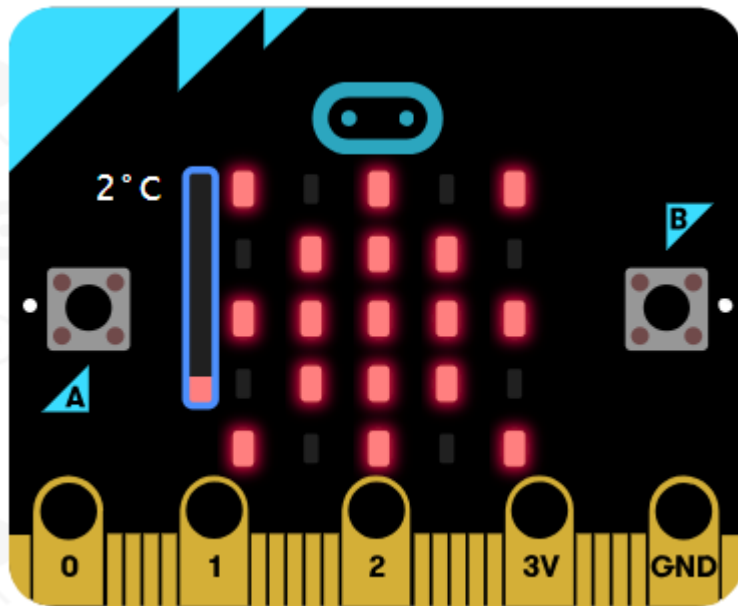
Para comprobar se a temperatura está por debaixo dos 3º centígraos usarase o operador que devolve verdadeiro sempre que 3 sexa maior que o valor reportado polo sensor de temperatura. O bloque localízase na categoría **Lóxica**.



Para amosar un icono con forma de folepa de neve usarase o bloque **amosar LED**, tamén se debe engadir o bloque **borrar a pantalla** para que non se quede prendido o panel LED cando suba a temperatura de 3º centígraos.



Unha vez terminado o código, aparecerá o símbolo de xeo, cando a temperatura sexa menor de 3º centígraos.



Proposta.

Engadir un aviso sonoro cando a temperatura sexa inferior a 3 graos celsius.

Reto 6. Temperatura óptima dunha neveira

A temperatura óptima dun frigorífico é de 7°C, mentras que a temperatura dun conxelador debe estar entono aos -18°C. Algunhas neveras incorporan un avisador acústico que se activa cando a temperatura non é a óptima.

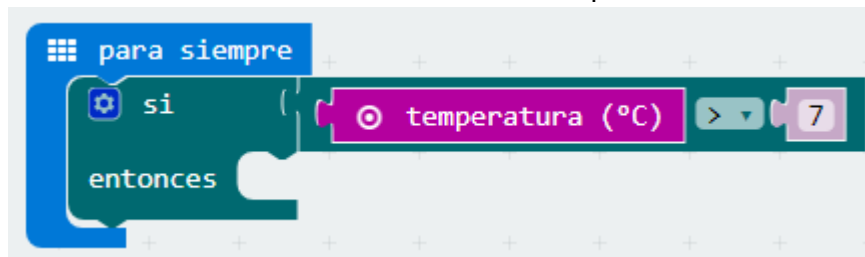
Obxectivo.

Amosar no panel LED da micro:BIT a temperatura do frigorífico e activar unha alarma cando a temperatura estea por riba de 7°C

Descrición do código.

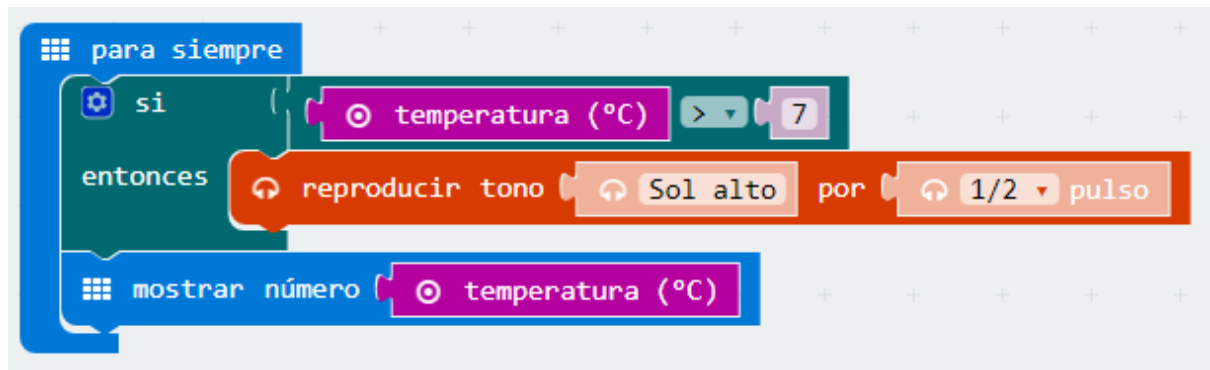
Neste reto incorpórase un altofalante como actuador externo. Conectarase un dos polos ao GND e o outro ao PIN 0.

Para iniciar o programa usárase o bloque “para sempre”. Dentro situarase a condición “se entón”. A condición a verificar será: se a temperatura é maior de 7 graos.



Se o valor é verdadeiro entón reproducirase un ton de aviso.

Tamén se amosa a temperatura medida.



No simulador pódese variar a temperatura para comprobar o correcto funcionamento do programa.

Proposta.

Modificar o código para monitorizar o conxelador.

Sensor de luminosidad

Os LEDs da placa micro:bit tamén poden actuar como entrada facendo que deteten a luz ambiente.

Reto 7. Amosar o valor do sensor de luz

Moitas son as aplicacións que se lle poden dar a un sensor de luz, a máis coñecida son as relacionadas coa fotografía, pero tamén se soen usar en instalacións como por exemplo prender as luces da rúa, de forma automática, cando a intensidade luminosa descende por debaixo dun valor fixado.

Obxectivo.

Amosar o valor do sensor de luz incorporado en la micro:bit.

Descrición do código.

Ao iniciarse a micro:BIT amosarase no panel LED o valor numérico lido polo sensor de luz. Usarase o evento **para sempre**, onde se introduce o bloque **amosar número** e usarase o bloque **nivel de luz** queamosa o valor medido polo sensor de luz.



Proposta.

Programar a micro:bit para que prenda tódolos LEDs do panel, cando o nivel de luz diminúe de 100 unidades.

Reto 8. Alarma de exposición ao sol

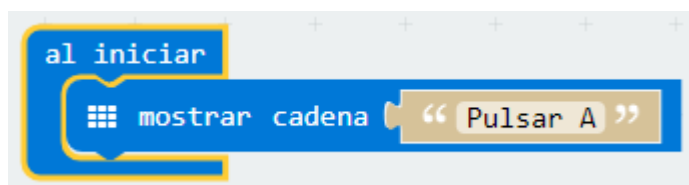
Na actualidade hai estudos que amosan a relación entre a exposición ao sol e o risco de padecer cáncer de pel. Os dermatólogos recomendan evitar a exposición ao sol durante as horas de máxima radiación e o uso de cremas protectoras. Proponse crear, usando a micro:BIT, un dispositivo que avise que estamos expostos a un valor de elevado de radiación solar.

Obxectivo.

Ao iniciar o programa indícase que hai que premer o botón A para que amose o valor da intensidade do sol. Ao premer o botón amosarase unha caraña feliz se o valor é inferior a 175^(*) e unha caraña triste se é superior. Pasados 3 segundos pasarase a modo de aforro de enerxía, quedando á espera de que se volva premer o botón A. Nota.- O valor usado como límite, carece de toda validez científica.

Descrición do código.

Ao iniciarse a micro:BIT amosarase o texto “Premar A”



Usarase o evento **ao premerse o botón A**, para iniciar a medición do valor da exposición. Introducirase a condición **se ... entón, se non**, para comprobar se se está por riba ou por debaixo do valor teórico tomado como límite. Se o valor do nivel de luz é menor a 175, amósase unha caraña feliz e aos 3 segundos apágase a pantalla. En caso contrario amósase unha caraña triste e entrarase en modo de aforro de enerxía pasados 3 segundos. O programa quedaría da seguinte forma.



Proposta.

Modificar o código, para que ademais de amosar unha “cariña triste” se faga sonar un aviso acústico.

Premedores

Hai dous botóns na cara frontal de micro:bit (etiquetados como A e B). Podes detectar cando son premidos de forma independente ou ás veces executar unha acción en cada caso.

Reto 9. Que botón premeches?

Ás veces pode interesar usar un premedor para iniciar, pausar ou deter un código ou subrutina. Algúns robots incorporan botoneiras para esta finalidade.

Obxectivo.

Crear un programa que amose en pantalla a letra do premedor que seaccionou.

Descrición do código.

Usarase bloque **ao iniciar** para que ao principio se amose o texto “Preme un botón”. A micro:BIT non amosa caracteres acentuados, polo que deixará un oco se se pon a tilde na palabra botón.

Posteriormente usarase o evento **ao premer o botón A** e introducirase o bloque **amosar cadea** onde se substituirá o texto por defecto pola letra “A”.

Isto mesmo repetirase cando se preme o botón B e a pulsación combinada de A+B.

O código quedaría da seguinte forma:



Proposta.

Crear unha caixa de música. Ao premer o botón A fará sonar unha melodía. O bloque necesario para facer sonar unha melodía atópase na categoría **Música**.

Reto 10. Medir a intensidade da luz ambiente

Os sensores de luz úsanse para detectar o nivel de luminosidade e producir un sinal de saída representativa da cantidade de luz detectada. Nalgunhas ocasións pode ser interesante coñecer o nivel lumínico que hai en diferentes situacións e así poder calibrar un dispositivo que dependa do valor da intensidade luminosa, como pode ser unha cámara fotográfica.

MakeCode dispón do **bloque nivel de luz** na categoría **Entrada**, que le o valor de luz aplicado na pantalla LED. Este valor móvese nun rango de 0 (escuro) a 255 (luminoso).

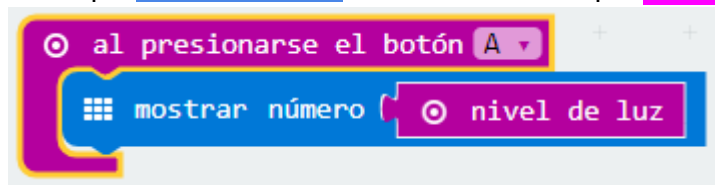
Obxectivo.

Ao premer sobre o pulsador A amosarase o valor numérico da intensidade da luz ambiente. Ao premer sobre B, amosarase graficamente a intensidade lumínica. Ao premer A e B de forma simultánea, apagarase a pantalla.

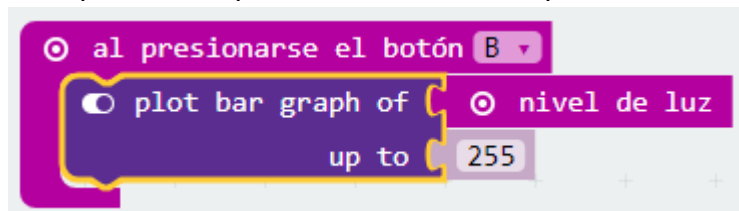
Descrición do código.

Proponse crear tres programas diferentes e que cada un se inicie segundo o botón ou combinación de botóns premidos.

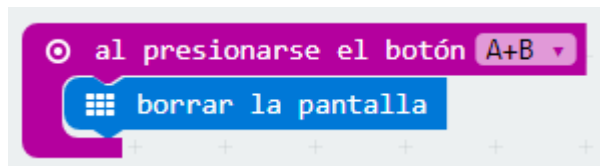
Ao premer sobre o botón A amosarase o valor numérico da intensidade luminosa. Usarase o bloque **amosar número** introducindo o bloque **nivel de luz** como valor a presentar.



Ao premer o botón B prenderanse os LEDs necesarios, para amosar de forma gráfica o valor do nivel de luz. Usarase o bloque **plot bar graph of ... up to ..** localizado na sección **LED**. En primeiro lugar insírese o bloque **nivel de luz** e no segundo termo substitúese o valor por defecto por 255, valor máximo que detecta o sensor de luz.



Para rematar usarase a pulsación combinada dos botóns A e B para borrar e apagar a pantalla LED.



Proposta.

Crear un código que permita prender todos os LEDs do panel da micro:BIT ao premer o botón A e apagalos ao pulsar o botón B, pero non se prenderá se o valor do sensor de luz é superior a 175.

Reto 11. Modificar o brillo dos LEDs

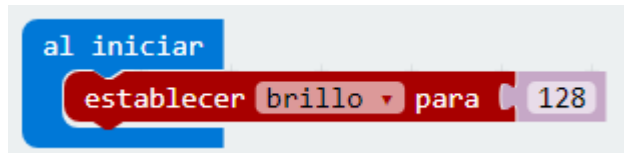
A maioría dos teléfonos móbiles incorporan a posibilidade de modificar a luminosidade da pantalla, para adaptarse á luz ambiente.

Obxectivo.

Deseñar un programa que permita modificar a luminosidade dos LEDs, se se preme o botón A a luminosidade aumentará, se se acciona o botón B a luminosidade diminuirá. Se se premen A+B a luminosidade situarase a un valor intermedio (128).

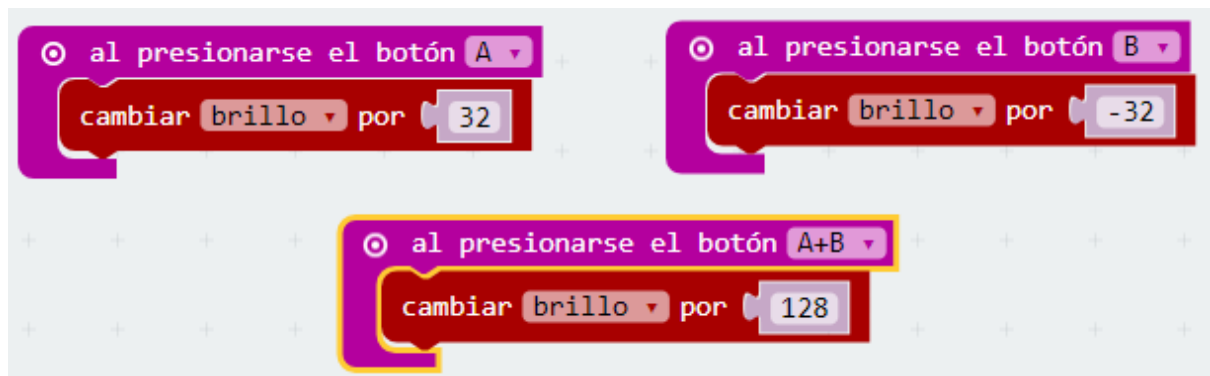
Descrición do código.

Para iniciar o programa definirase unha variable que se chamará “brillo”. Na categoría **Variables** poderase crear unha nova variable ou renomear a que sae por defecto. Establecerase un valor inicial de 128 para a variable creada.

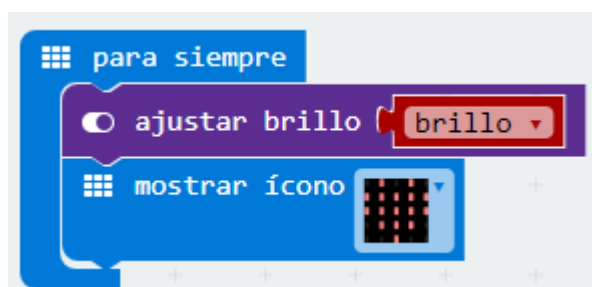


Proponse usar pasos por valor de 32 unidades para evidenciar o cambio de brillo nos LEDs, pero pódese usar calquera outro valor.

Ao accionar o premador A aumentarase a variable “brillo” en 32 pasos. Ao premer o botón B diminúese o valor da variable “brillo” en 32 pasos. Por último se se preme á vez as teclas A e B establece o valor inicial da variable “brillo”.



Xa para rematar faltaría amosar no panel LED un icono que amose os cambios no brillo dos LEDs. Débese usar a variable “brillo” como valor que toma o bloque **axustar brillo**.



Proposta.

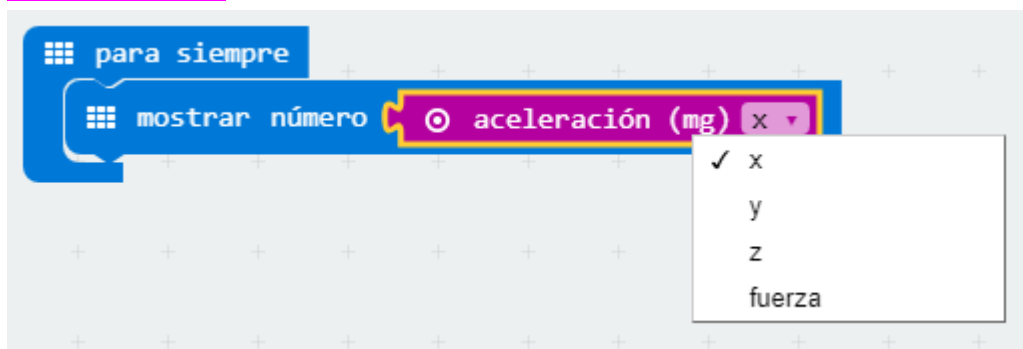
Proponse como reto facer un código que axuste o brillo de forma automática, a maior luminosidade medida polo sensor de luz, maior valor do brillo dos LEDs.

Acelerómetro

O acelerómetro mide a aceleración do teu micro:bit. Actívase cando a tñ/a placa se move e tamén pode detectar outras accións como axitar, xirar e ata soltar o teu micro:bit en caída libre!

Reto 12. Amosar o valor do acelerómetro

O acelerómetro incluído na micro:bit permite coñecer as aceleracións segundo os eixes X, Y, Z, ademais da forza da aceleración. Para coñecer estes valores Makecode incorpora o bloque **aceleración (mg)** permitindo seleccionar os diferentes valores.



Obxectivo.

Amosar o valor do acelerómetro incorporado na micro:bit, segundo o eixe X.

Descrición do código.

Ao iniciarse a micro:BIT amosarase no panel LED o valor numérico lido polo acelerómetro. Usarase o evento **para sempre**, onde se introduce o bloque **amosar número** e usarase o bloque **aceleración (mg)** que amosa o valor medido polo sensor de luz. Débese seleccionar o “x” do desplegable.



Proposta.

Programar a micro:bit para que amose o valor da forza da aceleración que se lle aplica.

Reto 13. Número ao chou entre 0 e 100

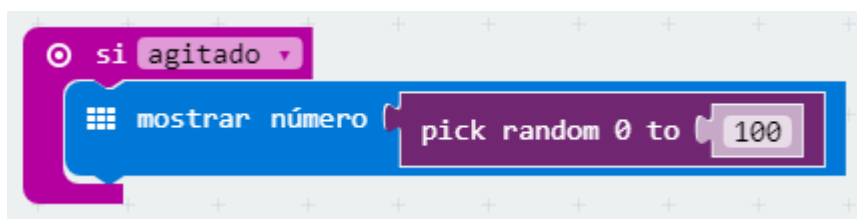
Entre os eventos dispoñibles para iniciar a micro:bit dispónse de eventos que usan o acelerómetro incorporado para iniciar un programa.

Obxectivo.

Cando se axite a micro:bit, amosar un número ao chou entre 0 e 100.

Descrición do código.

Ao iniciarse a micro:BIT queda á espera de ser axitada. Unha vez se axita amosarase un número ao chou entre 0 e 100. Usarase o bloque **amosar número**, introducindo como valor numérico o obtido de usar o bloque **pick random 0 to 100**



Proposta.

Cando a micro:bit teña o panel LED cara arriba amosará unha cariña feliz, cando estea orientado cara abaixo amosará unha cariña triste.

Brújula

A brújula detecta o campo magnético terrestre polo que se pode saber en que dirección está orientada a micro:bit. (Necesita ser calibrada para asegurar un resultado preciso.)

Reto 14. Amosar o valor da brújula

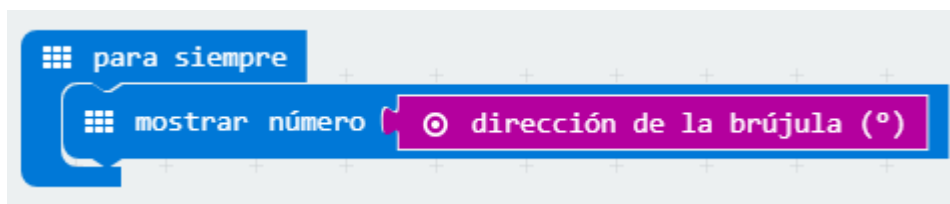
Hoxe en día moitos dispositivos electrónicos como móbiles, GPS, etc. incorporan unha brújula entre os seus sensores.

Obxectivo.

Amosar o valor da brújula incorporada na micro:bit.

Descrición do código.

Ao iniciarse a micro:BIT amosarase no panel LED o valor numérico lido na brújula que incorpora a micro:bit. Usarase o evento **para sempre**, onde se introduce o bloque **amosar número** e usarase o bloque **dirección da brújula (°)** que amosa o valor medido pola brújula.



Proposta.

Programar a micro:bit para que cando se oriente cara algún dos puntos cardinais en vez de amosar o valor numérico, amose a súa inicial.

Reto 15. Aviso sonoro de orientación Norte

Cando se necesita orientarse cun mapa, o que primeiro que hai que facer, é situar o mapa con orientación norte. Para eles débese usar unha brújula.

Obxectivo.

Crear un programa que mediante un aviso acústico se advirta de que a micro:BIT está orientada ao Norte.

Descrición do código.

Usar o evento **para sempre**, para iniciar o programa.

Crear unha variable onde se almacenará o valor do sensor obtido polo bloque **dirección da brújula (°)**.

Introdúcese o condicional **se entón se non** para comprobar en qué intervalo se atopa o valor obtido polo sensor.

Se o valor se atopa entre 315° e 45° a micro:BIT atópase orientada ao Norte, polo que se amosará unha N e reproducirase unha nota. Se non se cumpre a hipótese, bórrase a pantalla.



Proposta.

Facer un código que sitúe de forma máis precisa o Norte Magnético.

Radio

A radio permíteche comunicar a túa micro:bit con outras micro:bit. Por exemplo, podes conectar todas as tarxetas dentro dun aula a unha mesma emisora, usala para enviar mensaxes entre elas e moito máis!.

Reto 16. Que botón se premeu? Versión inalámbrica

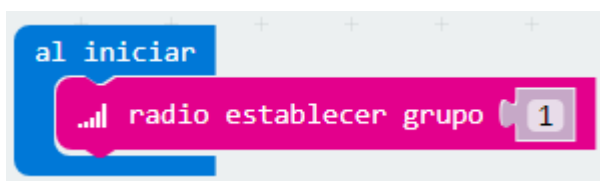
Os mandos a distancia usados habitualmente utilizan principalmente dúas tecnoloxías, infravermellos ou radiofrecuencia. Os mandos por infravermellos emiten pulsos de luz infravermella e os de radiofrecuencia ondas de radio. No caso da micro:BIT a tecnoloxía usada é a de radiofrecuencia.

Obxectivo.

Crear un código que permita visualizar no panel LED da micro:BIT receptora, a letra do botón que se remeu na tarxeta emisora.

Descrición do código.

Deberanse crear dous códigos, un para a tarxeta emisora e outro para a receptora. Para que as tarxetas se poidan comunicar entre sí, ambas deben estar no mesmo grupo de comunicación. Por este motivo, deberase establecer o grupo de comunicación mediante o bloque **radio establecer grupo** situado na categoría **Radio**. Recoméndase usar o evento **ao iniciar** para establecer o grupo de comunicación no código de ambas tarxetas.

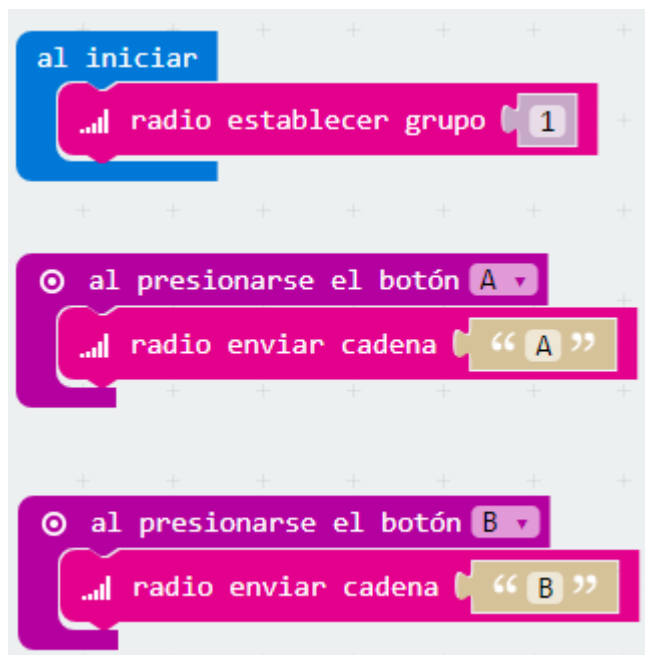


micro:BIT emisora:

Tras establecer o grupo de comunicación, enviarase a letra do botón premido.

Proponse usar o evento **al premer o botón A**, situado na categoría **Entrada** para enviar a cadea "A". Para realizar o envío da cadea usarase o bloque **radio enviar cadea**. En segundo lugar, copiarase o código e modificarase de forma que **ao premer o botón B** a cadea enviada da letra B.

O código da tarxeta emisora debería quedar da seguinte forma:

**micro:BIT receptora:**

A cadena enviada gárdase na micro:BIT receptora nunha variable que por defecto se chama “receivedString”. Usarase o bloque **mostrar cadea**, para visualizar o nome do botón premido. Tras un segundo apagarase o panel LED. Non esquecer establecer ao inicio o grupo de comunicación.

O código da tarxeta receptora quedaría da seguinte forma:



Tras volcar o código en cada unha das micro:BIT poderase comprobar o seu funcionamento. Tamén se pode usar o simulador, o único que hai que ter en conta que só tras premer un dos botóns actívese o emisor e aparece o simulador da segunda micro:BIT.

Proposta.

Modificar o código para que tamén se envíe que foron premidos os dous botóns á vez e se amose en pantalla A+B.

Reto 17. Chivato a distancia

A gran maioría dos nenos prefire alargar a hora de irse á cama. Para moitos deles, ese periodo de estar a soas e ás escuras resúltalles canto menos inquedande. A solución pasa, nalgúnhas ocasións, por prender a luz do dormitorio cando os pais se foron.

Obxectivo.

Deseñar un programa para que a micro:BIT avise mediante unha alarma visual e acústica que se prendeu a luz no cuarto dos nenos.

Descrición do código.

Usaranse dúas micro:BIT, a emisora enviará o valor do sensor de luz, a receptora avaliará ese valor e se é maior de 50 (valor teórico) fará sonar unha alarma.

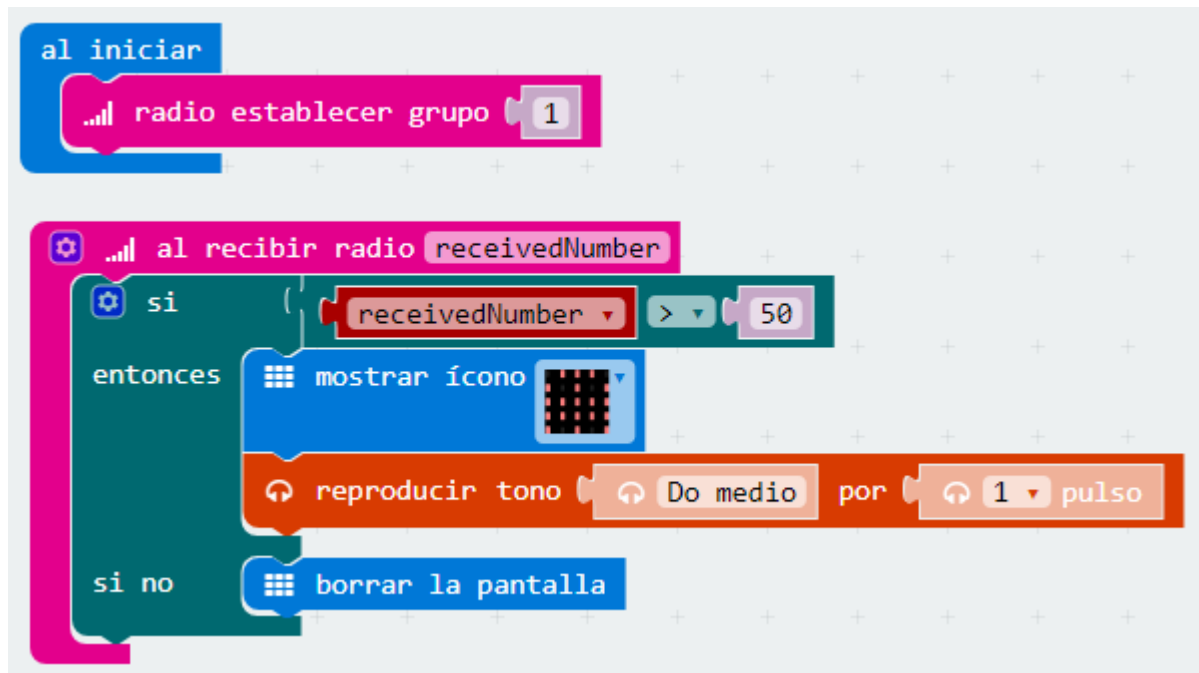
micro:BIT emisora:

Ao iniciar establecerase o grupo de emisión. Enviarase o valor do nivel de luz mediante o bloque **radio send number**, establecerase unha pausa de 1 segundo e creárase a animación dun corazón latindo para confirmar que o programa funciona e está emitindo o valor medido polo sensor de luz..



micro:BIT receptora:

Ao iniciar establecerase o grupo de emisión. O valor emitido gárdase na variable “receivedNumber”, se este dato é maior a 50 (valor teórico) amosarase o icono dunha pantasma e emitirá un son. En caso contrario apagarase a pantalla.



Proposta.

Modificar o código para que sexan 2 os dormitorios supervisados e na micro:BIT receptora se identifique o cuarto onde se prendeu a luz.

Pins de entrada e saída

A micro:BIT dispón de 25 conectores situados no bordo inferior . A través deles poderanse programar motores, LEDs ou calquera outro compoñente ou sensor externo.

Reto 18. Hai continuidade?

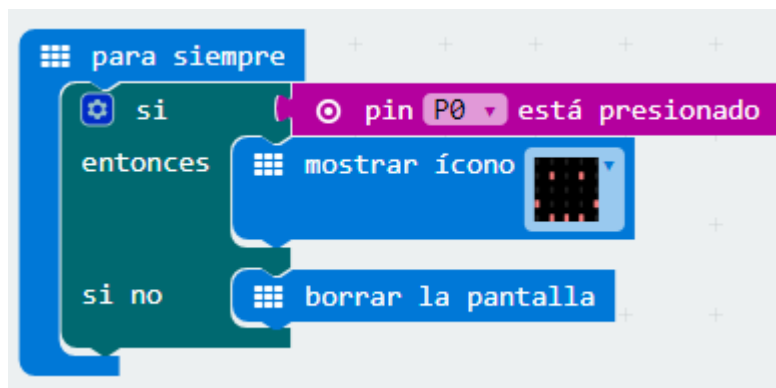
Nalgunas ocasións interesa coñecer se un cable ten continuidade en toda a súa lonxitude. Para comprobar a continuidade sóese usar un multímetro, medindo a resistencia do conductor. Unha lectura de resistencia infinda indicaría unha interrupción nalgunha parte do interior do cable.

Obxectivo.

Deseñar un programa que permita comprobar se un cable ten continuidade. Se non está cortado, mostrará unha cara feliz.

Descrición do código.

Usarase o bloque **pin (P0) está premido** da categoría **Entrada** para comprobar a continuidade. Proponse empregar o evento **para sempre** e usar un condicional que verifique se o P0 está premido. Se se cumpre a condición amósase unha carinha feliz, en caso contrario, o panel permanece apagado. O código podería quedar da seguinte forma:



Para comprobarol funcionamento pódese usar un cable con bananas conectados ao GND, se se conecta co P0 e o cable ten continuidade, deberá aparecer unha carinha feliz no panel LED.

Proposta.

Modificar o código para que cando o cable a comprobar teña continuidade faga sonar a melodía “jump up” e se non ten continuidade que se emita a melodía “jump down”.

Reto 19. Movendo un servomotor

Un servomotor (tamén chamado servo) é un dispositivo semellante a un motor de corrente continua que ten a capacidade de situarse en calquera posición dentro do seu rango de operación, e manterse estable na devandita posición.

Obxectivo.

Deseñar un programa que ao premer o botón A o servo se posicione a 180°, se se acciona o premedor B debe situarse a 0° e a premer conxuntamente A+B situarase a 90°.

Descrición do código.

O bloque **escribir servo pin (P0) a (180)** que serve para posicionar o servo, atópase situado na categoría **Pines**, que aparece ao premer na sección Avanzado do menú principal.

Pódese elegir en qué pin se conecta o servo e con qué ángulo se posiciona o brazo do servo. Usarase o bloque **ao premerse o botón**, para iniciar os diferentes programas segundo o premedor accionado.

El programa quedará da seguinte forma:



No simulador aparece a imaxe dun servo e fará o programado segundo o premedor accionado.

Proposta.

Crear un código que permita mover o servo en pasos de 15° ao premer o botón A e devolva o brazo a 0° ao accionar o premedor B.

~ Biografía do autor ~

José Francisco Muñoz Fernández (Almería, 1968). Arquitecto superior, dende que caeu nas súas manos un ordenador quedou fascinado por esta ciencia dixital. A tecnoloxía catalizou o seu innato espírito investigador e innovador e levouno a integrala en todas as facetas da súa vida formativa e laboral.

Tras anos exercendo como arquitecto, decide dar máis protagonismo as súas outras dúas paixóns, a tecnoloxía e a educación, comezando a exercer como mestre de robótica.

Esta experiencia docente sérvelle para comprobar o gran potencial que ten a robótica nas aulas e empúrrao a querer axudar a outros docentes na posta en práctica rápida e eficaz da Robótica Educativa nos seus centros.

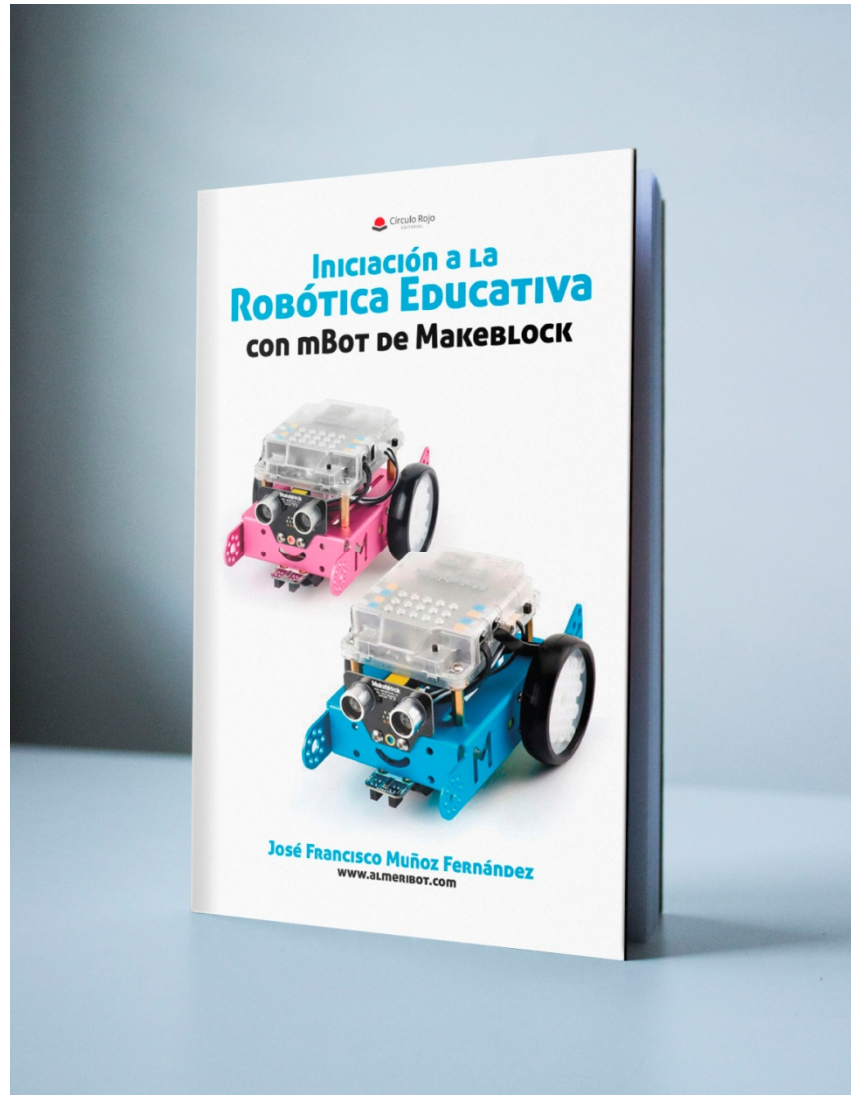


~ Otros libros do autor ~

“Iniciación á Robótica Educativa con mBot de Makeblock”

Disponible para a súa compra en:

<https://www.makeblock.es/productos/libro-iniciacion-robotica-educativa-mbot/>



~ Queres participar nesta guía? ~

En Microes.org creamos guías didácticas como a que tes nas túas mans para axudar aos mestres e usuarios a sacar o máximo partido a micro:bit.

Se es experto en robótica e tes algún exercicio orixinal que queres compartir coa comunidade ponte en contacto connosco.

micro:es

Comunidad micro: bit en España

Dubidas? ¿Quieres colaborar coa comunidade? Contáctanos:

e-mail: info@microes.org

teléfono: 945 29 80 50 (de 8:00 a 14:00)

Podes comprar a tarxeta micro:bit e infinidad de complementos compatibles en:

robotopia

<http://robotopia.es/>