

Robótica para profes

Micro:bit y Lego Spike Prime

Noelia Rubio Martín

Programación estructurada:

SECUENCIA Y CONDICIONAL



La **programación estructurada** es un paradigma de la programación que se basa en tres estructuras básicas: secuencia, condicional y bucles.

SECUENCIA

Hace referencia al orden en el que se ejecutan las instrucciones que forman parte de un programa. Cuando diseñamos un algoritmo y escribimos el programa, debemos pensar en las instrucciones y decidir en qué orden las ponemos.

Hay que tener presente que las instrucciones se ejecutan una detrás de otra, tal como aparecen en el programa, y este orden puede determinar el éxito del resultado.

CONDICIONAL

Durante la ejecución de un programa se pueden dar circunstancias que alteren su ejecución. Por este motivo, necesitamos poder tomar decisiones alternativas para adaptarnos a situaciones imprevistas.

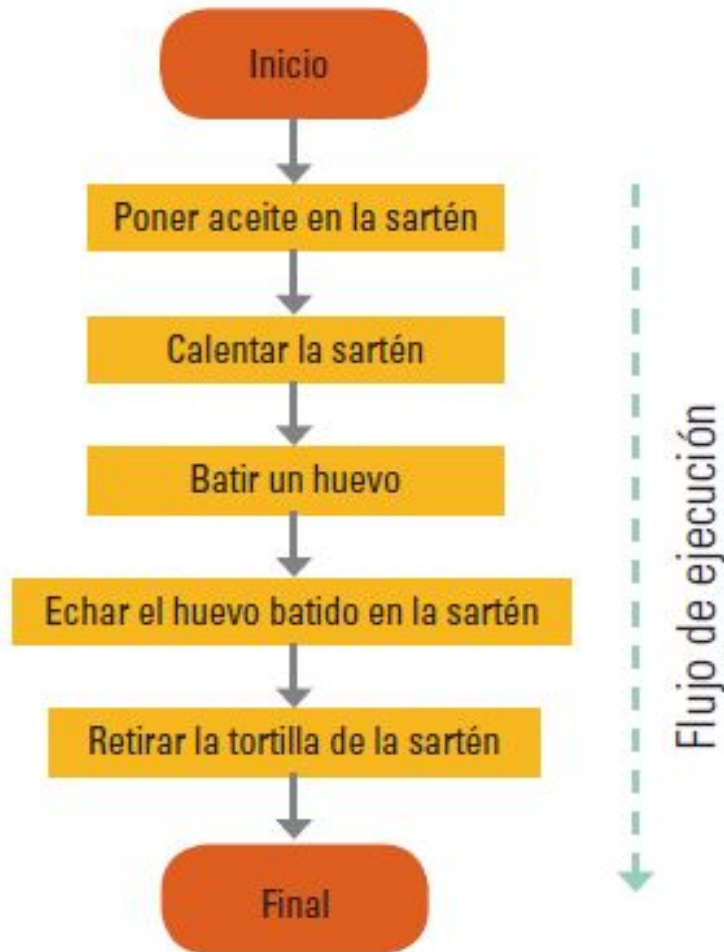
La estructura condicional nos permite añadir una pregunta en cualquier posición de nuestro programa. Esta pregunta sólo puede tener dos únicas respuestas posibles: SÍ o NO.

SECUENCIA

EJEMPLO

Vamos a diseñar un algoritmo para resolver el problema *¿Cómo cocinar una tortilla?*

1. Enumerar la lista de instrucciones necesarias:
 - a. Retirar la tortilla de la sartén
 - b. Echar el huevo batido en la sartén.
 - c. Poner aceite en la sartén.
 - d. Batir un huevo.
 - e. Comer la tortilla.
 - f. Calentar la sartén.
2. Ordenar las instrucciones de forma correcta. Recuerda que el orden de las instrucciones de la secuencia determina el resultado de la ejecución.
3. Indicar el inicio y el final del algoritmo.
4. Todos los conectores (flechas y líneas) deben indicar el sentido del flujo de la ejecución.



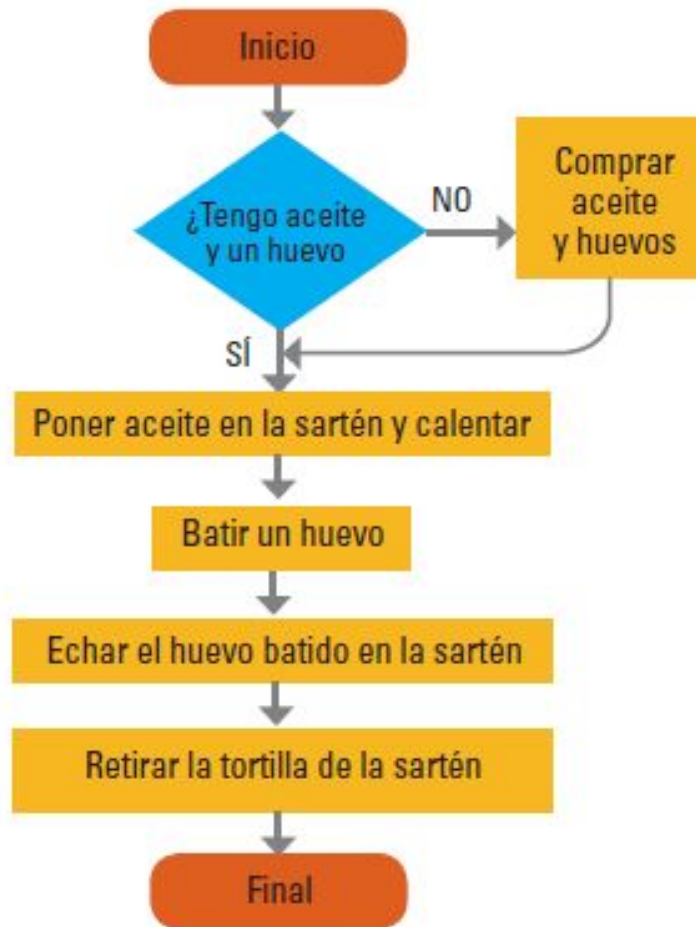
CONDICIONAL

EJEMPLO

¿Cómo debemos actuar si no tenemos aceite y un huevo para cocinar la tortilla?

- 1.º Deberíamos comprobar que tenemos estos ingredientes antes de seguir con la receta.
- 2.º Añadimos el bloque condicional con la pregunta que nos hemos planteado: *¿Tengo aceite y un huevo?*

El bloque condicional siempre se representa con forma de rombo y contiene la pregunta y dos conectores de salida: la respuesta afirmativa (SÍ) y la negativa (NO).

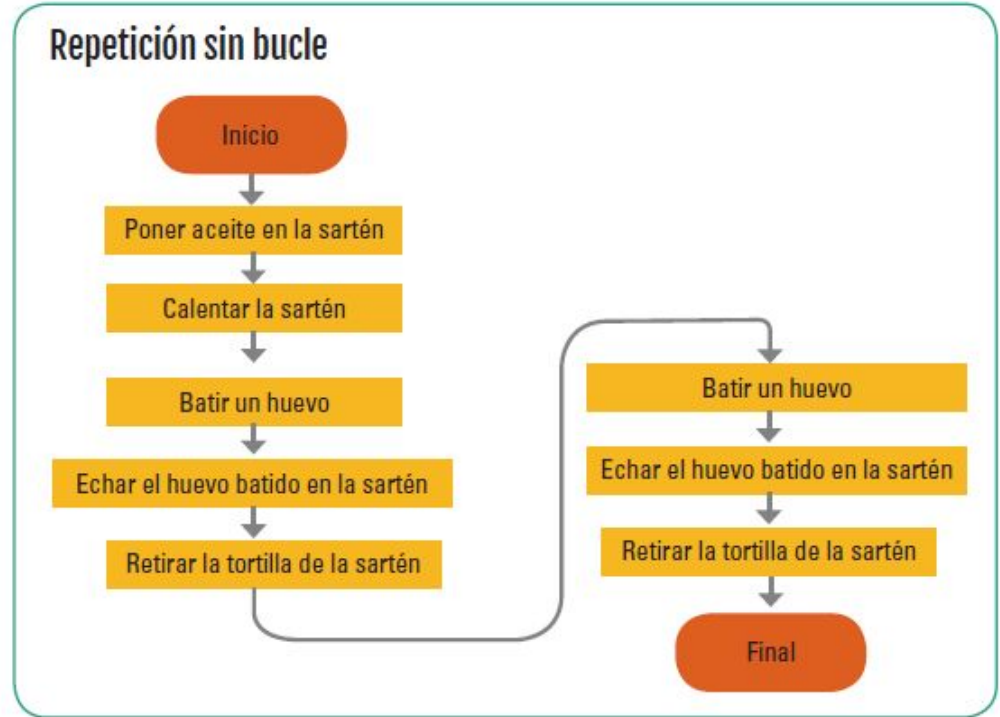


BUCLES Y VARIABLES

BUCLES

Al diseñar un programa, nos podemos encontrar que se tienen que repetir determinadas tareas para solucionar un problema. Una manera de indicar en un programa la repetición de instrucciones es simplemente copiandolas, una detrás de otra.

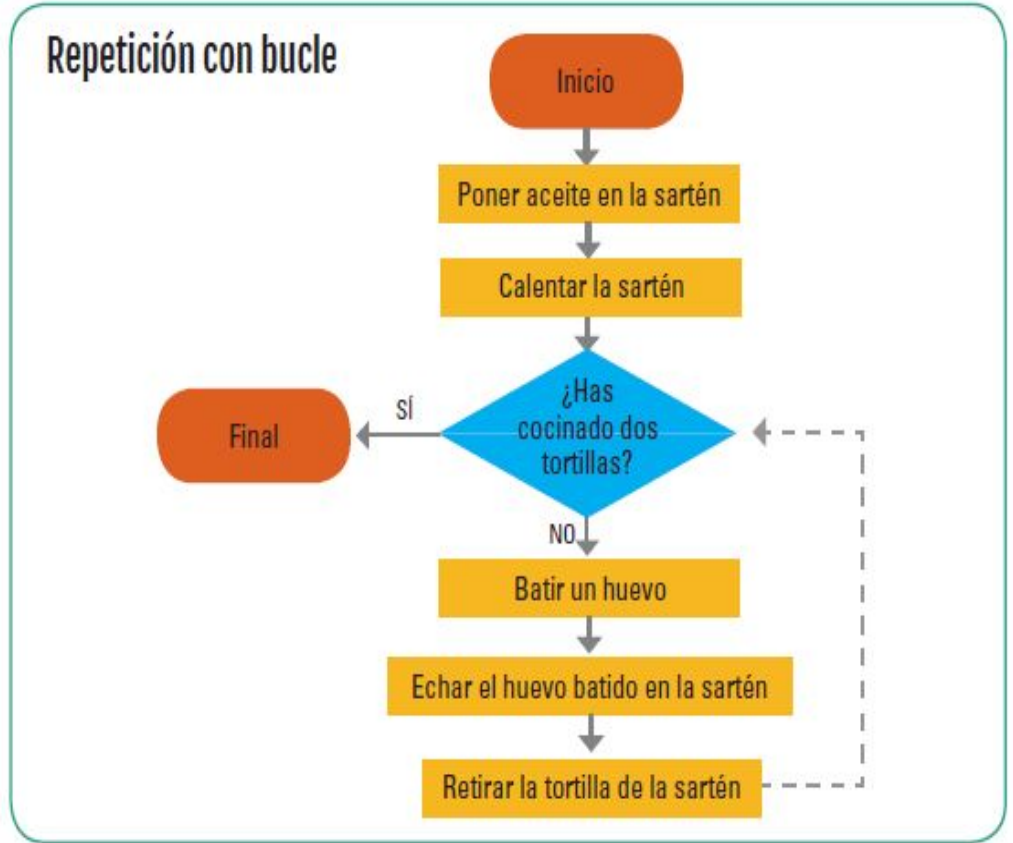
Imaginemos un algoritmo para solucionar el problema *¿Cómo cocinar dos tortillas?:*



BUCLES Y VARIABLES

BUCLES

Esta opción da lugar a programas excesivamente largos y poco eficientes. Un bucle es un camino para volver a repetir un conjunto de pasos del mismo programa sin necesidad de reescribirlos. Para ello se deben agrupar las instrucciones que hay que repetir y añadir un condicional que determine el final de la repetición.



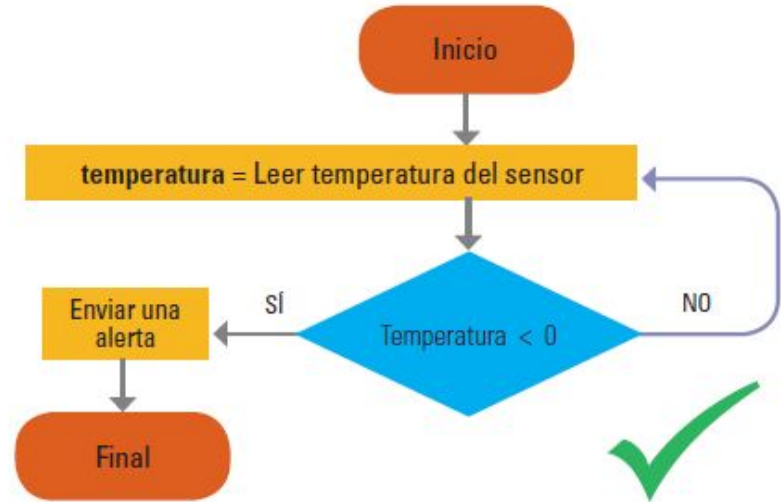
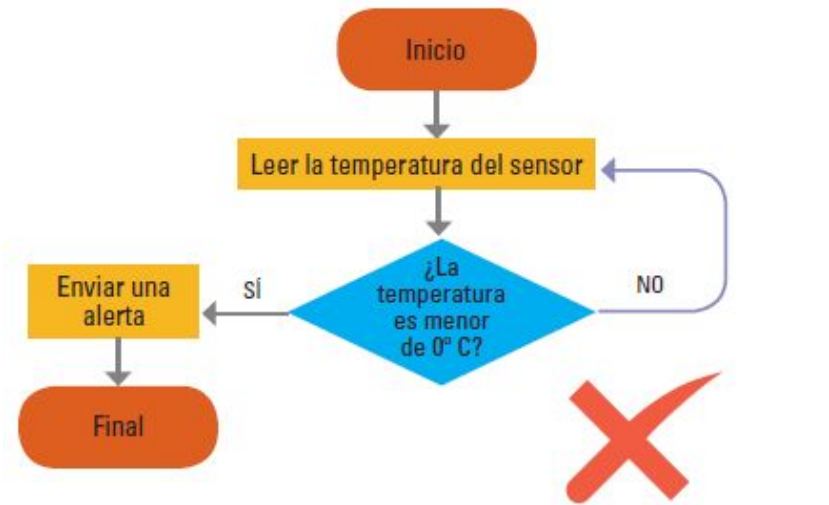
BUCLES Y VARIABLES

VARIABLES

Un ordenador o sistema computacional tiene la capacidad de ejecutar instrucciones y hacer cálculos muy complejos a gran velocidad. Pero, a diferencia de los humanos, por defecto los ordenadores no recuerdan las instrucciones ejecutadas ni los resultados calculados previamente.

Los programadores son los responsables de decidir e indicar en el programa qué y cuándo se debe guardar en la memoria y de que se pueda consultar en las siguientes instrucciones.

La variable nos posibilita guardar un valor, tanto en números como en letras. Es necesario identificar las variables con un nombre que nos permita hacer referencia en el programa a su contenido, consultarlo o modificarlo.



ACTIVIDAD

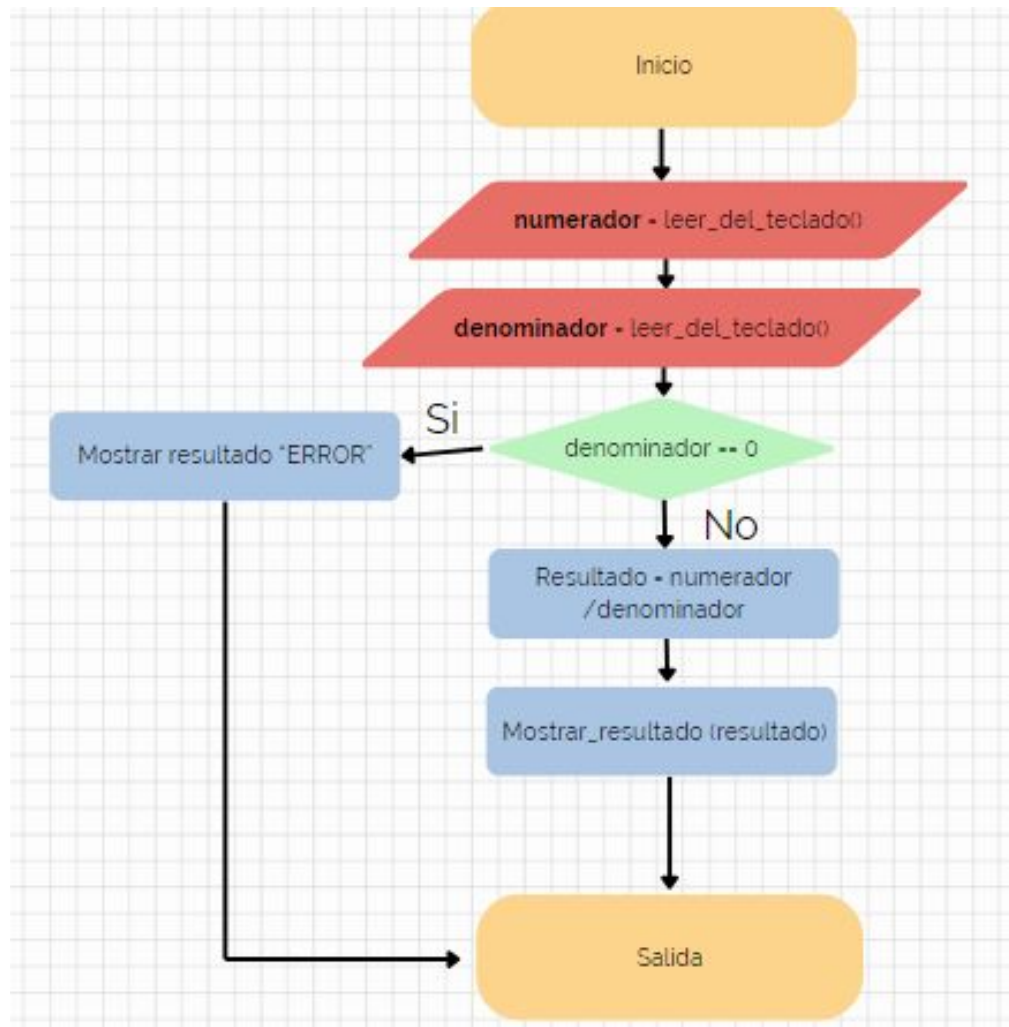
Diseña el diagrama de flujo de un sistema que calcule la división de dos números. El programa debe leer del teclado dos números (numerador y denominador) y dividirlos. Debes tener en cuenta:

— La división entre 0 no es posible. Si el denominador es igual a 0, la pantalla debe presentar un mensaje de error.

— En cualquier otro caso hay que mostrar el resultado de la operación.

— Dispones de dos operaciones: `leer_del_teclado` y `mostrar_resultado`. Sugerencia: usa tres variables: numerador, denominador y resultado.

SOLUCIÓN

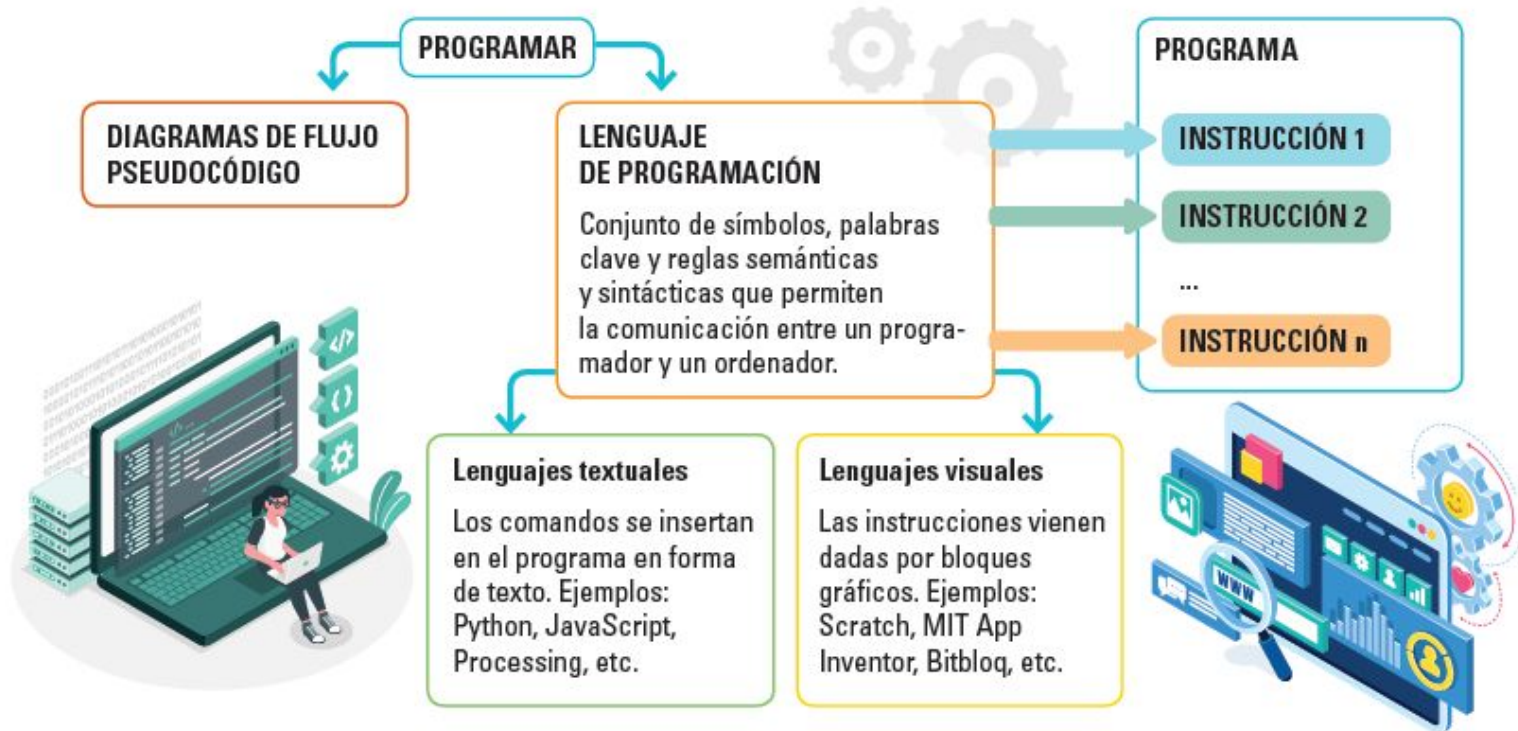


LENGUAJES DE PROGRAMACIÓN



¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?

Es el conjunto de símbolos (gráficos y/o textuales) y reglas (semánticas y/o sintácticas) que permiten escribir programas.



PROGRAMACIÓN CON MICRO:BIT

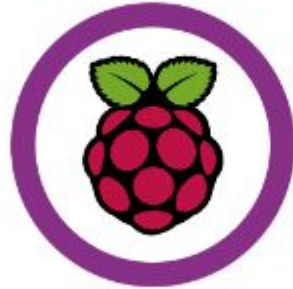


PROGRAMACIÓN CON MICRO:BIT

Una **placa de circuito impreso (o PCB)**, es una superficie no conductora con un circuito integrado formado por pistas o canales de un material conductor. Estas pistas conectan los diferentes componentes electrónicos de la placa mediante electricidad.



micro:bit



Raspberry Pi



Makey Makey



Arduino

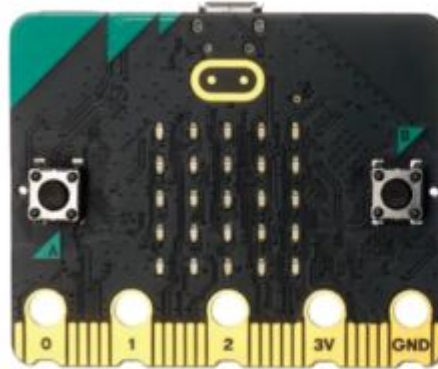
PLACA MICRO:BIT

Permite su programación e interactuar con todos los recursos que tiene integrados, como botones frontales, matriz de leds, sensores, etc.

Con micro:bit se pueden crear:

- Programas sencillos, para los que solamente se necesita la propia placa, como hacer un programa que simule los latidos del corazón en los leds, así como su sonido.

- Programas más avanzados y complejos, en los que podemos emplear componentes externos diseñados expresamente para poder conectarse con micro:bit, como un robot seguidor de líneas o un sistema que acciona un ventilador cuando se llega a determinada temperatura.



Vista delantera



Vista trasera

PROGRAMACIÓN DE LA PLACA MICRO:BIT

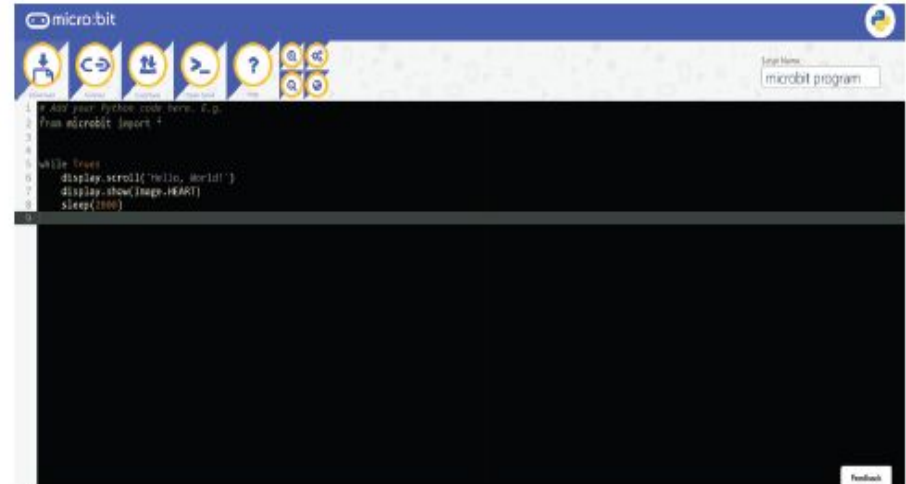
Se necesita un editor de código, que es una herramienta *software* que permite escribir el código de los programas.

Se pueden utilizar diferentes editores de código. Python y MakeCode cuentan con la acreditación oficial de la fundación micro:bit y son los que se usan principalmente para programar la placa.

Python

Desde la plataforma micro:bit se nos ofrece un editor de código para el lenguaje Python. Este lenguaje textual multiplataforma de código abierto puede emplearse para distintos propósitos y ofrece más potencia y posibilidades que un lenguaje de tipo visual.

No obstante, Python requiere más experiencia en el desarrollo de programas y conocimientos más profundos de programación.

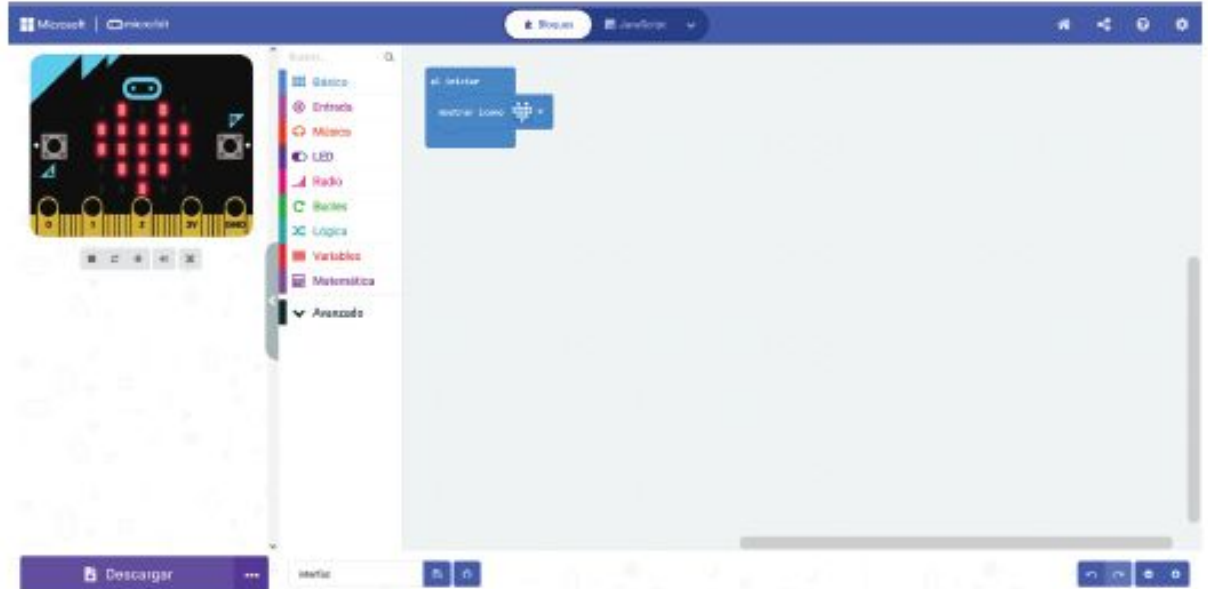


MAKECODE

Es un editor de código de tipo visual desarrollado por Microsoft. Se trata de una aplicación web, por lo que no es necesario instalar nada en el ordenador.

MakeCode permite programar los proyectos que queremos ejecutar en la placa con un lenguaje visual, como Scratch, o textual, como Python y JavaScript. Es una buena opción para iniciarse en la programación y adentrarse en el mundo de los lenguajes textuales.

MAKECODE WEB



BLOQUES DE MAKECODE

1 Bloques azules

Dan acceso a las funciones básicas de micro:bit.



2 Bloques rojos

Permiten generar notas musicales.



3 Bloques lilas

Permiten configurar la matriz de leds.



4 Bloques fucsias

Permiten generar y recibir señales (eventos) y dan acceso a los datos de los sensores.



5 Bloques verdes

Permiten crear estructuras de programación, como bucles de programación.



6 Bloques turquesa y morados

Posibilitan hacer operaciones lógico-matemáticas en el programa.



7 Bloques rosas

Permiten la comunicación entre placas mediante señales de radio.




EJEMPLO

Entra en la plataforma en línea de MakeCode y programa un proyecto que muestre en la matriz de leds una cuenta atrás, los números del 5 al 1, cuando se presione el botón A. Si dispones de alguna placa en tu centro, trata de conectarla y probar tu programa.

1. Entrar en la web makecode e iniciar sesión con la cuenta de google o de microsoft

The screenshot shows the Microsoft MakeCode website. At the top, there is a blue navigation bar with the Microsoft logo, a link to 'micro:bit', a settings gear icon, and a 'Iniciar sesión' button with a user profile icon. Below the navigation bar is a large hero section with a background image of a micro:bit board, a USB cable, and a blue keyboard. The text in the hero section reads '¿Nuevo? ¡Empieza aquí!' and 'Empezar tutorial'. Below the hero section is a 'Mis proyectos' section with a 'Ver todos' link and an 'importar' button. There are two main buttons: 'Nuevo proyecto' (with a plus icon) and 'Proyectos en la nube' (with a cloud and person icon). At the bottom, there is a 'Tutoriales' section with a row of six tutorial cards. The first card has a red heart icon and the text '¿Nuevo? ¡Empieza aquí!'. The other cards show various icons: a micro:bit board, a yellow smiley face, a blue micro:bit board, two people shaking hands, and a person holding a micro:bit board.

2. Crear nuevo proyecto y ponerle el nombre cuenta atrás. Completar el tutorial

Crear un proyecto 🥰🥰🥰 

Pon un nombre a tu proyecto.

▼ Opciones del código

 ▼

crear ✓

3. Borrar el bloque al iniciar, arrastramos hacia el panel de bloques

Microsoft | micro:bit

Bloques javascript

Buscar...

- Básico
- Entrada
- Música
- LED
- Radio
- Bucles
- Lógica
- Variables
- Matemática
- Extensiones
- Avanzado

al iniciar

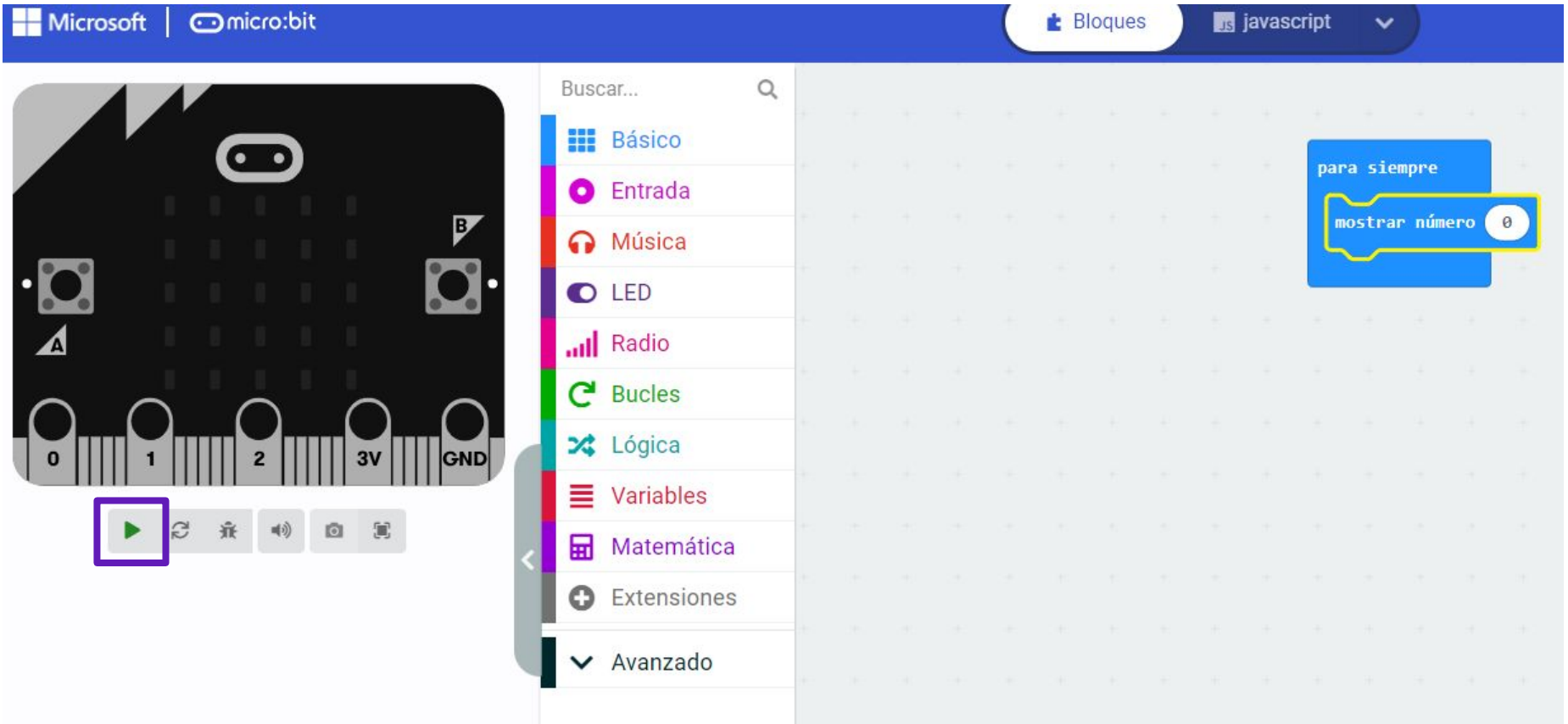
para siempre

4. Hacemos clic en el panel de básico y añadimos mostrar número o arrastrandolo hasta que encaje en el bloque para siempre

The image shows the Microsoft MakeCode editor interface for the micro:bit. On the left, there is a visual representation of the micro:bit board. The main workspace is divided into three sections: a left sidebar with a category menu, a central block palette, and a right workspace for code blocks.

- Left Sidebar:** A search bar at the top is followed by a list of categories: Básico, Entrada, Música, LED, Radio, Bucles, Lógica, Variables, Matemática, Extensiones, and Avanzado.
- Central Block Palette:** Titled 'Básico', it contains several blocks: 'mostrar número' (with a value of 0), 'mostrar LEDs', 'mostrar ícono', 'mostrar cadena' (with the text 'Hello!'), 'borrar la pantalla', 'para siempre', and 'al iniciar'.
- Right Workspace:** A 'para siempre' loop block is placed on the grid. A yellow highlight is shown on the 'para siempre' block in the palette, and another yellow highlight is on the 'mostrar número' block within the loop in the workspace, indicating the process of dragging and snapping the block into the loop.

5. Iniciamos la simulación en el botón



Microsoft | micro:bit

Bloques javascript

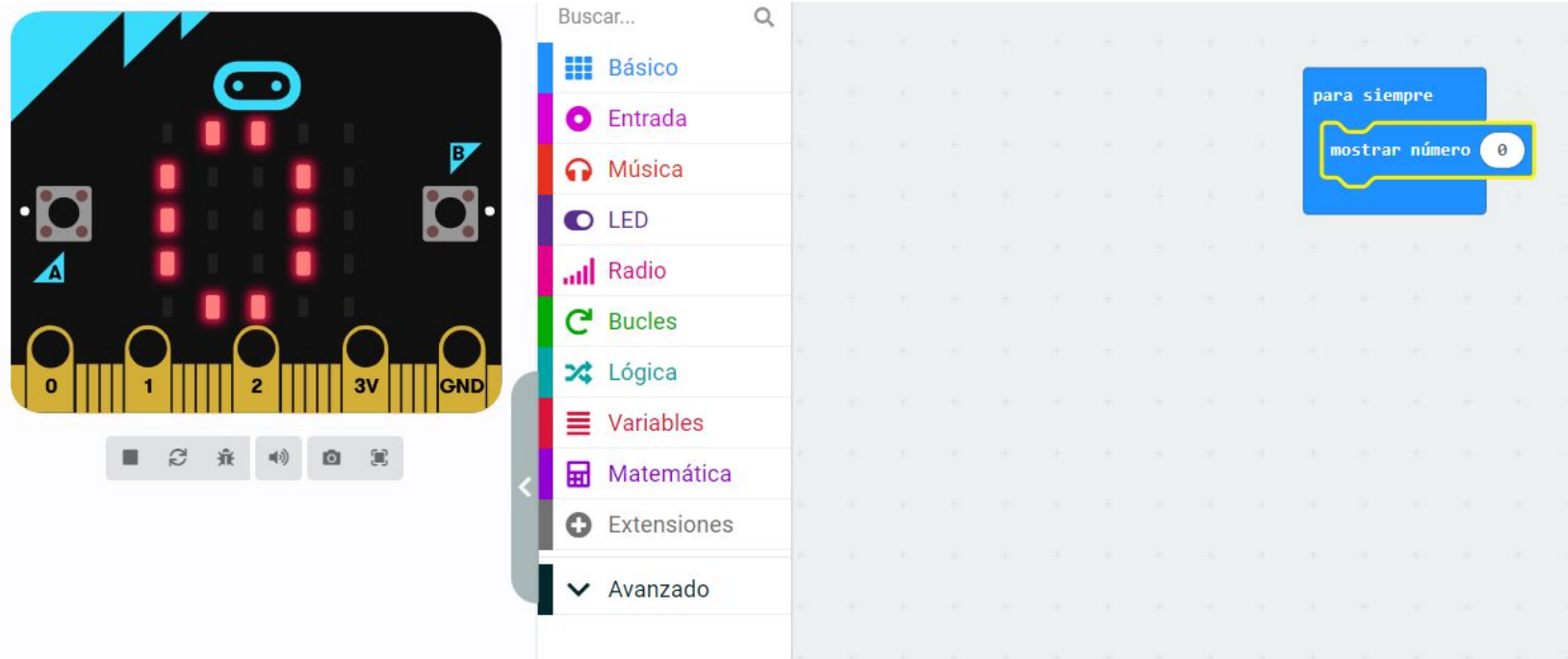
Buscar...

- Básico
- Entrada
- Música
- LED
- Radio
- Bucles
- Lógica
- Variables
- Matemática
- Extensiones
- Avanzado

para siempre

mostrar número 0

6. Y vemos en la pantalla que se cumple el programa



The image shows the Scratch IDE interface. On the left, a micro:bit is connected to a screen. The screen displays a 'para siempre' (forever) loop block containing a 'mostrar número 0' (show number 0) block. The Scratch IDE interface includes a search bar at the top, a category menu on the left, and a workspace on the right.

Buscar... 🔍

- Básico
- Entrada
- Música
- LED
- Radio
- Bucles
- Lógica
- Variables
- Matemática
- Extensiones
- Avanzado

para siempre

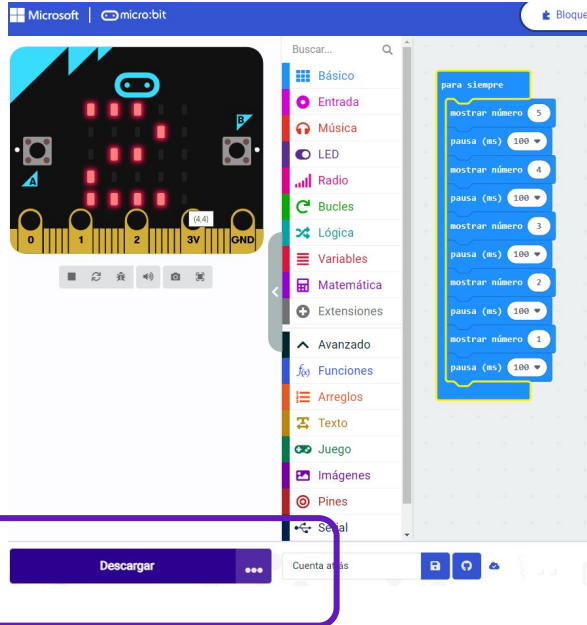
mostrar número 0

7. Completamos el código arrastrando los bloques necesarios para crear la cuenta atrás

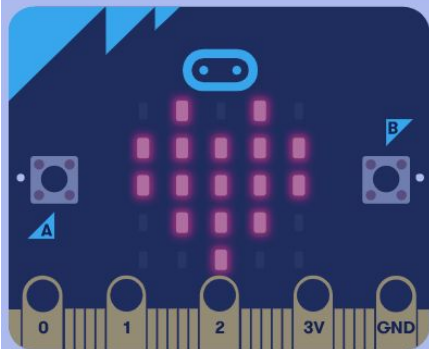
The image shows the Microsoft MakeCode editor interface for a micro:bit. On the left is a virtual representation of the micro:bit board with its pins labeled 0, 1, 2, 3V, and GND. Below the board are icons for play, refresh, save, volume, camera, and a help icon. In the center is a block palette with a search bar and categories: Básico, Entrada, Música, LED, Radio, Bucles, Lógica, Variables, Matemática, Extensiones, Avanzado, Funciones, Arreglos, and Texto. On the right, a script is being built within a 'para siempre' (forever) loop block. The script consists of the following blocks in order: 'mostrar número' (display number) with the value 5, 'pausa (ms)' (pause) with the value 100, 'mostrar número' with the value 4, 'pausa (ms)' with the value 100, 'mostrar número' with the value 3, 'pausa (ms)' with the value 100, 'mostrar número' with the value 2, 'pausa (ms)' with the value 100, 'mostrar número' with the value 1, and 'pausa (ms)' with the value 100. The entire script is enclosed in a yellow border.

8. Descargar el código a la placa

Tras conectar la micro:BIT al ordenador, aparece esta como una nueva unidad. Terminado el programa y comprobado su correcto funcionamiento, se debe pulsar sobre el icono de descarga, lo que copia el código al ordenador con un nombre por defecto. Otra opción es introducir el nombre elegido en la caja situada al lado del icono del disco y pulsar sobre este para descargar. Ya solo queda copiar el fichero



Copiado el código, el programa comienza a ejecutarse de forma automática. Si se quiere usar la micro:BIT desconectada del ordenador, se debe conectar un pack de pilas o batería al conector situado junto al conector microUSB e inmediatamente se ejecutará el código descargado.



Buscar...



Básico

Entrada

Música

LED

Radio

Bucles

Lógica

Variables

Matemáticas

Extensiones

Avanzado

para siempre

mostrar ícono

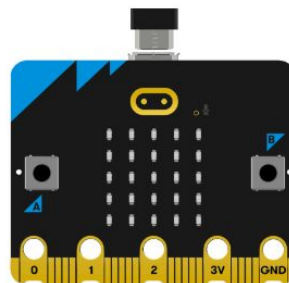


pausa (ms)

50



1. Conecta tu micro:bit a tu computadora

 **Siguiente**

Descargar



Reto 2. Animación de icono



2. Empareja tu micro:bit con tu navegador



Presione el botón Emparejar a continuación.

Aparecerá una ventana en la parte superior de su navegador.

Seleccione el dispositivo micro:bit y haga clic en Conectar.

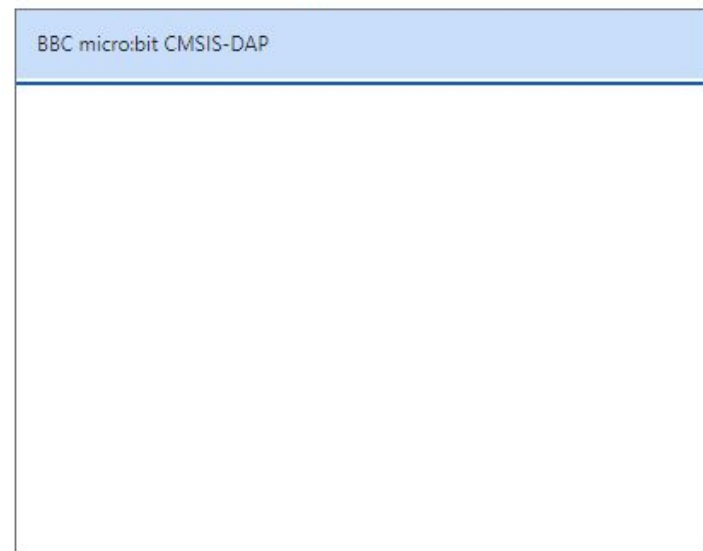


Descargar como archivo



Par

makecode.microbit.org quiere conectarse



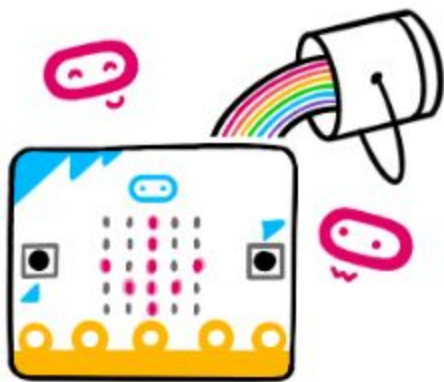
Conectar

Cancelar

✓ Conectado a micro:bit



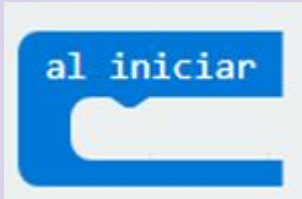
¡Tu micro:bit está conectado! Al presionar 'Descargar' ahora se copiará automáticamente tu código a tu micro:bit.



Descargar

RETO 1. Hola Mundo

Cuando se aprende a programar, se llama “Hola Mundo” al primer programa que se crea. Este código imprime el texto «¡Hola Mundo!» en un dispositivo de visualización. Objetivo. Al conectar la micro:BIT, se debe mostrar el texto «¡¡¡Hola Mundo!!!» en el panel LED de la tarjeta. Descripción del código. Se propone usar el evento al iniciar, situado en la categoría Básico. Este bloque permite ejecutar el código introducido, cuando se inicia la micro:BIT



A continuación se usará el bloque mostrar cadena, también localizado en la categoría Básico e introducir el texto “¡¡¡ Hola Mundo !!!”. Este bloque permite mostrar caracter a caracter, el texto introducido. El código quedará de la siguiente manera.



RETO 2 Animando los iconos

La animación es una técnica que logra crear sensación de movimiento a imágenes estáticas. Se consigue mediante una secuencia de imágenes ordenadas, que al ser mostradas consecutivamente, consiguen generar la ilusión visual de movimiento.

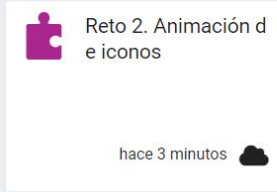
Objetivo. Se propone, que al conectar la micro:BIT, se muestre en el panel LED un corazón que late. Descripción del código. Se usará el evento “para siempre”, situado en la categoría Básico.. Situar en el interior el bloque anterior la instrucción mostrar icono, usar el llamado corazón. Agregar el bloque pausa ms (100) localizado en la categoría Básico. Repetir los dos bloques anteriores, pero seleccionando el icono “corazón pequeño”.

Podemos seguir el tutorial que ofrece MakeCode: Corazón intermitente como se muestra a continuación

Seguimos el tutorial que ofrece makecode

Mis proyectos [Ver todos](#)

 importar



Tutoriales



Flashing Heart

Make an animated flashing heart.

 Reproducir lección en vídeo

Bloques

Empezar tutorial

Python

Empezar tutorial

JavaScript

Empezar tutorial

Seguimos las instrucciones de cada paso

Microsoft | micro:bit

Corazón parpadeante - Step 2 of 6

Click on the **Basic** category in the Toolbox. Drag the **show leds** block into the **forever** block. Then in the **show leds** block, click on the squares to draw a heart design.

Caja de herramientas

Buscar...

Básico

para siempre

← 2 → Siguiente

El símbolo de la bombilla nos da pistas

Microsoft | micro:bit

Corazón parpadeante - Step 3 of 6

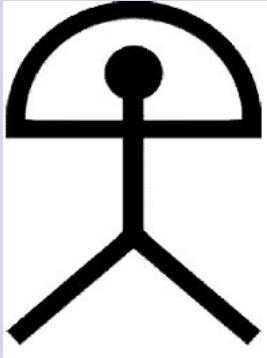
Drag another `show LEDs` block underneath the first.

The image shows the Microsoft MakeCode editor interface for a micro:bit. On the left side, there is a 3D model of the micro:bit board. On the right side, the code editor is visible, showing a 'para siempre' (forever) loop block containing two 'mostrar LEDs' (show LEDs) blocks. A yellow box highlights a lightbulb icon in the block palette, indicating a hint or tip. Another yellow box highlights the second 'mostrar LEDs' block in the code, suggesting it should be dragged underneath the first one. The text 'Drag another show LEDs block underneath the first.' is displayed above the code editor.

RETO 3 Diseñando iconos

MakeCode permite diseñar de forma gráfica los iconos a mostrar en el display LED. Para este menester dispone de un bloque específico situado en la categoría Básico. El bloque se llama “mostrar LEDs” y permite al usuario diseñar sus propios iconos marcando de forma independiente los LEDs que se quieren iluminar.

Objetivo. Diseñar un icono con forma de Indalo. Se trata de una figura rupestre encontrado en una cueva de la provincia de Almería y que representa una figura humana con los brazos extendidos y un arco sobre sus manos. Actualmente se considera un símbolo de la provincia de Almería.



RETO 4 Termómetro digital

Un termómetro es un instrumento que sirve para medir la temperatura; el más habitual consiste en un tubo capilar de vidrio cerrado y terminado en un pequeño depósito que contiene una cierta cantidad de mercurio o alcohol, el cual se dilata al aumentar la temperatura o se contrae al disminuir y cuyas variaciones de volumen se leen en una escala graduada.

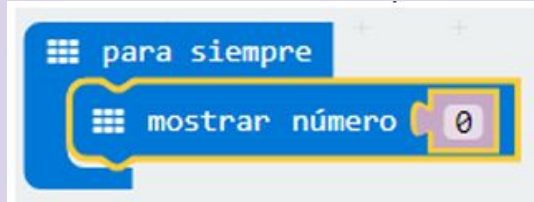
Los termómetros digitales son aquellos que, valiéndose de dispositivos transductores , utilizan circuitos electrónicos para convertir tensión en números las pequeñas variaciones de obtenidas, mostrando finalmente la temperatura en un visualizador.

Objetivo. Mostrar en el panel LED la temperatura detectada por el sensor de temperatura integrado en la micro:BIT.

RETO 4 Termómetro digital

Descripción del código.

Se propone iniciar el programa usando el evento “para siempre”. Añadir el bloque, localizado en la categoría Básico, mostrar número. Este bloque muestra en la pantalla LED el número introducido, desplazándose si es mayor de 1 cifra.



Sustituir el número “0” por el valor obtenido del sensor de temperatura.

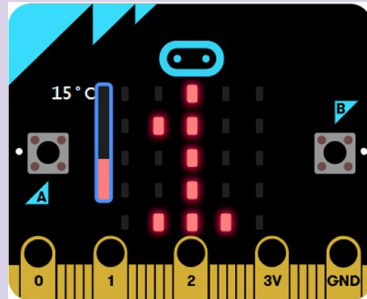


RETO 4 Termómetro digital

Este bloque se localiza en la categoría Entrada. A continuación se borra la pantalla y se introduce una pausa de 1000 ms. Quedando el programa de la siguiente forma:



En el simulador aparecerá una barra vertical que permite modificar la temperatura, siendo la temperatura marcada la mostrada en el panel LED.



RETO 4 Termómetro digital

Propuesta. Hacer un programa que al pulsar el botón A muestra la temperatura en grados celsius y al pulsar el botón B la temperatura mostrada sea en grados Fahrenheit.

RETO 5 Alarma de exposición al sol

En la actualidad hay estudios que demuestran la relación entre la exposición al sol y el riesgo de padecer cáncer de piel. Los dermatólogos recomiendan evitar la exposición al sol durante las horas de máxima radiación y el uso de cremas protectoras. Se propone crear, usando la micro:BIT, un dispositivo que avise que estamos expuestos a un valor de elevado de radiación solar.

Objetivo. Al iniciar el programa se indica que hay que pulsar el botón A para que muestre el valor de la intensidad del sol. Al pulsar el botón se mostrará una carita feliz si el valor es inferior a 175(*) y una carita triste si es superior. Pasados 3 segundos se pasará a modo de ahorro de energía, quedando a la espera de que se vuelva a pulsar el botón A.

Nota.- El valor usado como límite, carece de toda validez científica.

RETO 5 Alarma de exposición al sol

Descripción del código. Al iniciarse la micro:BIT se mostrará el texto “Pulsar A”



Su usará el evento al presionar el botón A, para iniciar la medición del valor de la exposición. Se introducirá la condición si ... entonces, si no, para comprobar si se está por encima o por debajo del valor teórico tomado como límite. Si el valor del nivel de luz es menor a 175, se muestra una carita feliz y a los 3 segundos se apaga la pantalla. En caso contrario se muestra una carita triste y se entrará en modo de ahorro de energía pasados 3 segundos. El programa quedaría de la siguiente forma.

Propuesta. Modificar el código, para que además de mostrar una “carita triste” se haga sonar un aviso acústico.

RETO 5 Alarma de exposición al sol



RETO 6 Aviso sonoro de orientación Norte

Cuando se necesita orientarse con un mapa, lo que primero que hay que hacer, es situar el mapa con orientación norte. Para ellos se debe usar una brújula.

Objetivo. Crear un programa que mediante un aviso acústico se advierta de que la micro:BIT está orientada al Norte.

Descripción del código. Usar el evento para siempre, para iniciar el programa. Crear una variable donde se almacenará el valor del sensor obtenido por el bloque dirección de la brújula (°). Se introduce el condicional si entonces si no para comprobar en qué intervalo se encuentra el valor obtenido por el sensor. Si el valor se encuentra entre 315° y 45° la micro:BIT se encuentra orientada al Norte, por lo que se mostrará una N y se reproducirá una nota. Si no se cumple la hipótesis, se borra la pantalla.

Propuesta. Hacer un código que ubique de forma más precisa el Norte Magnético.

RETO 6 Aviso sonoro de orientación Norte

Solución



RETO 7 Moviendo un servomotor

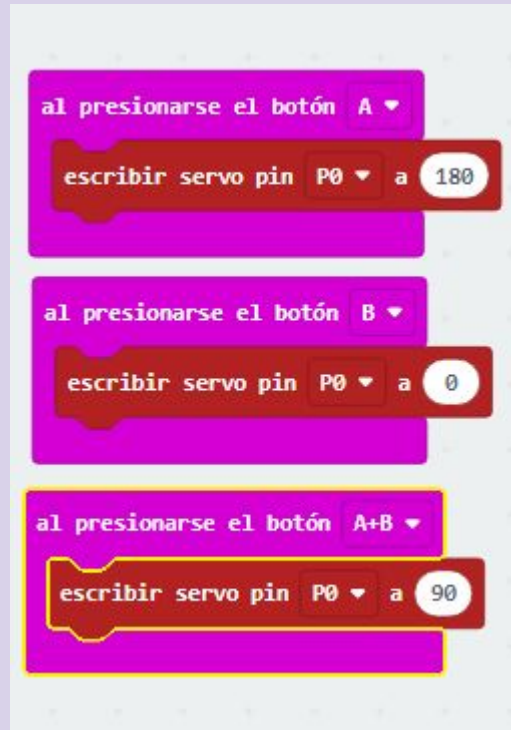
Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Objetivo. Diseñar un programa que al pulsar el botón A el servo se posicione a 180° , si se acciona el pulsador B se debe situar a 0° y al pulsar conjuntamente A+B se ubicará a 90° . Descripción del código. El bloque escribir servo pin (P0) a (180) que sirve para posicionar el servo, se encuentra ubicado en la categoría Pines, que aparece al pulsar en la sección Avanzado del menú principal. Se puede elegir en qué pin se conecta el servo y con qué ángulo se posiciona el brazo del servo. Se usará el bloque al presionar el botón, para iniciar los diferentes programas según el pulsador accionado.

Propuesta. Crear un código que permita mover el servo en pasos de 15° al pulsar el botón A y devuelva el brazo a 0° al accionar el pulsador B. Y mostrar por pantalla los grados

RETO 7 Moviendo un servomotor

Solución



PROPUESTAS

1. Dado digital
2. Juego de piedra, papel y tijera
3. Crear una melodía
4. Juego del pong

PROGRAMACIÓN CON LEGO SPIKE PRIME



Descargamos el programa spike lego education de la tienda de microsoft

Microsoft Store

Inicio

Aplicaciones


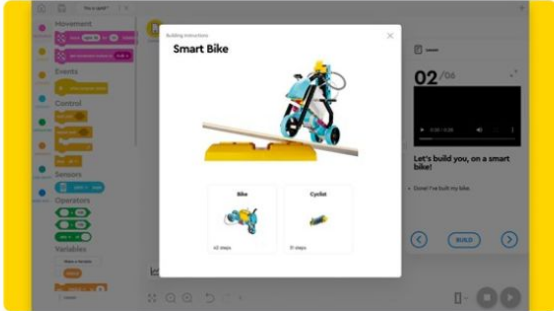
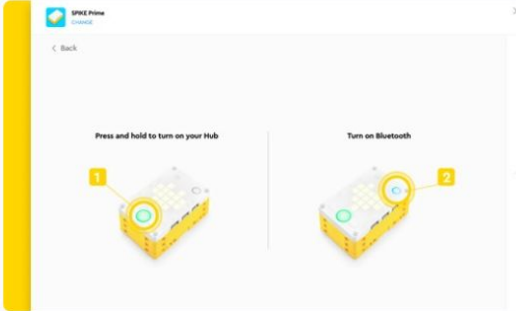
Juegos

Juego retro

Entretimiento


lego

Capturas de pantalla



SPIKE™ LEGO® Education
LEGO Education

Abrir

3,7  Promedio

3 Clasificación

Despierta el interés de todos los alumnos de forma lúdica por el aprendizaje STEAM con la app SPIKE™ de LEGO® Education. Diseñada para su uso con la gama SPIKE™ de LEGO® Education, la aplicación SPIKE es una potente herramienta que ayuda a los docentes a brindar un aprendizaje STEAM atractivo y práctico.

Cuando se combina con SPIKE Essential o SPIKE Prime, la aplicación SPIKE permite a los alumnos:

- Desarrollar habilidades de pensamiento computacional usando una progresión de programación basada en bloques de iconos, bloques de palabras y finalmente en texto.
- Construir, probar y experimentar, avanzando progresivamente desde creaciones sencillas hasta estructuras complejas.
- Aprender a programar, probar y retocar diseños, depurar y optimizar, crear historias con las Minifiguras LEGO®, prepararse para las competiciones de FIRST® LEGO® League y mucho más.

*** IMPORTANTE***

[Mostrar más](#)

Educación

Seleccionamos spike prime

Selecciona tu solución SPIKE™



SPIKE

Essential



SPIKE

Prime



Le damos a comenzar para ver un tutorial de actividades



The screenshot displays the SPIKE Prime software interface. On the left is a sidebar with navigation options: Inicio, Comenzar, Unidades, Construir, Mis proyectos, Ayuda, and Ajustes. The main content area features a large blue banner for the 'Primeros pasos con SPIKE™ Prime' tutorial, which includes a 'COMENZAR' button and a sub-header: '¡Vamos a aprender a usar SPIKE Prime en seis sencillos pasos!'. Below the banner is a 'Proyectos recientes' section with a 'Nuevo proyecto' button. At the bottom, there are two cards: 'Unidades didácticas' and 'Instrucciones de construcción', each with an illustration and a brief description.

Primeros pasos con SPIKE™ Prime

¡Vamos a aprender a usar SPIKE Prime en seis sencillos pasos!

COMENZAR

Proyectos recientes

Nuevo proyecto

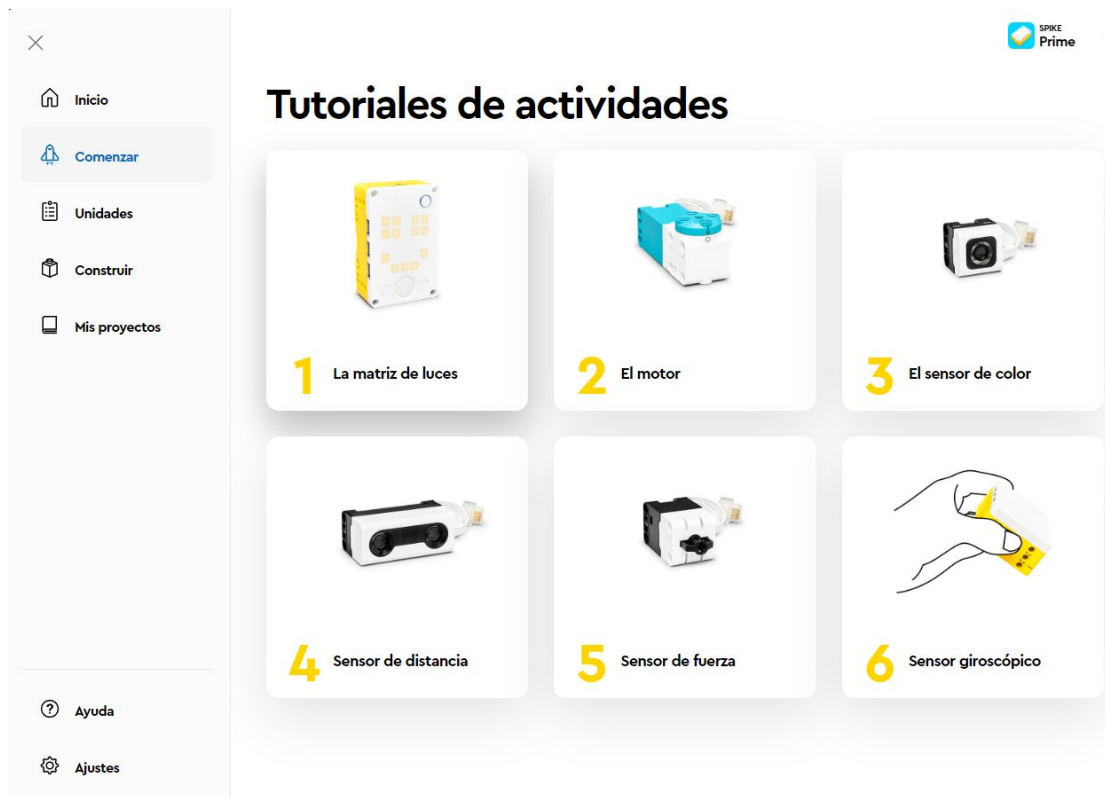
Unidades didácticas

Las lecciones SPIKE Prime se agrupan en unidades temáticas para despertar el interés de los alumnos de secundaria por el aprendizaje

Instrucciones de construcción

Una colección de instrucciones de construcción de todos los modelos SPIKE Prime.

En cada uno de los siguientes elementos encontraremos un tutorial de manejo de cada uno de ellos



The screenshot displays the SPIKE Prime application interface. On the left is a sidebar menu with the following items: 'Inicio' (Home), 'Comenzar' (Start) - which is highlighted in blue, 'Unidades' (Units), 'Construir' (Build), 'Mis proyectos' (My projects), 'Ayuda' (Help), and 'Ajustes' (Settings). The main area is titled 'Tutoriales de actividades' (Activity Tutorials) and features a grid of six tutorial cards, each with a numbered title and an image of the component:

- 1** La matriz de luces (The light matrix) - Image of a yellow and white light matrix.
- 2** El motor (The motor) - Image of a blue and white motor.
- 3** El sensor de color (The color sensor) - Image of a black and white color sensor.
- 4** Sensor de distancia (Distance sensor) - Image of a black and white distance sensor.
- 5** Sensor de fuerza (Force sensor) - Image of a black and white force sensor.
- 6** Sensor giroscópico (Gyroscopic sensor) - Image of a hand holding a yellow and white gyroscopic sensor.

In the top right corner of the app, the 'SPIKE Prime' logo is visible with a dropdown arrow.

En la pestaña de unidades podemos descargar las unidades didácticas que nos ofrecen.

The screenshot displays the 'Unidades didácticas' (Didactic Units) section of the SPIKE Prime app. On the left is a navigation sidebar with icons for Inicio, Comenzar, Unidades (highlighted), Construir, Mis proyectos, Ayuda, and Ajustes. The main content area features a grid of four units, each with a 'DESCARGA' button and a 'Primer ciclo de ESO' label. The units are: 1. 'Escuadrón de los inventos' (STEM, Ingeniería) with a green background and a question about inventing or fixing things. 2. 'Arranque de un negocio' (STEM, Matemática, Programación) with a pink background and a question about starting a business. 3. A unit with a blue background and a person at a computer. 4. A unit with a yellow background and a group of people. A link 'VER PLANES DE LECCIÓN Y APOYO DOCENTE' is visible at the top right of the main area.

Unidades didácticas [VER PLANES DE LECCIÓN Y APOYO DOCENTE](#)

DESCARGA Primer ciclo de ESO

STEM, Ingeniería Bloques de palabras

Escuadrón de los inventos

¿Inventas cosas? ¿Arreglas cosas? ¿Siempre ayudando a la gente con tus ideas? Entonces, ¿podrías ser miembro de élite del...

DESCARGA Primer ciclo de ESO

STEM, Matemática, Programación Bloques de palabras

Arranque de un negocio

Se te ha ocurrido una idea buenisísima y quieres que todo el mundo la conozca. El espíritu emprendedor puede surgir en cualquier...

DESCARGA Primer ciclo de ESO

DESCARGA Primer ciclo de ESO

Siguiendo los pasos de las lecciones nos enseñan a construir nuestro lego y a programarlo.

The image shows the LEGO Education SPIKE 3.4.3 software interface. The main window displays a programming sequence for a "Carrera de saltadores" (leaper race). The sequence starts with a yellow "cuando se inicia el programa" (when program starts) block, followed by a pink "fija E4 como motores en movimiento" (set E4 as moving motors) block, and another pink "fija velocidad de movimiento al 50 %" (set movement speed to 50%) block. This is followed by three purple "encender" (turn on) blocks, each set to "durante 1 segundos" (for 1 second). The sequence ends with a pink "mover" (move) block set to "durante 10 segundos" (for 10 seconds). A yellow sticky note is attached to the "mover" block with the text: "Ajustalo para cambiar la distancia que recorrerá tu saltador." (Adjust it to change the distance your leaper will travel).

On the right side, there is a video player overlay titled "Lección" (Lesson). The video is labeled "01/07" and shows a blue screen. The progress bar indicates "0:00 / 0:44". Below the video player, the text "¡Carrera de saltadores!" is displayed. At the bottom right of the video player, there is a blue circular arrow icon.

The bottom of the software interface shows a toolbar with various icons for navigation and execution, including a yellow play button.

Si preferís programar por vuestra cuenta podemos crear un nuevo proyecto.

The screenshot shows the SPIKE Prime software interface. On the left is a navigation sidebar with icons for Inicio, Comenzar, Unidades, Construir, Mis proyectos, Ayuda, and Ajustes. The main area features a blue header for the 'Primeros pasos con SPIKE™ Prime' tutorial, which includes a 'COMENZAR' button. Below this, a section titled 'Directorio reciente' contains a 'Nuevo proyecto' button with a plus sign, highlighted by a purple box, and two other project thumbnails. At the bottom, there are two cards: 'Unidades didácticas' and 'Instrucciones de construcción'.

Primeros pasos con SPIKE™ Prime

¡Vamos a aprender a usar SPIKE Prime en seis sencillos pasos!

COMENZAR

Directorio reciente

Nuevo proyecto

Crear un proyecto nuevo

Carrera de salt...
hace unos segun...

La matriz de lu...
hace 6 minutos

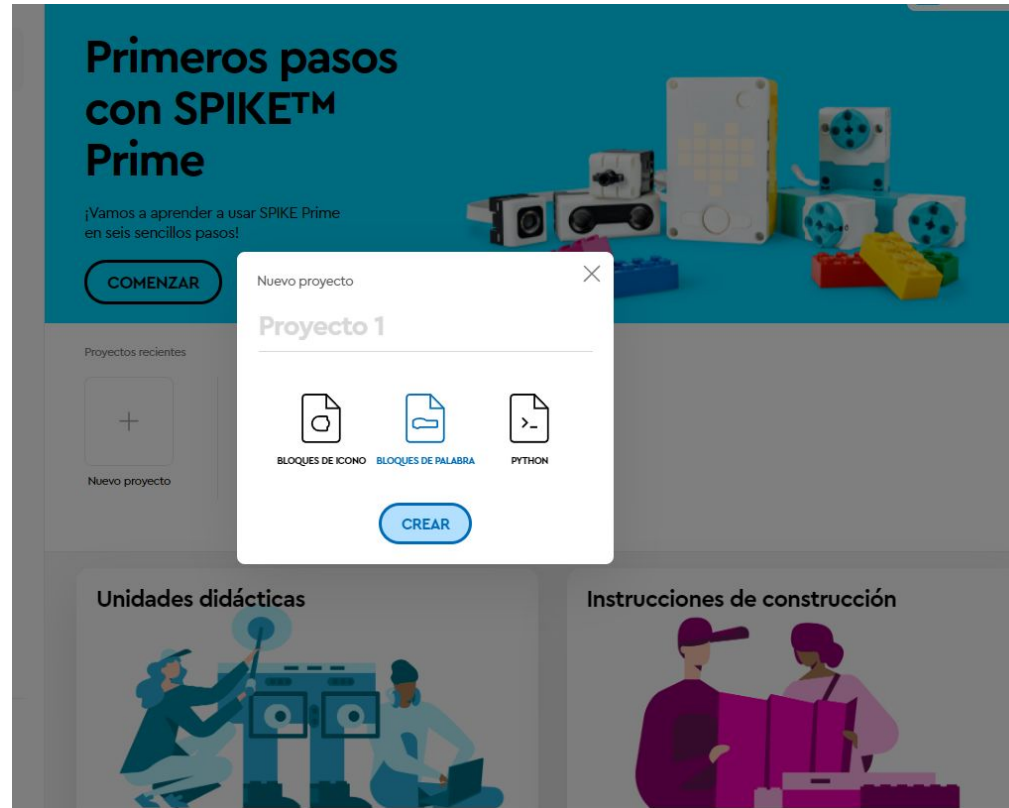
Unidades didácticas

Las lecciones SPIKE Prime se agrupan en unidades temáticas para

Instrucciones de construcción

Una colección de instrucciones de construcción de todos los

Ponemos nombre a nuestro proyecto y seleccionamos bloques de palabra



Primeros pasos con SPIKE™ Prime

¡Vamos a aprender a usar SPIKE Prime en seis sencillos pasos!

COMENZAR

Proyectos recientes

Nuevo proyecto

Nuevo proyecto

Proyecto 1

BLOQUES DE ICONO BLOQUES DE PALABRA PYTHON

CREAR


Unidades didácticas

Instrucciones de construcción

The image shows a software interface for SPIKE Prime. At the top, there's a blue banner with the text 'Primeros pasos con SPIKE™ Prime' and a sub-header '¡Vamos a aprender a usar SPIKE Prime en seis sencillos pasos!'. Below this is a 'COMENZAR' button. A 'Nuevo proyecto' dialog box is open in the center, titled 'Proyecto 1'. It offers three options: 'BLOQUES DE ICONO', 'BLOQUES DE PALABRA', and 'PYTHON'. A 'CREAR' button is at the bottom of the dialog. In the background, there are sections for 'Unidades didácticas' and 'Instrucciones de construcción' with corresponding illustrations.

PROPUESTAS

1. ¡Ayuda! (Primera actividad del escuadrón de inventos)



01

¡Ayuda!

Definir un problema


> MÁS

COMENZAR

🕒 30-45 min.

Detailed description: This block contains a card for the first activity. On the left is a square image showing a hand placing a blue Technic piece onto a yellow and white robot-like assembly. To the right of the image is the text '01 ¡Ayuda!' in bold, followed by 'Definir un problema' and '> MÁS'. Further right is a blue button with the word 'COMENZAR' and a clock icon with the text '30-45 min.' below it.

2. Haz tu pedido (Primera actividad de arranque de un negocio)



01

Haz tu pedido

Descomponer un problema

> MÁS


COMENZAR

🕒 30-45 min.

Detailed description: This block contains a card for the second activity. On the left is a square image showing a hand pointing at a more complex LEGO Technic assembly with various colored pieces. To the right of the image is the text '01 Haz tu pedido' in bold, followed by 'Descomponer un problema' and '> MÁS'. Further right is a blue button with the word 'COMENZAR' and a clock icon with the text '30-45 min.' below it.

PROPUESTAS

3. Break Dance (Primera actividad de invenciones caseras)



01 **Break dance**

Hacer operaciones relacionadas con el tiempo

> MÁS

COMENZAR

🕒 30-45 min.

4. Campamento de entrenamiento 1: ¡A conducir! (Primera actividad de listo para competir)



01 **Campamento de entrenamiento 1:
¡A conducir!**

Controlar movimientos con el sensor giroscópico

> MÁS

COMENZAR

🕒 30-45 min.