MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

LABORATORIOS de Innovación Educativa

Manual de uso
KIT DE INGENIERIA STEM



CISP
COMITATO INTERNAZIONALE PER LO SVILUPPO DEI POPOLI

Computadores para Educar



LABORATORIOS
de Innovación Educativa

Contenido

Presentación 3

Componentes estructurales del Kit de Ingeniería STEM 4
Building:bit: Robot de bloques de construcción 4
Bloques de construcción con agujeros 4
Bloques de construcción de ruedas 4
Bloques de construcción de eje transversal 5
Bloque de construcción del marco Liftarm 5
Bloques de construcción de pines 5
Bloques de construcción de bujes 5
Bloques de construcción de conector de eje/perno 5
Otros accesorios 6

Fichas Mega Bloques 7
Bloques de construcción 1x1 8
Bloques de construcción 1x2 8
Bloques de construcción 1x4 8
Bloques de construcción 2x2 8
Bloques de construcción carro 8
Bloques de construcción accesorios 8

Tableros ajedrezados 9

Tarjetas electrónicas del Kit de Ingeniería STEM 11

Arduino UNO 11
Programación 12
Conexión 12
Transferir el programa al Arduino 13

Micro: bit 13
Programación 14
Conexión 14
Transferir el programa al micro:bit 14



Building:bit **15**
Programación 15

Gamepad **19**
Componentes 19
Instalación 19
Programación 21

Cómo usar el Kit de Ingeniería STEM en las rutas didácticas .. **25**
Hardware **25**
Arduino 25
Micro:bit 28
Building:bit 31
GamePad 33
Tarjeta de expansión para sensores y actuadores 34
Conexión del micro:bit a las tarjetas de expansión 37
Alimentación del Arduino, el micro:bit y las tarjetas de expansión 38
Conexión de sensores y actuadores 45

Software **62**
Programación en MakeCode – Generalidades 62
Librerías del kit STEM 71
Programación de sensores 79
Programación de actuadores 85
¿Cómo crear y programar variables? 89
¿Cómo crear y programar funciones? 90
¿Cómo programar el micro:bit físico? 94
Programación en Arduino – Generalidades 101
Aspectos básicos de programación en Arduino 104
Cómo crear un nuevo proyecto en Arduino y descargarlo en la tarjeta 110

Instrucciones de mantenimiento **113**
Limpieza de Gamepad 113
Micro:bit 114

Garantías **114**

Sostenibilidad **115**



Presentación

El propósito de Computadores para Educar, al dotar a las instituciones educativas con Laboratorios de Innovación Educativa, es el de apoyarlas en la incorporación del enfoque STEM en el currículo que desarrollan como parte de su misión.

El enfoque STEM consiste en la integración de las Ciencias, las Tecnologías Digitales, la Ingeniería y las Matemáticas en el desarrollo de proyectos que tienen en cuenta el contexto de las comunidades educativas para identificar problemas y proponer soluciones basadas en los conocimientos y técnicas de las áreas mencionadas.

Los estudiantes, de la mano de los docentes, se ven involucrados en el HACER, mediante la construcción de prototipos de las soluciones que diseñan y desarrollan. Con ello no solo adquieren conocimientos teóricos y prácticos de las áreas STEM, sino que también se involucra el desarrollo de habilidades blandas como el trabajo colaborativo, la toma de decisiones, el desarrollo del pensamiento crítico, las habilidades de pensamiento y de búsqueda de información, de comunicación y, en general, lo que se ha denominado Habilidades del Siglo XXI.

Este manual busca dar soporte al uso del Kit de Ingeniería STEM que forma parte del Laboratorio de Innovación Educativa proporcionado a las instituciones focalizadas. Está escrito de forma sencilla, concreta y práctica y así los docentes de las instituciones tengan a la mano las instrucciones necesarias para que las nuevas soluciones tecnológicas puedan ser utilizadas como herramientas útiles en el desarrollo de sus proyectos.

El presente documento contiene un apartado que ilustra los componentes que vienen con el Kit de Ingeniería STEM, las instrucciones de instalación, de uso y de mantenimiento de este, adicionalmente al final del manual se encuentran instrucciones sobre el alcance de la garantía que viene con el kit y su mecanismo de activación.

Esperamos que este manual sea útil en su propósito de que los docentes empleen intensivamente las nuevas tecnologías en beneficio de la educación de los estudiantes.

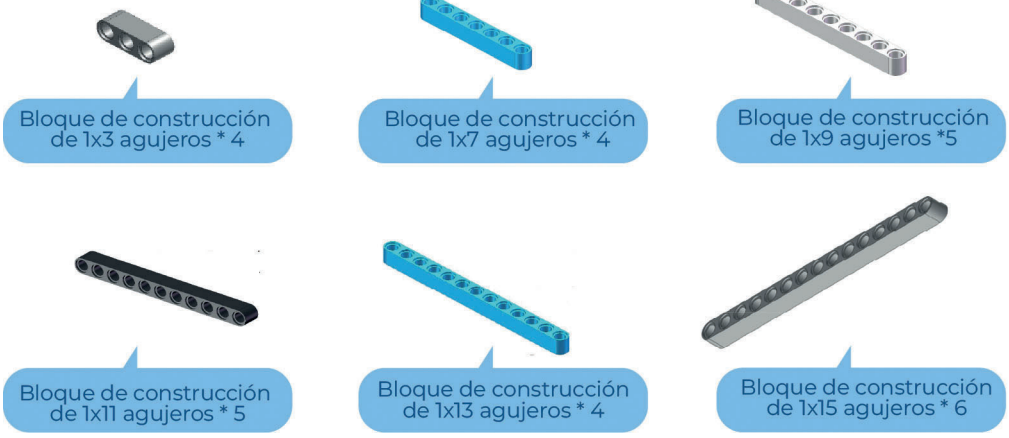


Componentes estructurales del Kit de Ingeniería STEM

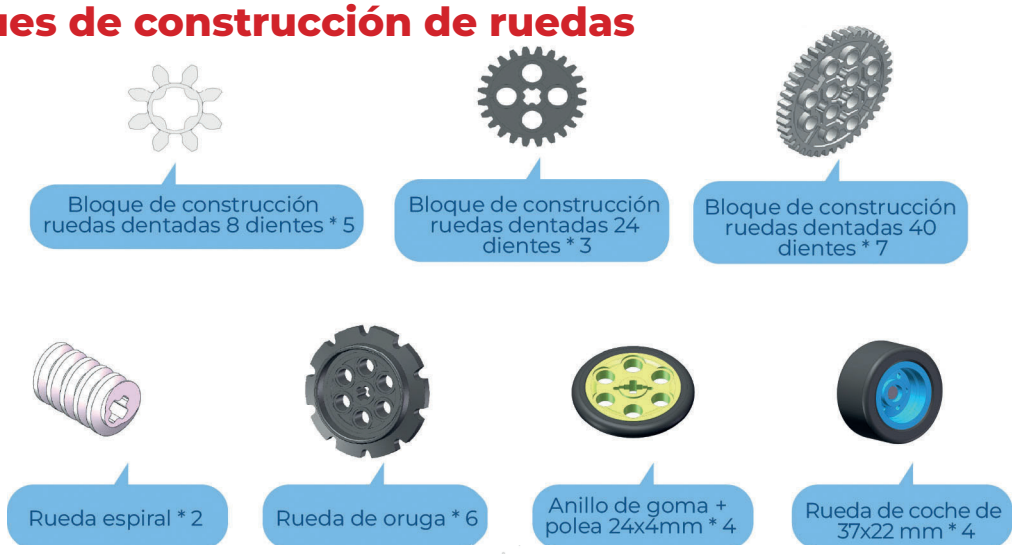
Building:bit. Robot de bloques de construcción

Este kit de bloques de construcción programable incluye un micro:bit, una tarjeta de expansión para micro:bit, una batería, motores, componentes electrónicos adicionales y más de 260 bloques de construcción; se incluye también un manual de montaje de bloques de construcción que proporciona los pasos para montar en 9 estructuras y robots diferentes.

Bloques de construcción con agujeros



Bloques de construcción de ruedas



Bloques de construcción de eje transversal



Bloque de construcción de eje transversal 1x8 * 3



Bloque de construcción de eje transversal 1x12 * 7



Bloque de construcción de eje transversal 1x4 * 3

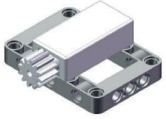
Bloques de construcción del marco Liftarm



Bloque de construcción de marco de viga 5x7 * 5



Bloque de construcción de marco de viga 5x11 * 2



Motor + haz 5x7
Bloque de construcción de marco * 2

Bloques de construcción de pines



Pin de fricción 1x2
Bloque de construcción * 68



1x2 pasador liso
Bloque de construcción * 7

Bloques de construcción de bujes

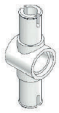


Buje 1x1 Bloque de construcción * 16



1/2 buje Bloque de construcción * 5

Bloques de construcción de conector de eje/perno



Bloque de construcción * 11



Bloque de construcción conector de perno 2 * 7



Bloque de construcción de conector de perno 3x3 * 7



1x2 Bloque de construcción conector eje y perno * 9



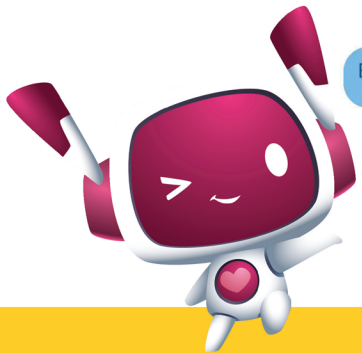
Bloque de construcción de eje 1x3 * 12



Bloque de construcción Conector de eje 1x2 * 3



Bloque de construcción Corte de eje 1x4 * 9



Otros accesorios



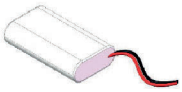
Pieza de oruga de plástico * 68



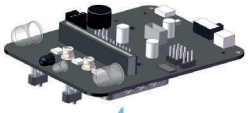
Gancho de elevación * 1



Removedor de bloques de construcción * 1



Batería * 1

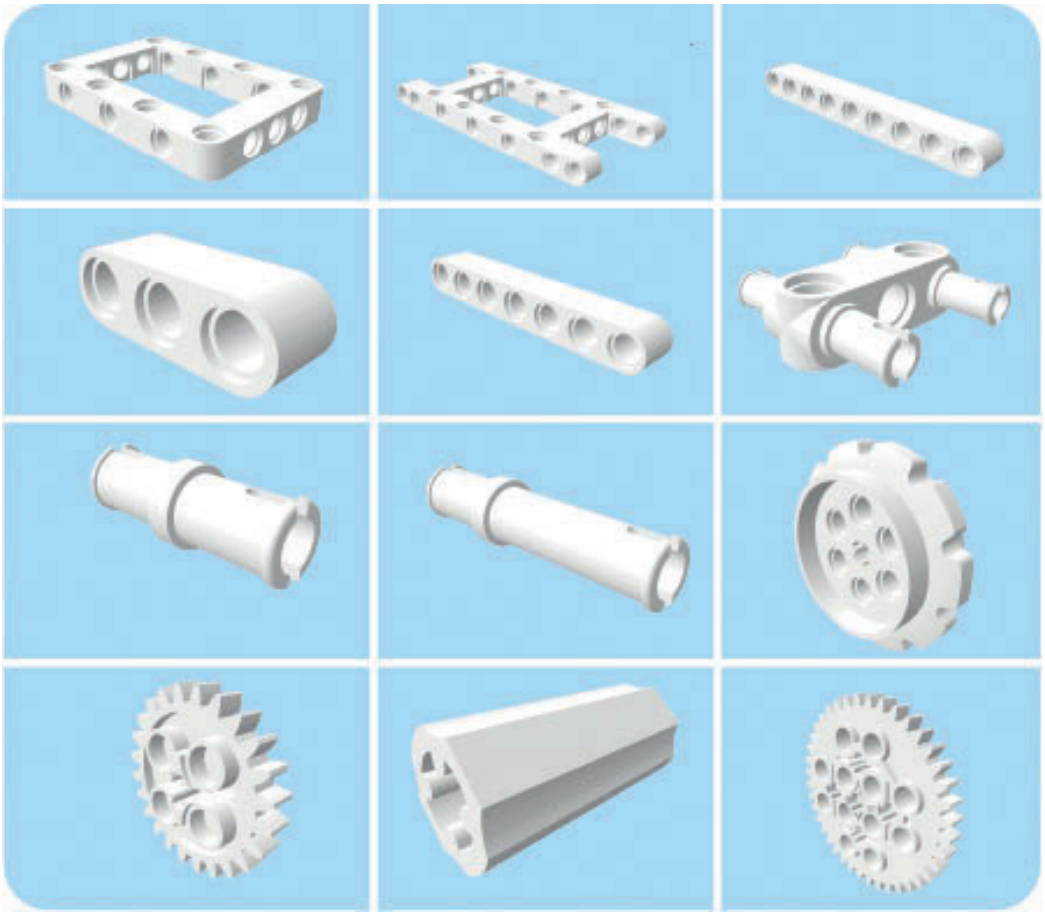


Tarjeta de expansión de micro:bit * 1



Micro:bit * 2

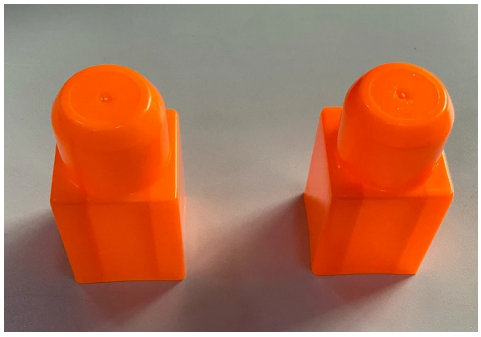
Es importante mencionar que los bloques de construcción más representativos del kit, se encuentran disponibles en formato adaptado para generar una impresión 3D en caso de pérdida o en caso de que sea necesario replicar algunos de los bloques para la realización de distintas prácticas. Estos bloques se encuentran en archivos cuyos formatos ya están parametrizados para su impresión 3D: el formato establecido es en GCode y no es necesario que el docente haga algún tipo de ajuste o edición para iniciar su impresión. Los archivos se encontrarán almacenados en el Gestor de Contenidos para poder ser descargados y utilizados según se necesite. Los bloques que se pueden imprimir en 3D son los siguientes:



Fichas mega bloques

Este kit de bloques de construcción de gran tamaño es para los niños más pequeños y para la realización de construcciones de grandes estructuras. Está compuesto por bloques modulares de diferentes formas y tamaños que se pueden conectar entre sí para formar todo tipo de estructuras. Este kit de fichas mega bloque está compuesto por los siguientes elementos:

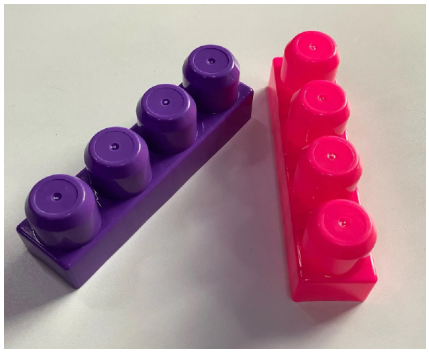
Bloques de construcción 1x1



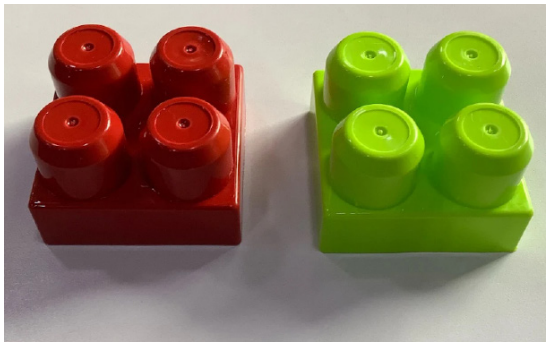
Bloques de construcción 1x2



Bloques de construcción 1x4



Bloques de construcción 2x2



Bloques de construcción carro

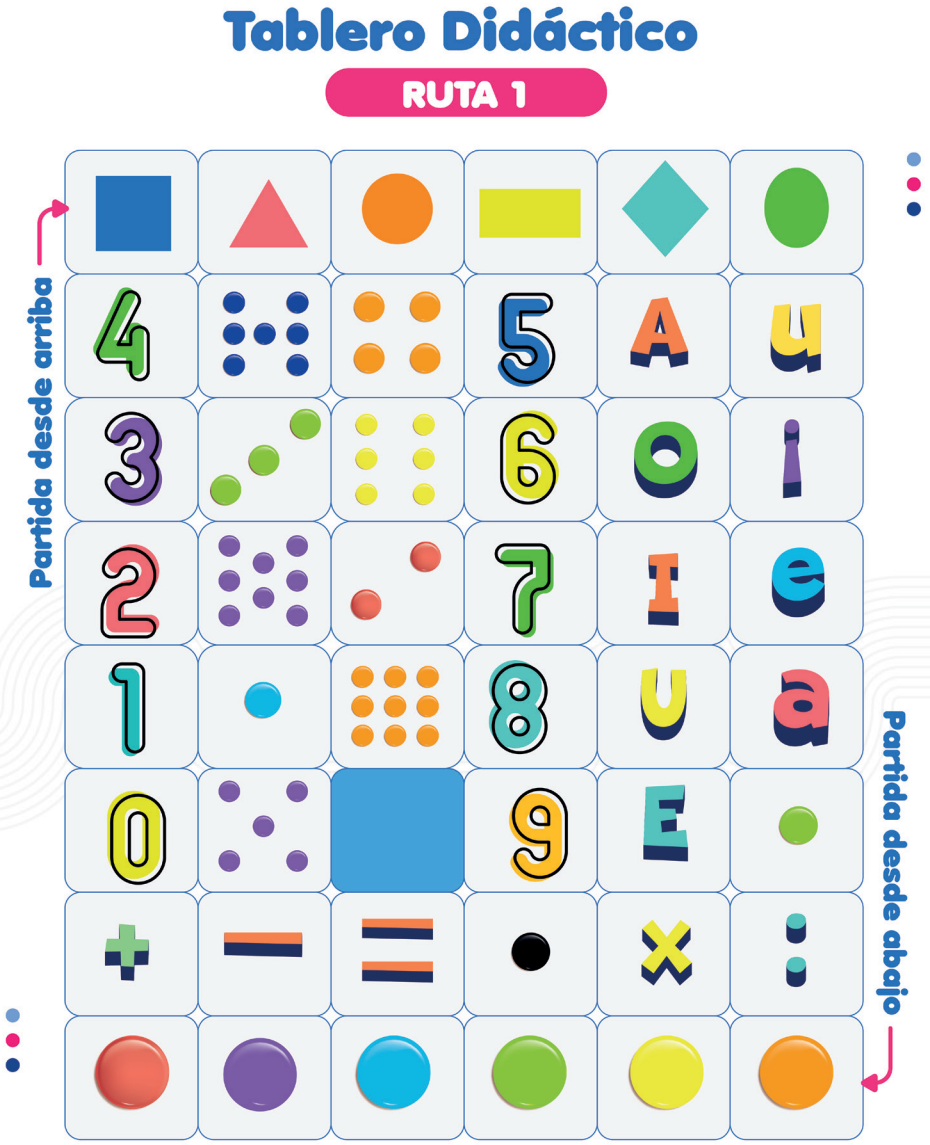


Bloques de construcción - accesorios



Tableros ajedrezados

A continuación, se presentan 2 tableros compuestos por casillas similares a las de un tablero de ajedrez que, permiten realizar prácticas de programación desconectada para los grados primero, segundo y tercero. Los tableros que encontrará en el kit de ingeniería STEM son los siguientes:



LABORATORIOS de Innovación Educativa

Kit de Ingeniería Stem



Tablero Didáctico

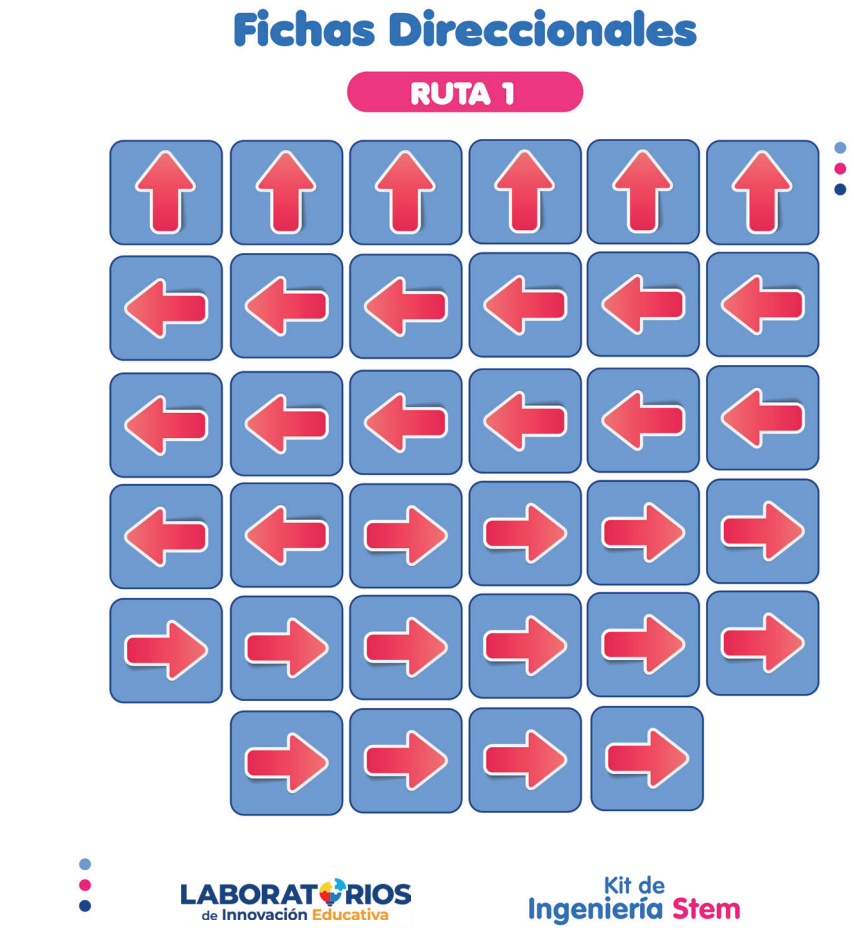
RUTA 1



LABORATORIOS de Innovación Educativa

Kit de Ingeniería Stem

Adicionalmente, se anexa un conjunto de flechas que se podrán cortar para usarlas en las actividades de programación desconectada, con el fin de indicar el camino que se debe tomar dentro de cualquiera de los tableros ajedrezados.



Tarjetas electrónicas del Kit de Ingeniería STEM

Arduino UNO

Es una tarjeta programable de hardware libre que permite crear diferentes microordenadores con una sola placa para darle diferentes usos, reprogramándola con un software libre llamado Arduino. Al hablar de software y hardware libre, se hace referencia a que tanto los componentes y las conexiones físicas de la tarjeta como el software de programación son de acceso libre para todo el mundo, por lo que cualquiera podría replicar tanto la tarjeta como el software para crear sus propios elementos.



El objetivo de la tarjeta Arduino es tener una placa que integre un microcontrolador con sus periféricos de entrada y salida y que se pueda programar directamente sin necesidad de un quemador.

Programación

Un conjunto de instrucciones compone lo que se conoce como algoritmo, el cual se descarga en las tarjetas programables como el Arduino para hacer que realicen las tareas deseadas por el programador. Los algoritmos se escriben en código, un lenguaje que tanto las computadoras como las personas pueden entender.

La tarjeta Arduino se programa con el software libre Arduino, donde se genera un espacio de trabajo para escribir líneas de texto que representan las instrucciones de programación siguiendo las leyes de sintaxis establecidas para el lenguaje de programación Arduino. Las líneas de instrucciones se ejecutan de forma secuencial y siempre se deben finalizar con punto y coma (;).

Necesitará:

Una computadora con el software libre de Arduino instalado y puertos USB para poder programar la tarjeta.

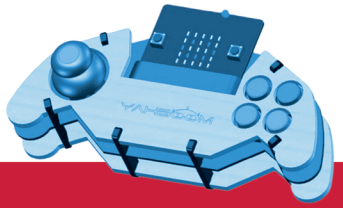
Cuando haya escrito su código, podrá conectarse y transferirlo al Arduino conectándolo al computador, vinculándolo al software, cargándolo y haciendo clic en un botón.

NOTA

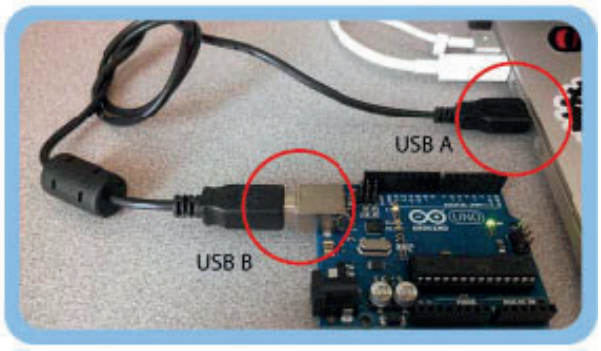
En la sección "Cómo usar el Kit de Ingeniería STEM en las rutas didácticas – Software" se explica detalladamente el proceso de programación del Arduino junto con una descripción de las estructuras básicas de programación de la tarjeta, y los pasos necesarios para transferir el programa realizado del software a la tarjeta física.

Conexión

Conecte el Arduino a la computadora. Necesitará un cable USB tipo B para conectar el Arduino al puerto USB de la computadora.







Transferir el programa al Arduino

Para transferir su programa al Arduino, debe copiarlo en la memoria flash del Arduino.

El Arduino se pausará y el led rojo de la placa parpadeará rápidamente mientras su programa se transfiere. Una vez copiado, su programa comienza a ejecutarse en el Arduino.

La forma de transferir su programa a la tarjeta desde una computadora es: Conecte el Arduino a la computadora y dentro del proyecto de Arduino que desea cargar en la tarjeta, seleccione en la pestaña de herramientas el tipo de placa a programar (Arduino UNO) y el puerto en el que está conectado. Una vez realizados estos pasos, haga clic en “verificar” para comprobar que el algoritmo no tiene errores de sintaxis y luego haga clic en “cargar”. Luego, espere a que se programe la tarjeta y que el centro de notificaciones del Arduino diga “compilado”.

Micro:bit

Es una tarjeta de circuitos de 4x5cm con una serie de 25 ledes y un chip Bluetooth para conexión inalámbrica.

Puede ser programada para mostrar letras, números y otros símbolos y caracteres.

También puede agregarse a otros dispositivos para formar el cerebro de un robot o desarrollar un instrumento musical.



Programación

La tarjeta micro:bit es una pequeña computadora capaz de seguir un conjunto de instrucciones llamado "algoritmo" que le indican qué tareas realizar, cómo, en qué orden, cuándo y durante cuánto tiempo. Estos algoritmos los escribe un programador en una plataforma de programación usando un lenguaje que puede entender, y luego la plataforma convierte esas instrucciones en un código que solo la tarjeta puede entender y así se le descarga la información para lograr un correcto funcionamiento.

Puede programar su micro:bit en los editores online MakeCode de bloques o Python y JavaScript de texto. Al momento de programar la micro:bit puede elegir el que más le convenga, en este caso se usará el de bloques.

Necesitará:

Una computadora con un navegador web y acceso a Internet para programar la tarjeta micro:bit a través de la plataforma de programación MakeCode.

Cuando haya escrito su código, podrá conectarse y transferirlo a la micro:bit.

NOTA

En la sección "Cómo usar el Kit de Ingeniería STEM en las rutas didácticas – Software" se explica detalladamente el proceso de programación del Arduino junto con una descripción de las estructuras básicas de programación de la tarjeta, y los pasos necesarios para transferir el programa realizado del software a la tarjeta física.

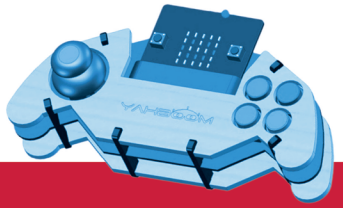
Conexión

Conecte su micro:bit a la computadora. Necesitará un cable micro USB para conectar la micro:bit al puerto USB de la computadora.

Transferir el programa al micro:bit

Para transferir su programa a la micro:bit debe copiarlo en la memoria flash de la micro:bit.

La micro:bit se pausará y el led amarillo en la parte posterior parpadeará mientras su programa se transfiere. Una vez copiado, su programa comienza a ejecutarse en la micro:bit.



Hay dos maneras de transferir su programa desde una computadora:

Arrastra y suelta. Es como copiar un archivo descargado desde su computadora a una memoria USB. Funciona en cualquier computadora.

Flasheo directo. Envía su programa desde el editor de código MakeCode directamente a tu micro:bit. Funciona en cualquier computadora que tenga el navegador Google Chrome actualizado, preferiblemente en su versión más reciente.

Building:bit

Los bloques de construcción y los módulos electrónicos se utilizan para construir un robot móvil inteligente, que se puede programar para evitar obstáculos y rastrear automáticamente.

El kit Building:bit de bloques de construcción se puede ampliar a otros modelos gracias a sus bloques de construcción tipo “Lego” y la versatilidad de programación que ofrece la placa micro:bit. Los niños pueden completar sus propios trabajos con una gran imaginación, conectando diferentes piezas entre sí para crear todo tipo de modelos nuevos.

Programación

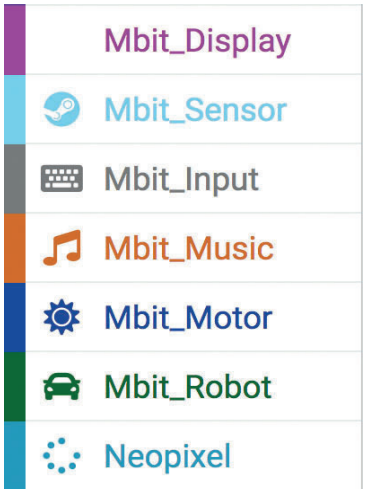
Antes de comenzar a programar este robot móvil, se debe cargar en la plataforma de MakeCode la extensión que facilita su programación y que contiene los bloques necesarios para controlar cualquier movimiento del robot, lo cual se realiza descargando del Gestor de Contenidos el archivo “Librerias-CPE.hex” y cargándolo en el proyecto de programación donde se vaya a utilizar. Este archivo contiene todas las extensiones necesarias para controlar cualquiera de los accesorios del micro:bit.

NOTA

En la sección “Cómo usar el Kit de Ingeniería STEM en las rutas didácticas – Software” se explica detalladamente cómo cargar una librería en la plataforma de programación MakeCode, qué nuevas categorías deben aparecer y cómo usarlas para programar diferentes dispositivos electrónicos. Adicionalmente, se explica detalladamente como programar el micro:bit y sus accesorios, cómo usar los bloques de programación y qué tareas realizan.

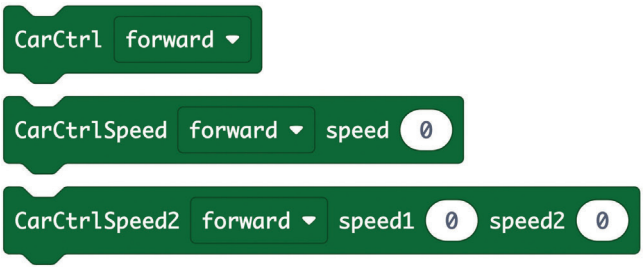


Una vez cargada la extensión aparecerán varias categorías nuevas, se necesitarán las siguientes para programar el robot móvil:



Desplazar

Los bloques necesarios para programar el movimiento del robot hacia delante, atrás y hacia los lados, son los que se muestran a continuación:



Con cualquiera de ellos puede mover el robot, seleccionando la opción deseada a partir del listado de movimientos que ofrece cada bloque:



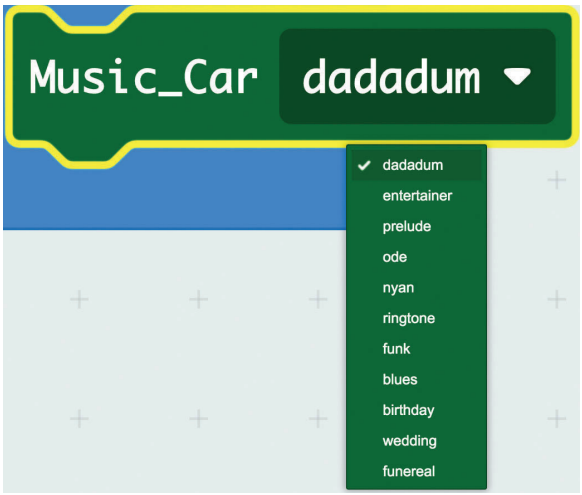
En el espacio designado para “speed” se programa la velocidad del robot móvil entre 0 y 255, siendo 255 la máxima velocidad y 0 sin movimiento.

Sonidos

Para generar un sonido con el buzzer o parlante del robot móvil, se debe usar el siguiente bloque:

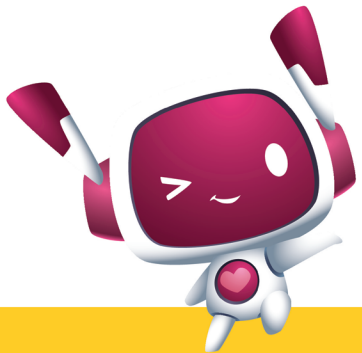


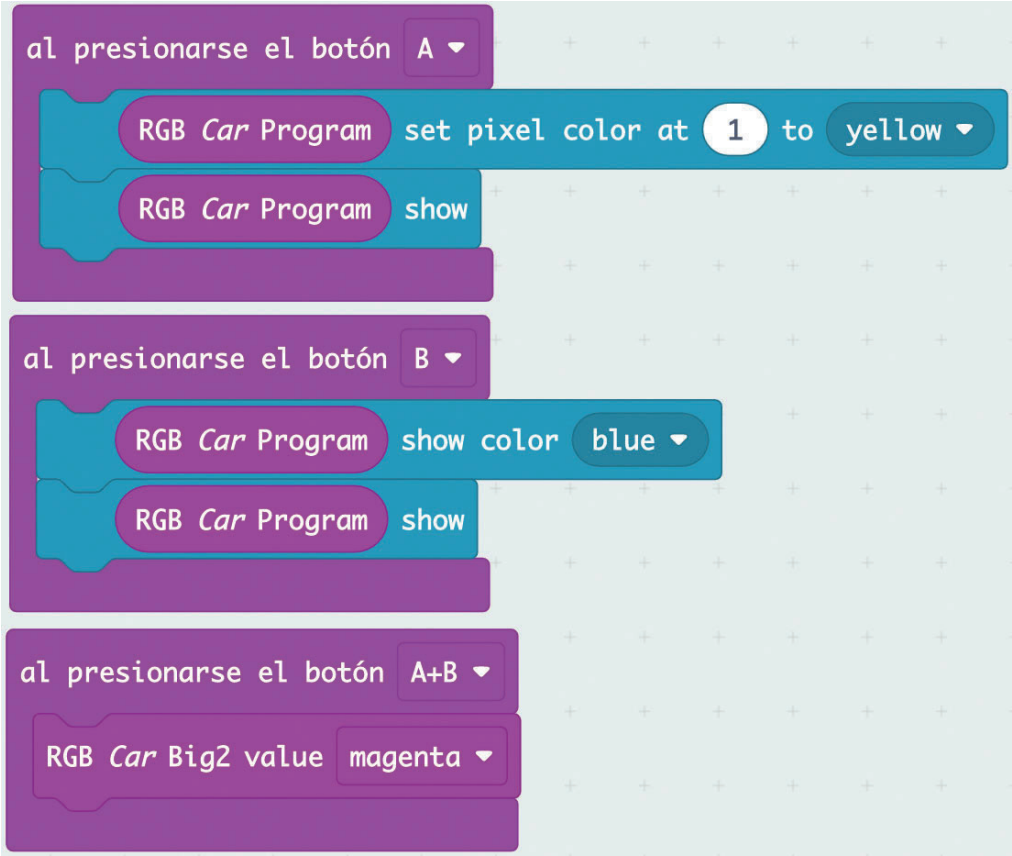
Al desplegar el menú de opciones, se pueden encontrar diferentes tipos de sonidos:



Luces

Para programar el encendido de los ledes del robot y, adicionalmente, definir el color que se desea mostrar en un led específico o en todos al mismo tiempo, es necesario emplear los siguientes bloques:





- Bloques ubicados dentro de “al presionarse el botón A”: Con estos bloques se selecciona un solo led RGB, en este caso el led número 1, y se enciende con un color específico (amarillo para el ejemplo).
- Bloques ubicados dentro de “al presionarse el botón B”: Con estos bloques se encienden todos los ledes al mismo tiempo con el mismo color seleccionado (azul para el ejemplo).
- Bloques ubicados dentro de “al presionarse el botón A+B”: Con estos bloques se encienden los ledes RGB grandes de la parte frontal del vehículo, ambos con el mismo color seleccionado (magenta para el ejemplo).
- El bloque de color fucsia “RGB Car Program” permite programar los ledes ubicados en la parte delantera del robot, encima de la board donde se conecta el micro:bit.

GamePad

Este poderoso dispositivo está diseñado para programar miles de aplicaciones gracias a sus botones completamente programables, y miles de accesorios más que pueden adaptarse al GamePad para llevar a cabo todo tipo de actividades y aplicaciones desde programar videojuegos en el micro:bit hasta controlar robots móviles inalámbricamente, y mucho más.

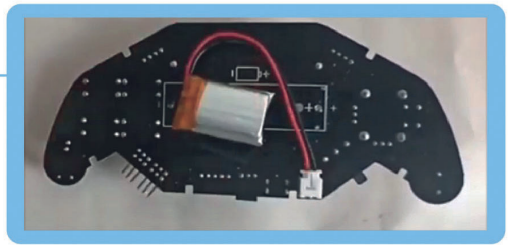
Componentes

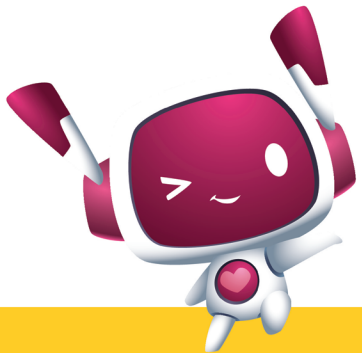
 Game Pad	 Tablero 1	 Tablero 2	 Batería
 Buckle	 Adhesivo		

Instalación

Paso 1

Pegue la batería al GamePad con el adhesivo.





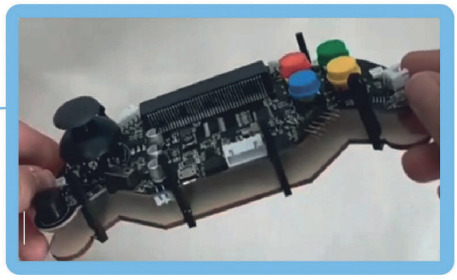


Paso 2

Ubique 6 buckles en los espacios que están en los bordes del tablero 2.

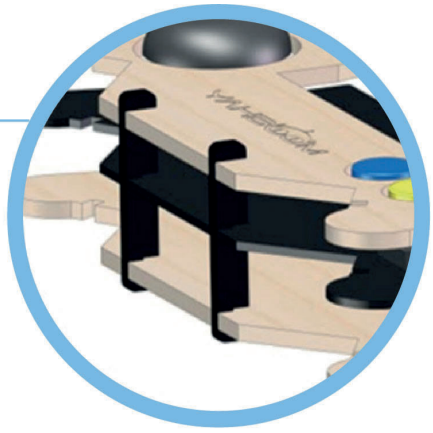
Paso 3

Una el GamePad con el tablero 2, insertándolo en los buckles que ubicó en el paso anterior.



Paso 4

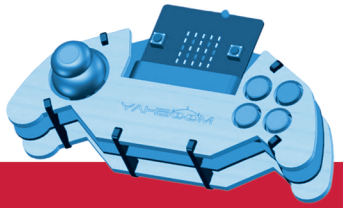
Una el tablero 1 insertándolo en los buckles, de manera que queden las 3 partes juntas.





Paso 5

Inserte el micro:bit en la parte superior del control.



Programación

Antes de comenzar a programar este GamePad, se debe cargar en la plataforma de MakeCode la extensión que facilita su programación y que contiene los bloques necesarios para controlar cualquiera de los elementos del GamePad, lo cual se realiza descargando del Gestor de Contenidos el archivo "Librerias-CPE.hex" y cargándolo en el proyecto de programación donde se vaya a utilizar. Este archivo contiene todas las extensiones necesarias para controlar cualquiera de los accesorios del micro:bit.

NOTA

En la sección "Cómo usar el Kit de Ingeniería STEM en las rutas didácticas – Software" se explica detalladamente cómo cargar una librería en la plataforma de programación MakeCode, qué nuevas categorías deben aparecer y cómo usarlas para programar diferentes dispositivos electrónicos. Adicionalmente, se explica detalladamente como programar el micro:bit y sus accesorios, cómo usar los bloques de programación y qué tareas realizan.

Una vez cargada la extensión aparecerán varias categorías nuevas, se necesitará la siguiente para programar el GamePad:

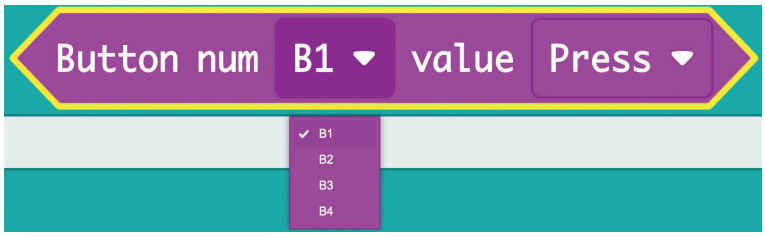


Botones

Los bloques necesarios para programar los botones del GamePad son:



Con cualquiera de ellos se puede programar los botones del GamePad, seleccionando el botón específico a programar a partir del listado que ofrece cada bloque:



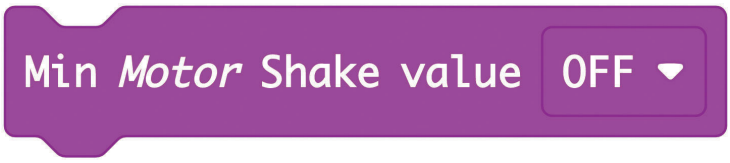


En la siguiente imagen, se muestran los nombres de cada uno de los botones del GamePad para su programación:



Motor de vibración

Para activar y desactivar el motor de vibración que tiene incluido el GamePad se debe usar el siguiente bloque:



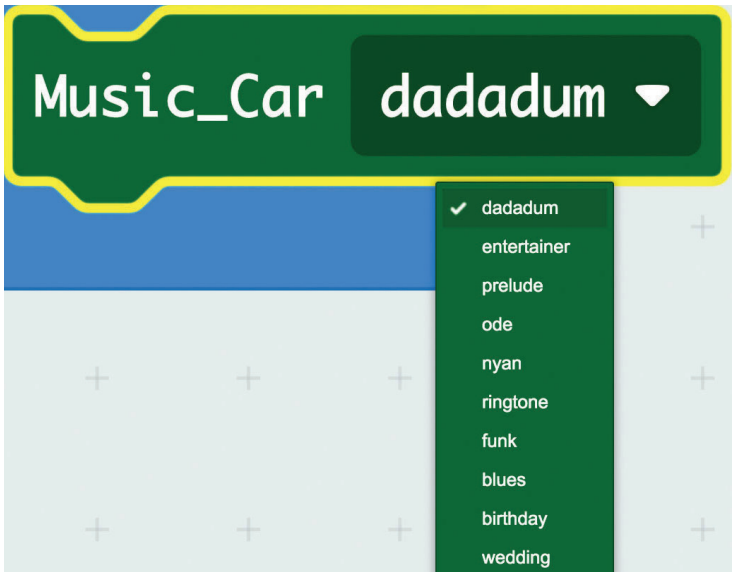
- ON: El motor de vibración se enciende.
- OFF: El motor de vibración de apaga.

Sonidos

Para generar un sonido con el buzzer o parlante del GamePad, se debe usar el siguiente bloque:

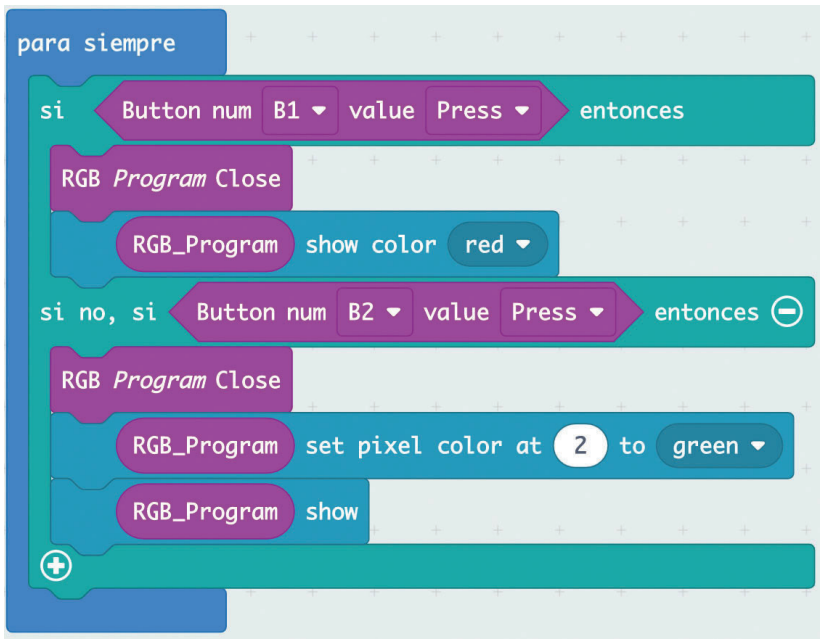


Al desplegar el menú de opciones, se pueden encontrar diferentes tipos de sonidos para escoger:



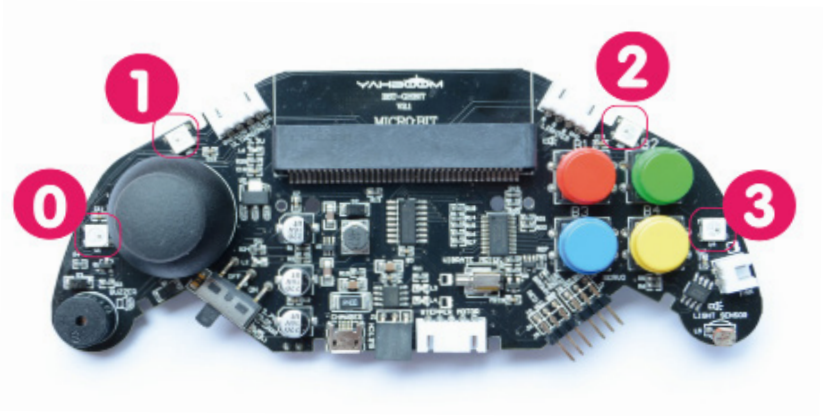
Luces

Para programar el encendido de los ledes del GamePad y, adicionalmente, definir el color que se desea mostrar en un led específico o en todos al mismo tiempo, es necesario emplear los siguientes bloques:



- Con los bloques que están dentro de la instrucción “Button num B1 value Press” se encienden todos los ledes al tiempo con el mismo color seleccionado (rojo para el ejemplo).
- Con los bloques que están dentro de la instrucción “Button num B2 value Press” se selecciona un solo led RGB, en este caso el led número 2, y se enciende con un color específico (verde para el ejemplo).
- El bloque de color magenta “RGB_Program” permite programar los ledes del GamePad.
- El bloque de color magenta “RGB Program Close” reinicia los ledes RGB para permitir la programación de un nuevo color. Debe usarse siempre antes de definir y encender los ledes del GamePad con cualquier color.

En la siguiente imagen se muestra cómo están numerados los ledes RGB del GamePad para que identifiques en qué orden se ubicaron y dónde están dispuestos.



Sensor de luz

Para leer el sensor de luz que tiene incorporado el GamePad se debe emplear el siguiente bloque:



Al desplegar el menú haciendo clic en la flecha se puede seleccionar el estado del sensor de luz, ya sea iluminado (Bright) u oscuro (Dark). Este bloque se puede poner en un condicional para programar acciones en el GamePad según la intensidad de luz ambiente.

Cómo usar el Kit de Ingeniería STEM en las rutas didácticas

Este manual es un documento de apoyo que incluye información sobre la conexión, programación y uso del kit de ingeniería STEM. Con ayuda de este manual puede completar las actividades propuestas en clase y se le pedirá conectar diferentes elementos electrónicos entre sí y programarlos para que trabajen en conjunto.

En el apartado de Hardware encontrará toda la información relacionada con la conexión de los dispositivos electrónicos, y una descripción de los elementos integrados que componen las tarjetas programables y de expansión que se le han entregado.

En el apartado de Software encontrará toda la información relacionada con la programación de cada uno de los elementos del kit, y cómo descargar sus proyectos en el micro:bit para usarlos en sus aplicaciones.

Hardware

A continuación, encontrará toda la información relacionada con la conexión e integración de los diferentes elementos electrónicos del kit de ingeniería STEM, además de una guía de apoyo sobre las partes que componen cada tarjeta electrónica que se le ha entregado, y los diferentes métodos de conexión de cada elemento con las tarjetas electrónicas, dependiendo de los requerimientos de la actividad que se vaya a realizar.

Arduino

Arduino es una plataforma de programación basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

La placa electrónica Arduino es una placa de circuito impreso o PCB por sus siglas en inglés (Printed Circuit Board) que implementa un determinado diseño de circuitería interna para que de esta forma el usuario final no se preocupe por las conexiones eléctricas que necesita el microcontrolador para funcionar,



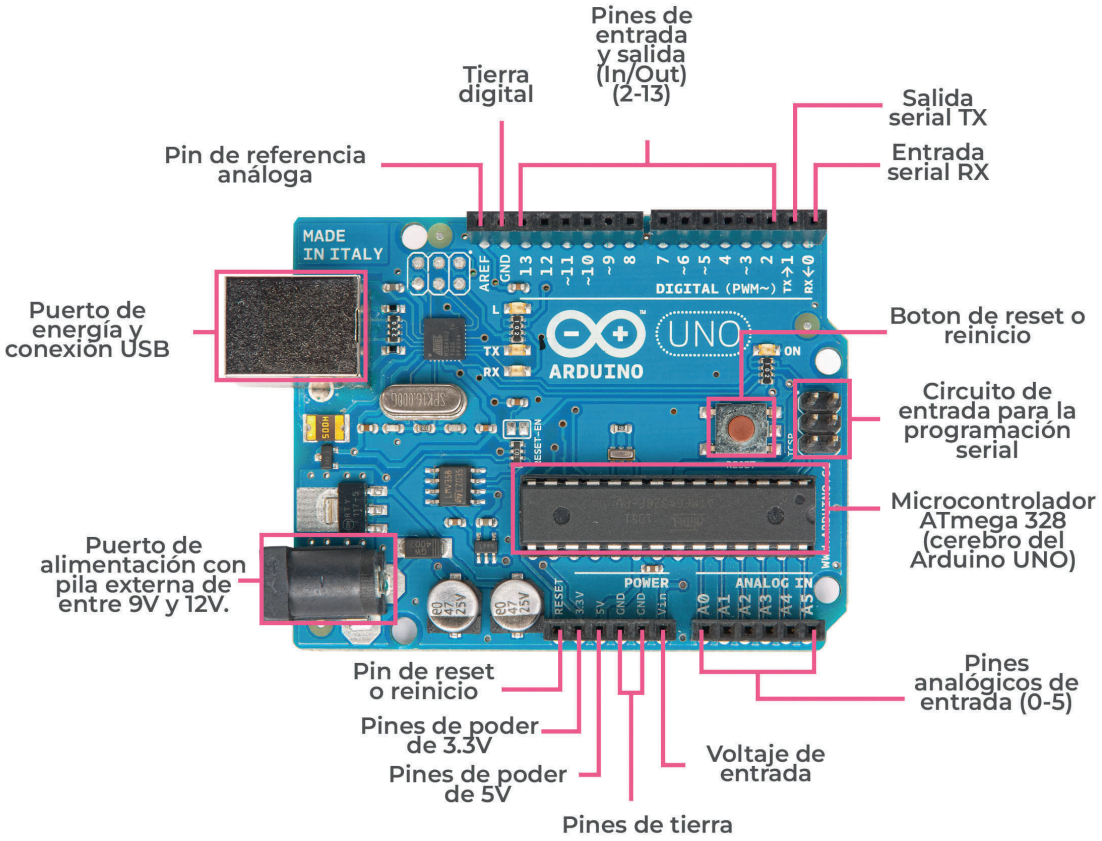
y puede empezar directamente a desarrollar las diferentes aplicaciones electrónicas que requiera .

Este dispositivo es el cerebro y centro de mando de las aplicaciones que se realizarán en clase, principalmente en los grados decimo y once, y se puede programar varias veces con el fin de usarlo en más de una aplicación. Con esta tarjeta será posible programar y controlar todo tipo de dispositivos electrónicos y construir increíbles proyectos STEM.

Partes del Arduino UNO

El Arduino UNO cuenta con un grupo de elementos electrónicos integrados en su placa que permiten su correcto funcionamiento y su uso en diferentes aplicaciones tecnológicas.

Las principales partes del Arduino UNO son:



[1] Extraído de: https://arduino.cl/que-es-arduino/

Los pines de poder y de entrada/salida serán los más usados en las actividades. Cada uno de ellos realiza las siguientes tareas:

- **Pines de poder de 3.3V o 5V, y pines de tierra:** Permiten alimentar dispositivos electrónicos suministrando energía de baja potencia perfecta para sensores, ledes, botones, resistencias y servomotores, pero no para motores DC ni paso a paso.
- **Pines de entrada/salida digital:** Se usan para controlar dispositivos electrónicos digitales o para obtener información de ellos. Solo manejan datos digitales y los que tienen la virgulilla (~), estos permiten controlar servomotores.
- **Pines analógicos de entrada:** Se usan para controlar dispositivos analógicos y obtener información de ellos. Están diseñados exclusivamente para leer valores analógicos.
- **Puertos de energía:** Con el "puerto de alimentación de pila externa" se puede alimentar la tarjeta Arduino y todos sus componentes sin necesidad de conectarla al computador, lo que permite ponerla en cualquier prototipo o robot móvil que se haya desarrollado, mientras que el puerto de energía y conexión USB permite energizar la tarjeta y todos sus componentes mediante el computador y al mismo tiempo programarla cada vez que sea necesario. Es importante tener en cuenta que solo se debe usar una fuente de alimentación, no se deben conectar 2 o más fuentes simultáneamente ya que esto puede dañar la tarjeta.

NOTA

En la sección "Cómo usar el Kit de Ingeniería STEM en las rutas didácticas - Hardware - Conexión de sensores y actuadores" se muestran ejemplos sobre cómo conectar dispositivos electrónicos en esta tarjeta.



Micro:bit

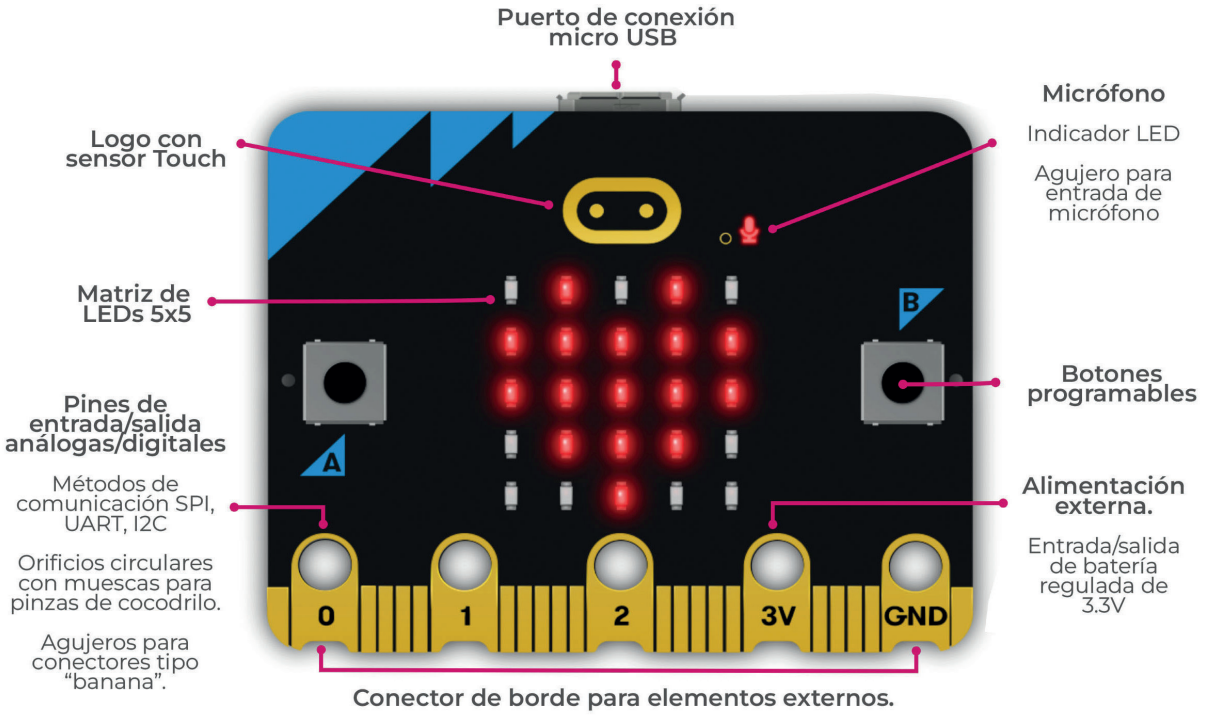
El micro:bit es una tarjeta programable de 4x5cm, donde se cargarán los algoritmos realizados en clase para cada proyecto propuesto. Este dispositivo es el cerebro y centro de mando de todas las aplicaciones que se realizarán en clase, y se puede programar varias veces con el fin de usarlo en más de una aplicación.

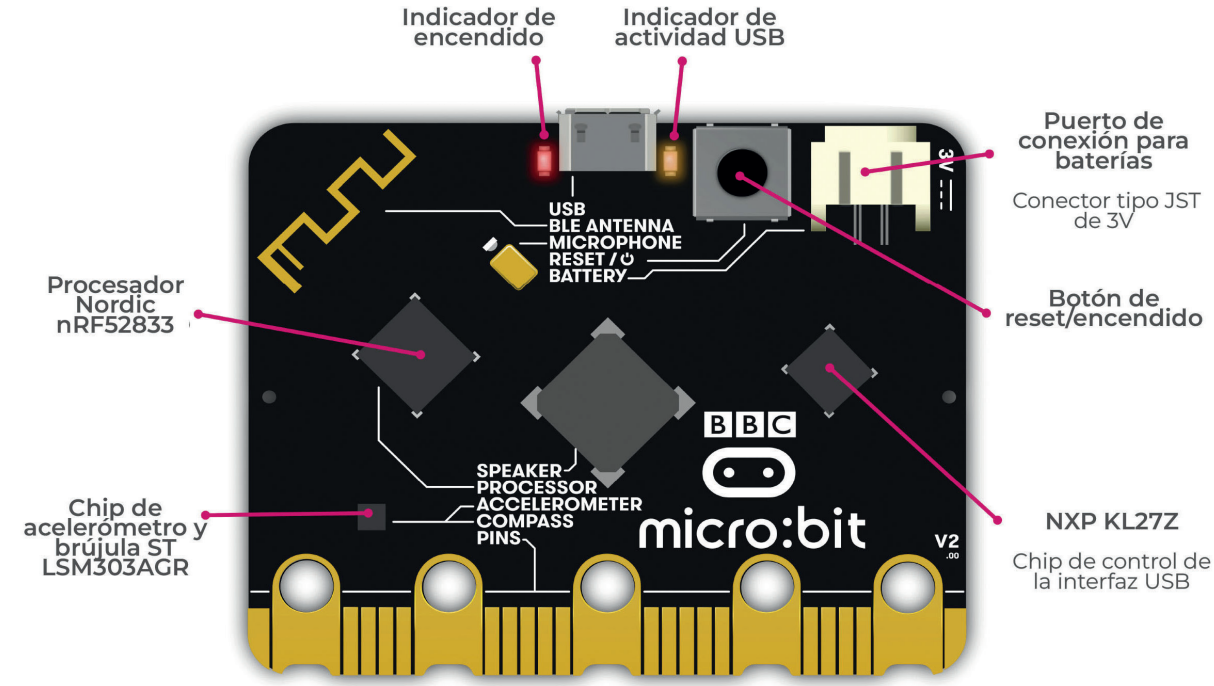
Partes del micro:bit

El micro:bit cuenta con diferentes elementos de entrada y salida para la realización de algoritmos con múltiples aplicaciones prácticas, sin necesidad del uso de elementos externos.

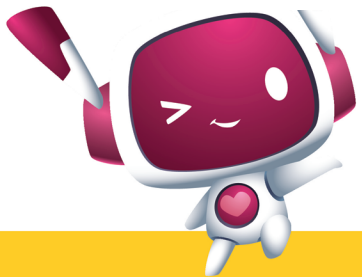
Las principales partes del micro:bit son:

Vista frontal





Vista posterior

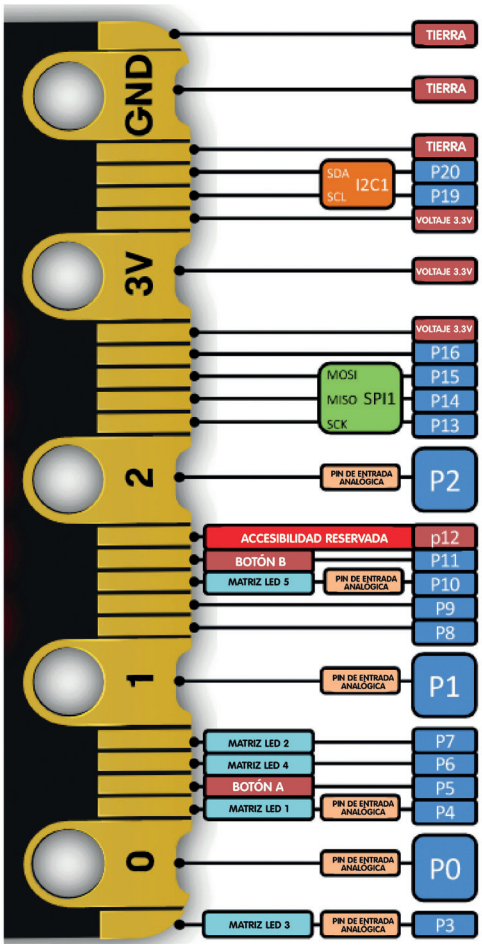


Conector de borde para elementos externos

El puerto de conexión que posee el micro:bit para el control de elementos externos está conformado por más de 20 pines de conexión, donde la mayoría de ellos está conectado con otras partes del hardware del micro:bit (como la matriz de ledes, los botones, los puertos de comunicación, entre otros).

Esta relación entre los pines de entrada/salida analógicos/digitales con otras partes del hardware del micro:bit, lleva a la necesidad de revisar con cuidado qué pines se pueden programar sin problemas y cuáles presentarán interferencias con algunas de las tareas internas del micro:bit.

En la siguiente imagen se puede observar el nombre de cada pin y la parte del hardware del micro:bit con la que está conectado:



Esto significa que:

- Si se va a programar un pin que esté conectado con la matriz led, se debe agregar al algoritmo del proyecto (en el bloque "al iniciar") la instrucción de "activar LEDs falso" que se encuentra en la categoría "LED" de MakeCode, y evitar programar los ledes del micro:bit en el algoritmo, ya que esto puede ocasionar fallas en el control de los dispositivos externos debido a interferencias entre los pines y la matriz led.
- Si se va a programar un pin que esté conectado con el botón A o B, se debe evitar programar los botones del micro:bit, ya que esto puede ocasionar fallas en el control de los dispositivos externos debido a la interferencia entre los pines y los botones.
- El pin P12 no es programable.
- Los pines P20 y P19 están restringidos únicamente al uso de la comunicación I2C.
- Los pines P13, P14 y P15 están restringidos únicamente al uso de la comunicación SPI.
- Los pines P0 a P4 y P10 son los únicos que funcionan como entradas analógicas.

Building:bit

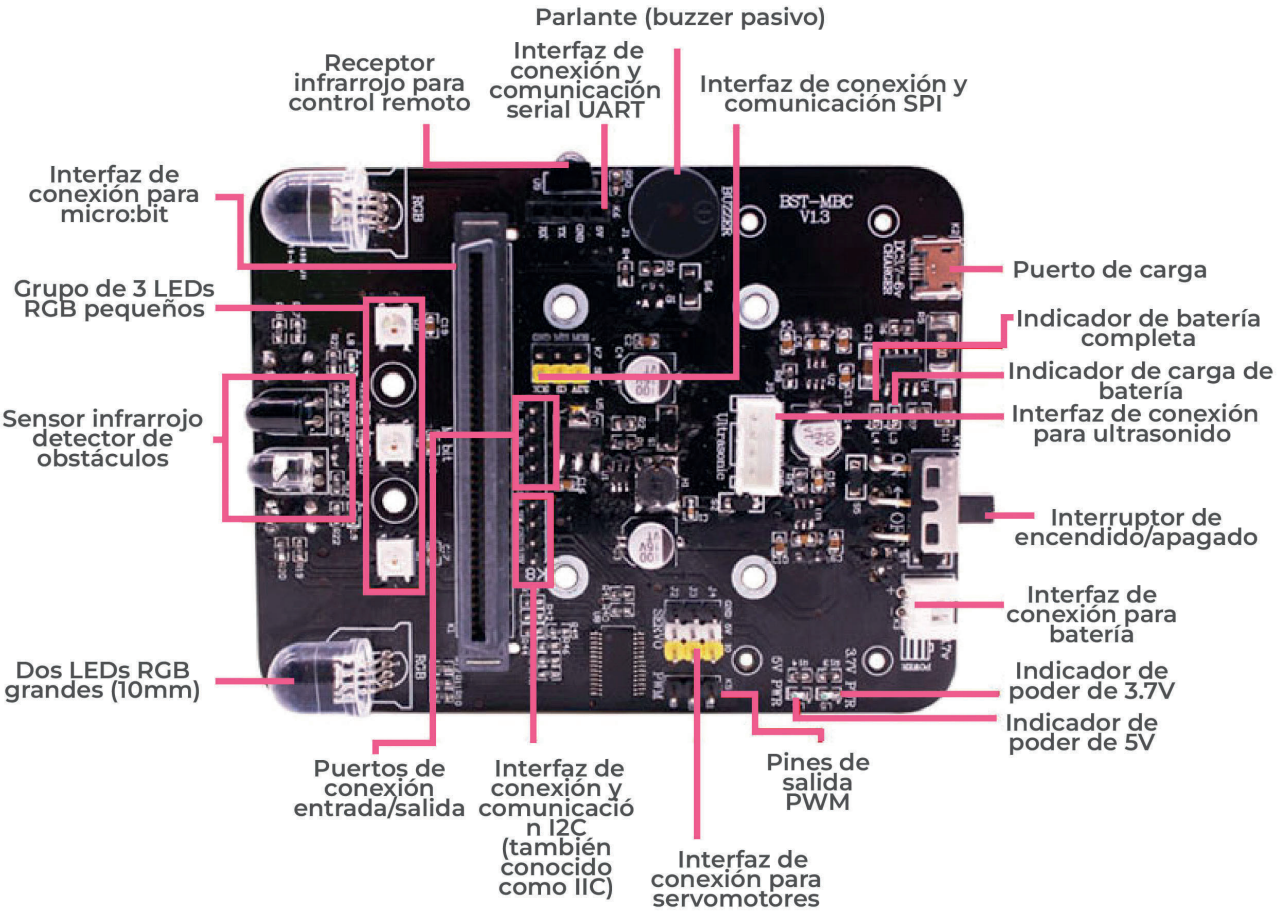
Esta es una tarjeta de expansión donde se puede conectar el micro:bit para programar y controlar estructuras robóticas como las que se proponen en el manual del Building:bit y las que se plantean en clase. Adicionalmente, le permite al micro:bit controlar motores, sensores, ledes RGB, parlantes, entre otros dispositivos electrónicos gracias a que convierte el conector de borde del micro:bit en pines de fácil acceso para cables.

Partes de la tarjeta de expansión del Building:bit

El kit de robótica Building:bit cuenta con una tarjeta de expansión que permite convertir el conector de borde del micro:bit en pines de fácil acceso para conectar elementos externos mediante cables y, al mismo tiempo, controlar los distintos elementos electrónicos que posee esta tarjeta de expansión integrados en su placa, como buzzer, motores, sensores, ledes RGB, receptores infrarrojos, entre otros.

Las partes de esta tarjeta se muestran a continuación:





En los siguientes puertos se pueden conectar elementos externos para controlarlos con el micro:bit:

- En los puertos de conexión entrada/salida se pueden conectar sensores, actuadores, interruptores, y otros dispositivos electrónicos.
- En los puertos de conexión para servomotores se pueden conectar servomotores, o usar sus pines de voltaje y tierra (rojo y negro respectivamente) para alimentar dispositivos electrónicos externos que se hayan conectado en los puertos de conexión entrada/salida.

En la parte inferior de la tarjeta (no visible en la imagen) se pueden conectar motores DC para controlar su dirección y velocidad de giro (ir a “conexión de sensores y actuadores” para observar el puerto y el modo de conexión).

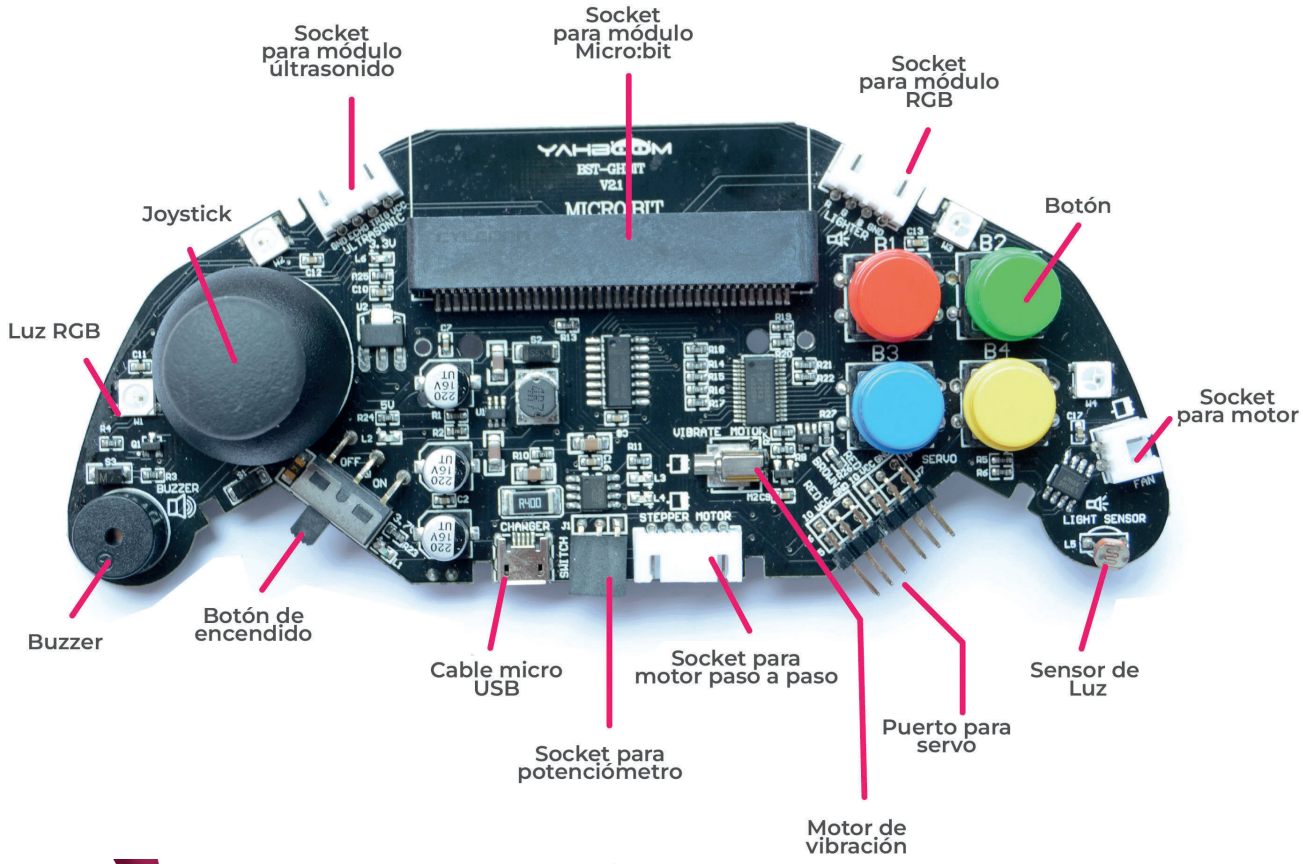
GamePad

Esta es una tarjeta de expansión donde se puede conectar el micro:bit para programar y controlar sensores y actuadores externos, como los que hacen parte del GamePad y controlar los elementos integrados a esta tarjeta mediante el conector de borde del micro:bit.

Partes de la tarjeta de expansión del GamePad

El GamePad cuenta con una tarjeta de expansión que tiene integrados diferentes elementos electrónicos en su placa como buzzer, motor de vibración, sensor de luz, ledes RGB, entre otros, además de poseer puertos de conexión para controlar y programar elementos externos adicionales como el sensor de ultrasonido, el motor con ventilador, el módulo RGB, entre otros.

Las partes de esta tarjeta se muestran a continuación:

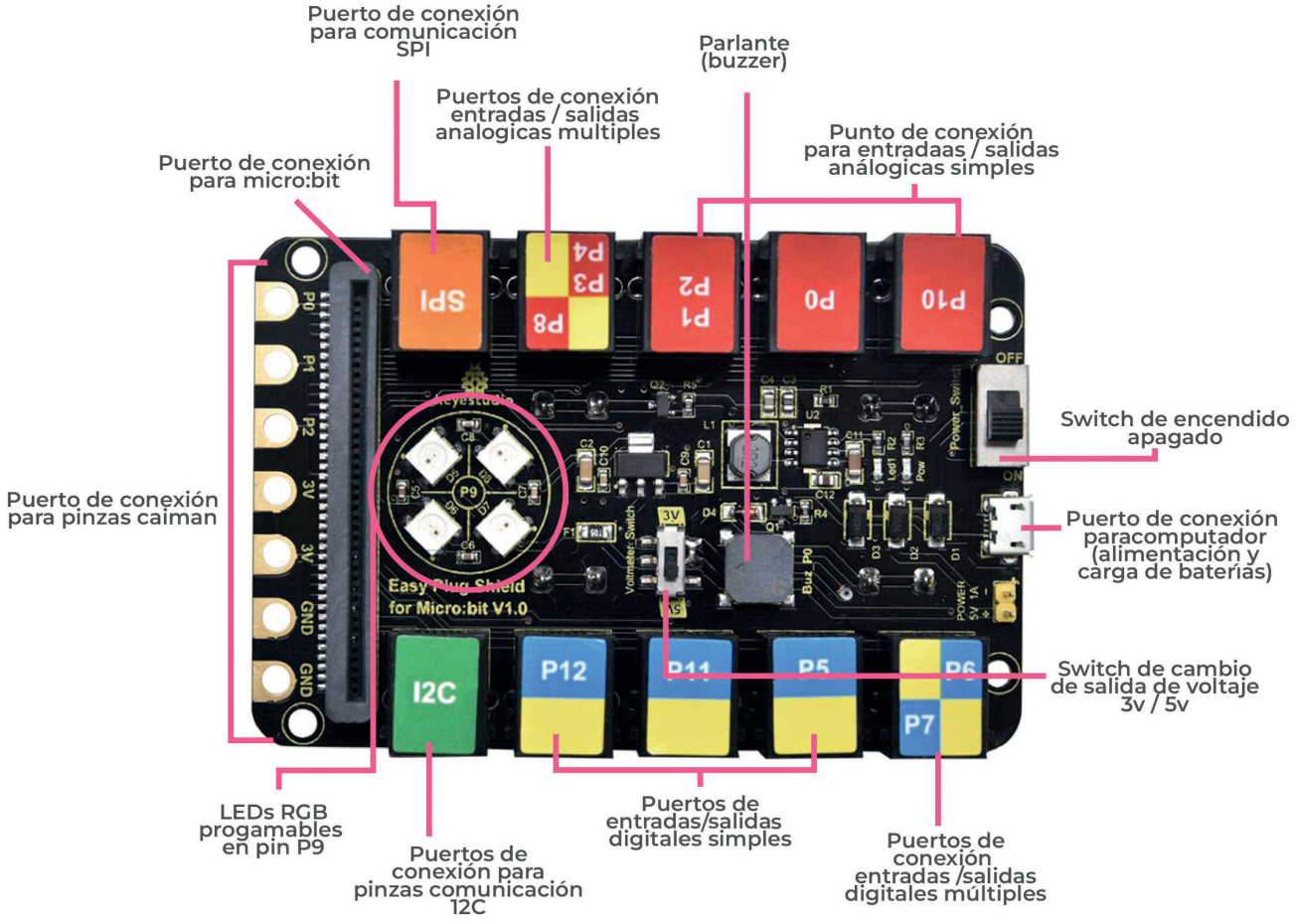


Tarjeta de expansión para sensores y actuadores

Esta es una tarjeta de expansión que convierte el conector de borde del micro:bit en pines de fácil acceso tipo RJ11, en los cuales se puede conectar todo tipo de sensores y actuadores para controlarlos y programarlos con el micro:bit sin riesgo a conectar algún elemento de forma invertida o errónea.

Partes de la tarjeta de expansión para sensores

Las principales partes de esta tarjeta son las que se muestran a continuación:



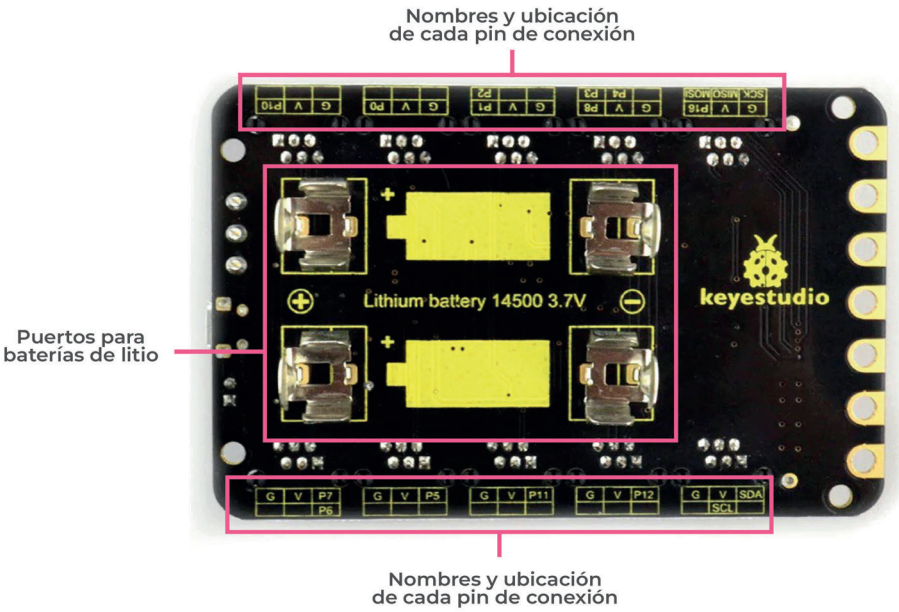
Como se puede observar, cada puerto de conexión RJ11 tiene uno o dos colores que lo caracterizan, los cuales indican el tipo de elemento que se puede conectar en ellos, de tal manera que:

- El color ***azul*** representa puertos de conexión para entrada y salida de dispositivos electrónicos digitales. Sin embargo, también se pueden usar para "escritura analógica" en caso de ser necesario.
- El color ***rojo*** representa puertos de conexión para entrada y salida de dispositivos electrónicos ***analógicos.*** Sin embargo, también se pueden usar para lectura/escritura analógica/digital en caso de ser necesario (excepto P8 que no funciona como entrada analógica).
- El color ***verde*** representa los puertos de conexión para comunicación I2C.
- El color ***naranja*** representa puertos de conexión para comunicación SPI.
- Los puertos con ***4 cuadros de colores*** (2 amarillos y 2 de otro color) representan puertos de conexión para elementos que requieren 2 o más pines de conexión para programarse.

NOTA

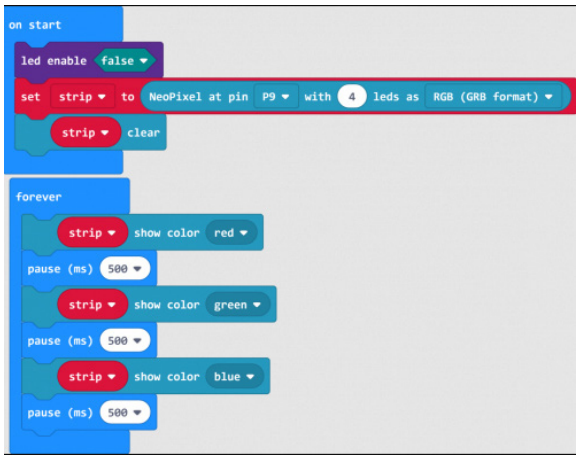
En la sección "Cómo usar el Kit de Ingeniería STEM en las rutas didácticas - Hardware - Conexión de sensores y actuadores" se muestran ejemplos sobre cómo conectar dispositivos electrónicos en esta tarjeta.

En la parte posterior de la tarjeta se puede ver a detalle cada uno de los pines del micro:bit que corresponden a cada puerto RJ11. Esto nos permitirá saber qué pines programar en nuestros proyectos cuando vayamos a usar elementos externos como sensores o actuadores. Adicionalmente, se encuentra marcada la forma correcta de conectar las baterías de litio.

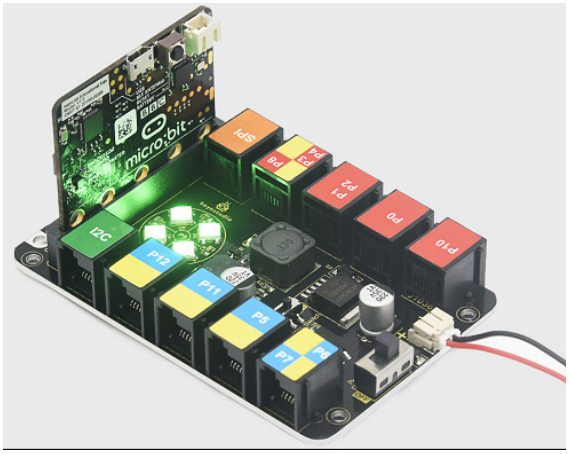


Por otra parte, con el switch de cambio de salida de voltaje 3v/5v podemos definir el voltaje de alimentación para los elementos externos que se conecten a la tarjeta de expansión, ya sea 3V o 5V. Lo más recomendable es usar 5V, aunque casi todos los sensores pueden funcionar también con 3V.

Los otros elementos que están integrados en esta tarjeta son:



- Parlante (buzzer). Está conectado en P0 y se puede programar para generar sonidos.
- Ledes RGB. Están conectados en P9 y se pueden programar para variar sus colores usando los bloques de la categoría “Neopixel” que se encuentran en la librería anexa a los cursos.



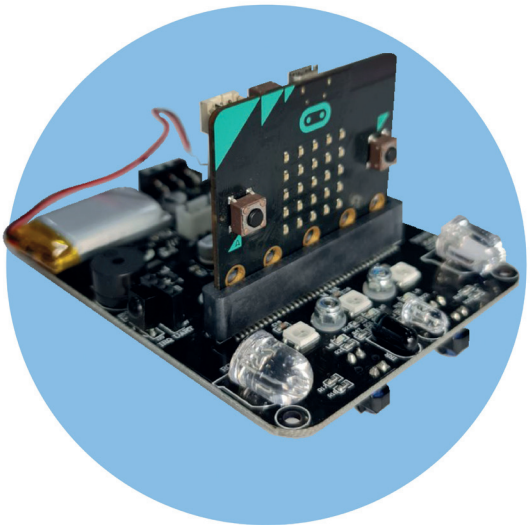
- **Puerto de conexión para pinzas tipo caimán:** En este puerto se pueden conectar cables con pinza tipo caimán para programar elementos externos tanto digitales como analógicos, y que sean tanto de lectura como de escritura. También tiene pines de salida de voltaje y tierra para alimentar circuitos o dispositivos electrónicos externos.

Conexión del micro:bit a las tarjetas de expansión

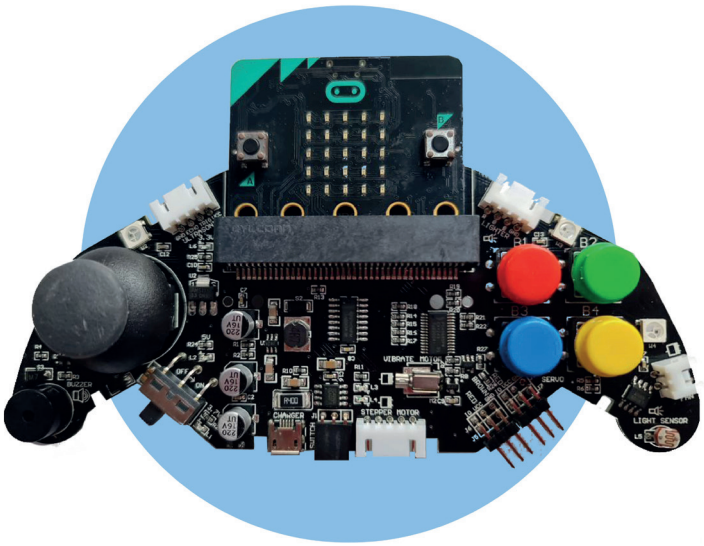
El micro:bit está diseñado para conectarse con otros elementos electrónicos y controlarlos a través de su conector de borde. Para integrar el micro:bit con las tarjetas de expansión de este kit de ingeniería STEM se debe tener en cuenta el sentido de conexión entre ambos elementos, ya que de esto dependerá el correcto funcionamiento de todo el sistema.

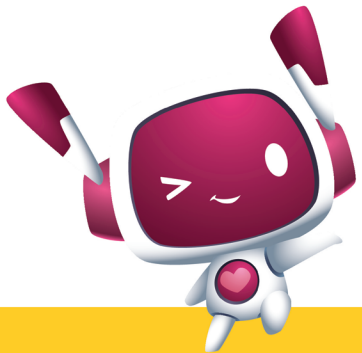
A continuación, se muestra cómo conectar el micro:bit en la tarjeta de expansión del Building:bit, el GamePad y la tarjeta de expansión para sensores y actuadores:

Building:bit

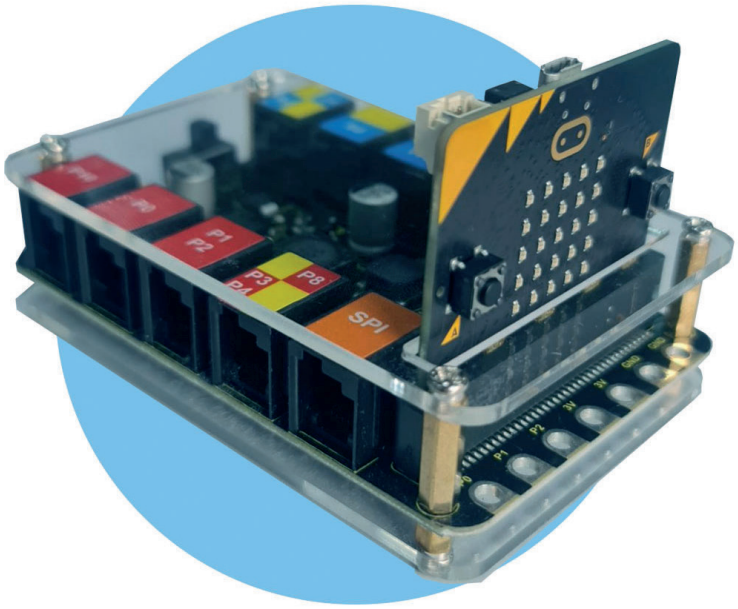


GamePad





Tarjeta de expansión para sensores y actuadores



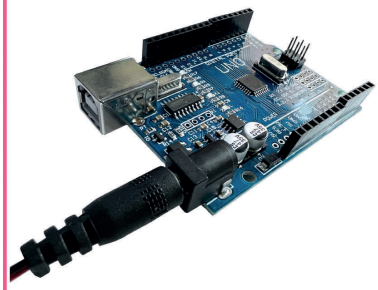
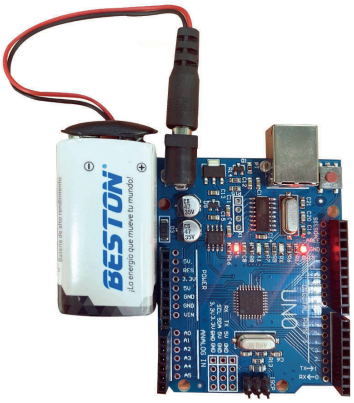
Alimentación del Arduino, el micro:bit y las tarjetas de expansión

Para utilizar cualquiera de las tarjetas electrónicas presentadas en este manual, es necesario alimentarlas con una fuente de energía que sea suficiente para activar cada una de las partes que la componen y los elementos adicionales que se conecten en cualquiera de ellas. Dependiendo de la tarjeta que se seleccione, debe usar un tipo diferente de alimentación y conexión, tal y como se muestra a continuación.

Arduino

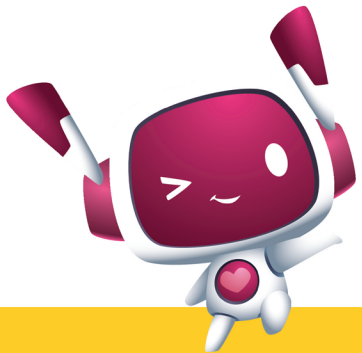
Esta tarjeta la puede alimentar de tres maneras distintas:

a. Pila de 9V. Tome la pila de 9V que viene en el kit, colóquele el adaptador para pilas de 9V y conéctelo en los pines de la batería y en el Arduino como se muestra a continuación, de esta manera el Arduino se encenderá automáticamente y comenzará a ejecutar el programa que le hayas descargado:

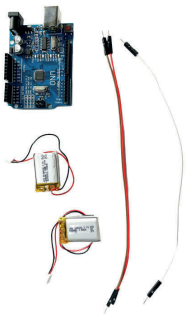
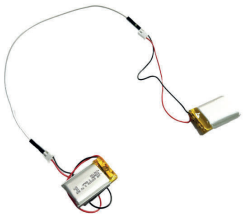
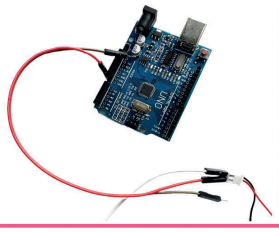
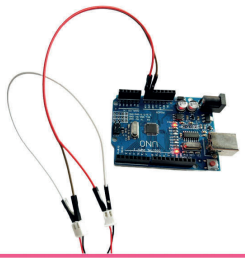
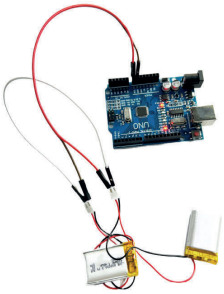
Necesitará:	Conecte el adaptador en la batería:	Conecte el adaptador al Arduino:	Resultado:
			

b. Cable USB. Conecte el puerto de energía y conexión USB del Arduino al cable USB, y conecte el otro extremo del cable USB en el computador como se muestra en la imagen. De esta manera el Arduino se encenderá automáticamente y comenzará a ejecutar el programa que le hayas descargado, y también te permitirá descargar un nuevo programa en cualquier momento.

Necesitará:	Conecte el cable al Arduino:	Conecte el cable al computador:	Resultado:
			



c. Pila externa de 6V. Tome las dos pilas de 3V de litio que vienen en el kit, conéctelas en serie usando un cable que una el voltaje de una pila con la tierra de la otra y, mediante cables macho-macho, conecte el voltaje de las pilas en el pin de voltaje de entrada del Arduino y la tierra de las pilas en el pin de tierra del Arduino, como se muestra a continuación, de esta manera el Arduino se encenderá automáticamente y comenzará a ejecutar el programa que le hayas descargado.

Necesitará:	Use un cable para unir la tierra de una pila con el voltaje de la otra:	Conecte el extremo positivo de las pilas en el pin de voltaje de entrada del Arduino:	Conecte el extremo negativo de las pilas en el pin de tierra del Arduino:	Resultado:
				

¡Importante! Asegúrate de tener ambas baterías cargadas para asegurar un correcto funcionamiento de la tarjeta y todos los elementos electrónicos que conectes en ella.

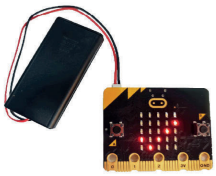
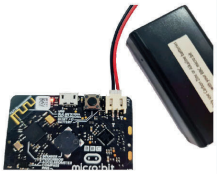
NOTA

Ten en cuenta que este método de alimentación NO tiene protección contra exceso de voltaje o conexiones invertidas, por lo que debes revisar detalladamente la conexión antes de alimentar el Arduino.

Micro:bit

Esta tarjeta la puede alimentar de dos maneras distintas:

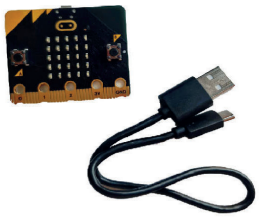
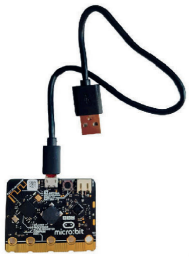
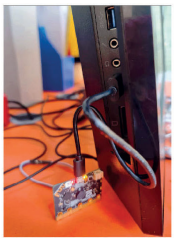
a. Pilas AAA. Tome el portapilas AAA, inserte las dos pilas AAA que trae el micro:bit y conecte el porta pilas en el puerto de conexión para baterías como se muestra en la imagen, de esta manera el micro:bit se encenderá automáticamente y comenzará a ejecutar el programa que le haya descargado:

Necesitará:	Verifica la polaridad:	Conecte el micro:bit:	Resultado:
			

NOTA

Verifique que la polaridad de las pilas sea correcta antes de conectar el portapilas al micro:bit.

b. Cable micro-USB. Conecta el micro:bit al cable micro USB y luego, conecte el otro extremo del cable a un computador con puerto USB como se muestra en la imagen, de esta manera el micro:bit se encenderá automáticamente y comenzará a ejecutar el programa que le haya descargado, y también te permitirá descargar el nuevo programa en cualquier momento.

Necesitarás:	Conecta el puerto micro-USB al micro:bit:	Conecta el cable al puerto USB del computador:	Resultado:
			

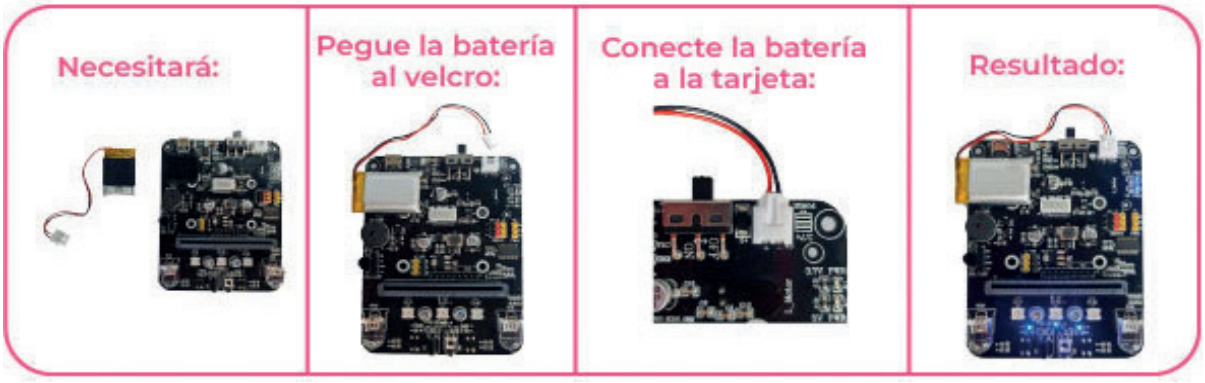
Building:bit

Esta tarjeta se alimenta con una batería de litio la cual se puede recargar conectando el robot al computador como se muestra a continuación:

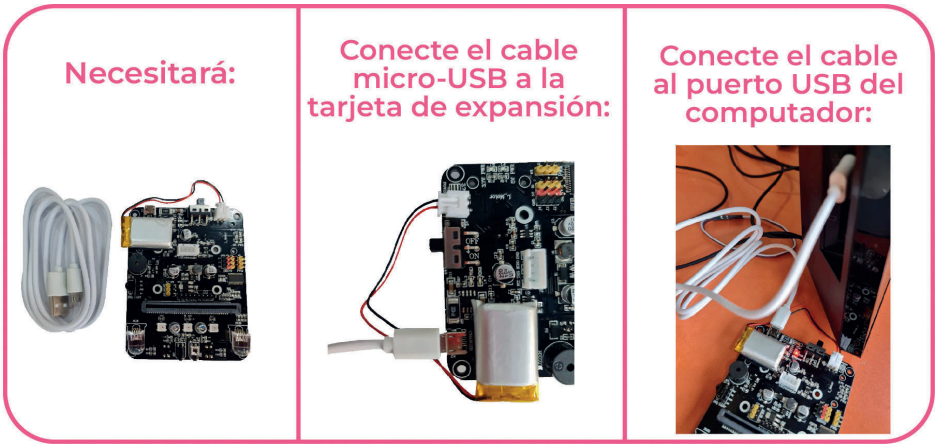
a. Alimentar la tarjeta con la batería de litio: Pegue la batería de litio en la tarjeta de expansión del Building:bit con el velcro, conecte la



batería en el interfaz de conexión para la batería de la tarjeta y listo, con solo mover el interruptor de encendido a ON se encenderá la tarjeta y estará lista para funcionar. Esta batería alimentará la tarjeta y todos los componentes que se conecten en ella como el micro:bit y los motores. Mueva el interruptor de encendido a OFF para apagar la tarjeta.



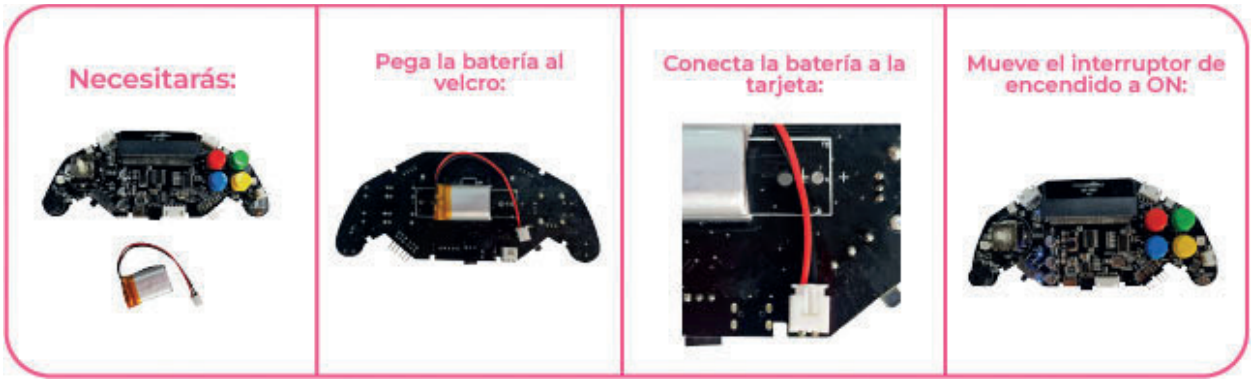
b. Cargar la batería de litio. Apague la tarjeta de expansión del Building:bit (moviendo el interruptor de encendido a OFF) con la batería conectada a la interfaz de conexión para la batería y conecte la tarjeta de expansión al computador usando el cable micro-USB, se encenderá un led rojo indicando que la batería se está cargando. Cuando la carga esté completa, el led pasará a color verde y podrá desconectarlo.



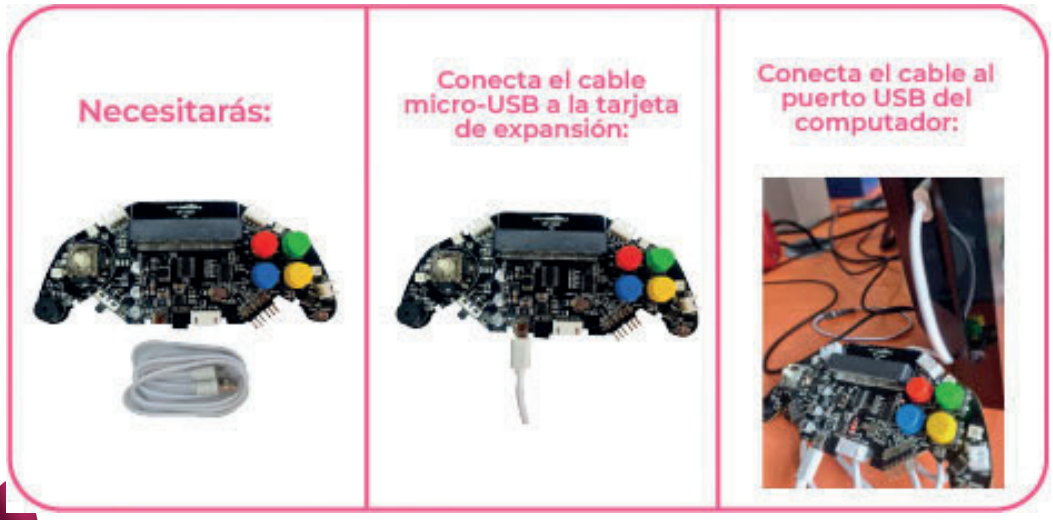
GamePad

Esta tarjeta se alimenta con una batería de litio, la cual se puede recargar conectando el GamePad al computador como se muestra a continuación:

a. Alimentar la tarjeta con la batería de litio: Pegue la batería de litio en la tarjeta de expansión del GamePad con el velcro, conecte la batería en el puerto de conexión para la batería de la tarjeta y listo, con solo mover el interruptor de encendido a ON se encenderá la tarjeta y estará lista para funcionar. Esta batería alimentará la tarjeta y todos los componentes que se conecten en ella como el micro:bit, los motores, los sensores, los LEDs, entre otros. Mueva el interruptor de encendido a OFF para apagar la tarjeta.



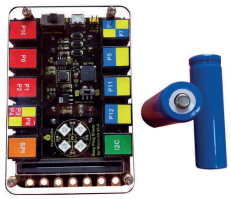
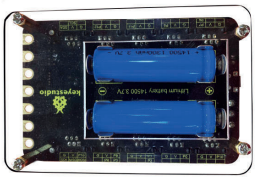
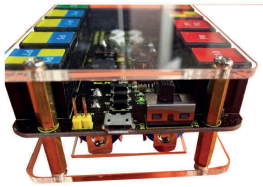
b. Cargar la batería de litio: Apague la tarjeta de expansión del GamePad (moviendo el interruptor de encendido a OFF) con la batería conectada al puerto de conexión para la batería y conecte la tarjeta de expansión al computador usando el cable micro-USB, allí se encenderá un led rojo indicando que la batería se está cargando. Cuando la carga esté completa, el led pasará a color verde y podrá desconectarlo.



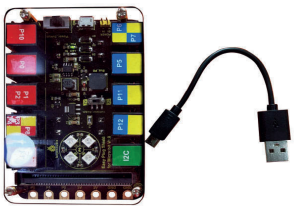
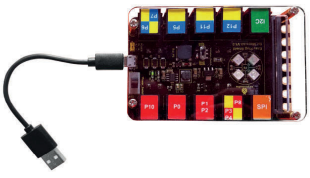
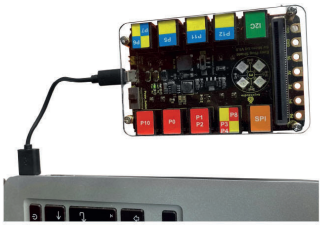
Tarjeta de expansión para sensores y actuadores

Esta tarjeta se alimenta con dos baterías recargables a través de su puerto de conexión para la fuente de alimentación, tal y como se muestra a continuación:

a. Alimentar la tarjeta con la baterías recargables: Conecte las baterías de litio en el puerto para baterías de litio en la parte inferior de la tarjeta teniendo en cuenta la orientación que se indica en la placa y listo, con solo mover el interruptor de encendido a ON se encenderá la tarjeta y estará lista para funcionar. Esta batería alimentará la tarjeta y todos los componentes que se conecten en ella como el micro:bit, los motores, los sensores, los ledes, entre otros. Mueva el interruptor de encendido a OFF para apagar la tarjeta.

Necesitará:	Conecte las baterías en el puerto para baterías:	Mueve el interruptor a ON:	Resultado:
			

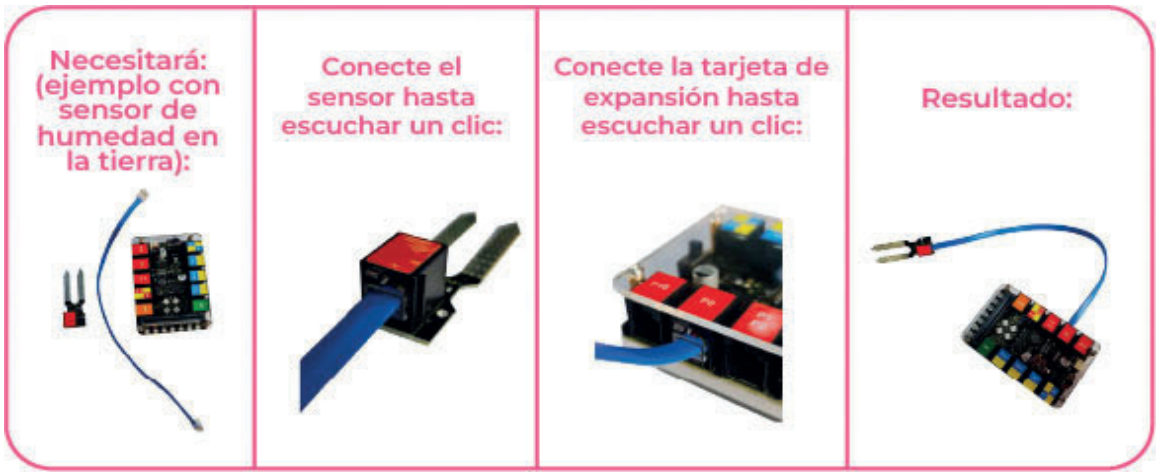
b. Cargar la batería de litio: Apague la tarjeta de expansión de sensores y actuadores (moviendo el interruptor de encendido a OFF) con la batería conectada al puerto de conexión para la batería y conecte la tarjeta de expansión al computador usando el cable micro-USB, se encenderá un led rojo indicando que la batería se está cargando. Cuando la carga esté completa, el led pasará a color verde y podrá desconectarlo..

Necesitará:	Conecte el cable micro-USB a la tarjeta de expansión:	Conecte el cable al puerto USB del computador:
		

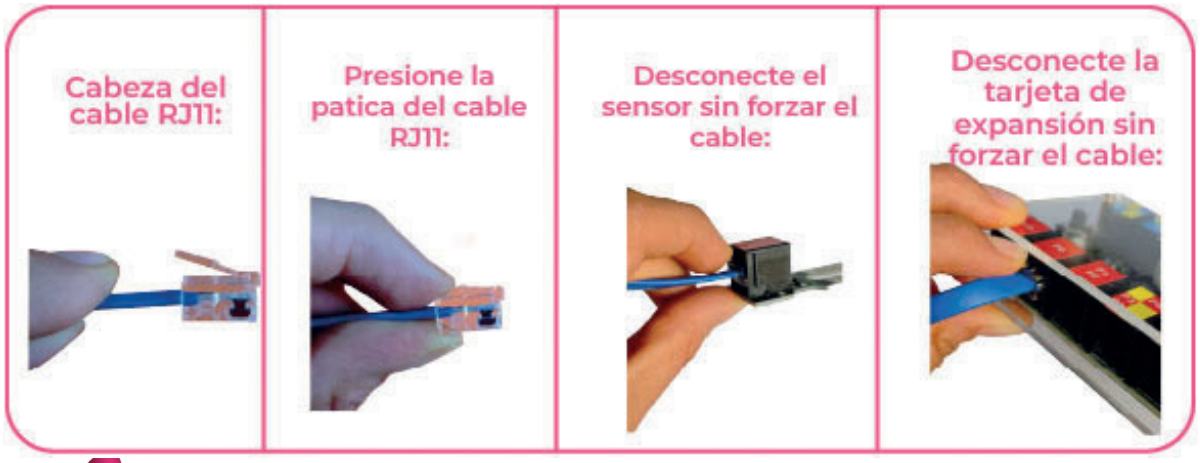
Conexión de sensores y actuadores

Algunos de los sensores que va a recibir tienen integrado el conector RJ11, lo que significa que para usarlos deberá conectarlos en la tarjeta de expansión para sensores, usando los cables de conexión RJ11 que vienen incluidos (cables azules), y siguiendo las indicaciones dadas en las cartillas. Estos sensores no se pueden conectar en otras tarjetas de expansión debido a que no poseen el puerto de conexión RJ11.

Estos sensores deberán conectarse a la tarjeta de expansión de forma similar a como se muestra a continuación:



Para desconectar los sensores y la tarjeta de expansión debe presionar las paticas de la cabeza del cable RJ11 como se muestra a continuación:





Los actuadores y sensores que no poseen el puerto de conexión RJ11 se deben conectar de la siguiente manera:

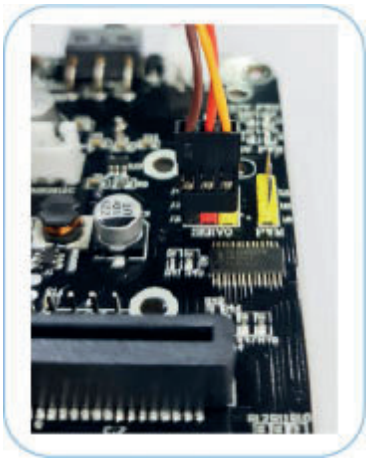
Servomotores

Son motores diseñados para controlar el ángulo de rotación de su eje. Este motor posee 3 cables de conexión:

a. Rojo. Aquí se debe conectar la fuente de alimentación o voltaje (V o rojo) preferiblemente de 5V.

b. Café. Aquí se debe conectar la tierra (GND o negro).

c. Amarillo o naranja. Aquí se debe conectar el pin de señal (S, J o amarillo), que será el encargado de enviarle al servomotor la información sobre el ángulo de rotación al que debe llegar.



En caso de conectar el servomotor a la tarjeta de expansión del **Building:bit,** se deben buscar los pines llamados SERVO y realizar la conexión como se indica a continuación:

- Cable rojo con 5V o pines rojos.
- Cable café con GND o pines negros.
- Cable amarillo con IO o pines amarillos.

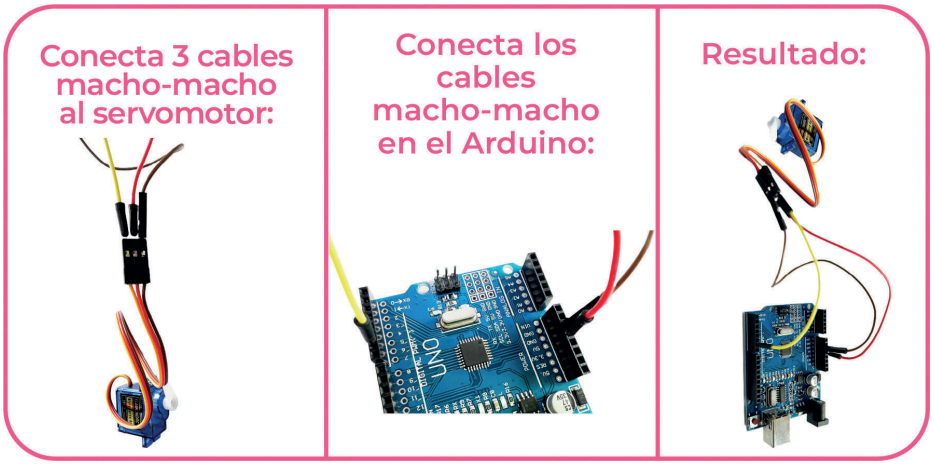


En caso de conectar el servomotor a la tarjeta de expansión del **GamePad,** se deben buscar los pines llamados SERVO y realizar la conexión como se indica a continuación:

- Cable rojo con VCC.
- Cable café con GND.
- Cable amarillo con IO.

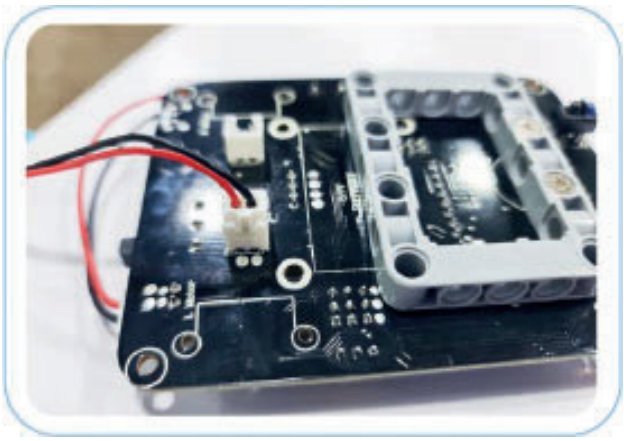
En caso de conectar el servomotor a la tarjeta Arduino, se deben realizar las siguientes conexiones:

- Cable rojo con 5V.
- Cable café con GND.
- Cable amarillo con los pines que tengan la virgulilla (~), como por ejemplo los pines 3, 5, 6, 9, 10, 11.



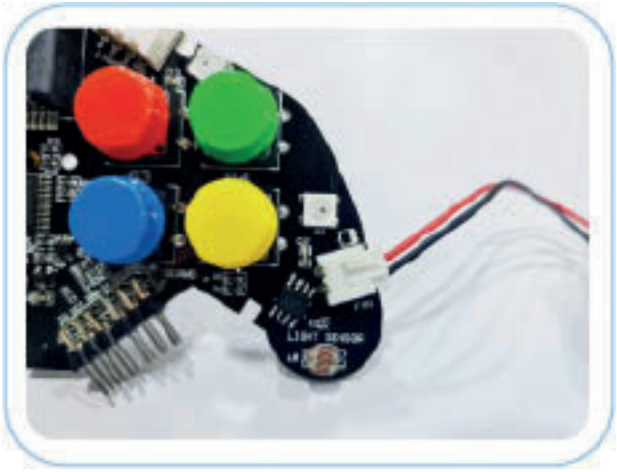
Motores DC

Estos motores están diseñados para controlar la velocidad y dirección de giro de sus ejes. Dependiendo del sentido de alimentación se puede girar el motor en una dirección o en otra, y dependiendo de la cantidad de voltaje que se le suministre, se varía la velocidad de rotación. Este motor tiene 2 cables, uno rojo y otro negro, y un puerto de conexión que nos indica en qué sentido conectar el elemento a la tarjeta de expansión (ya sea la del Building:bit o la del GamePad) como se muestra en las siguientes imágenes:

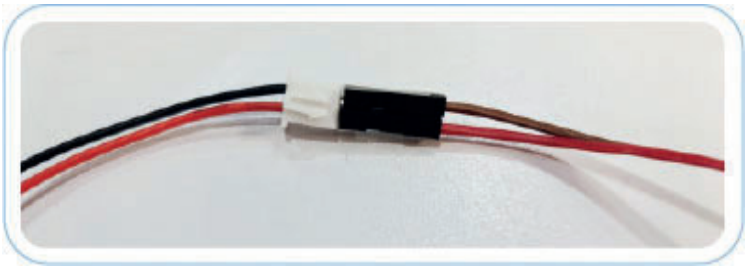


Conexión en el Building:bit (cara inferior de la placa):

Conexión en el GamePad:

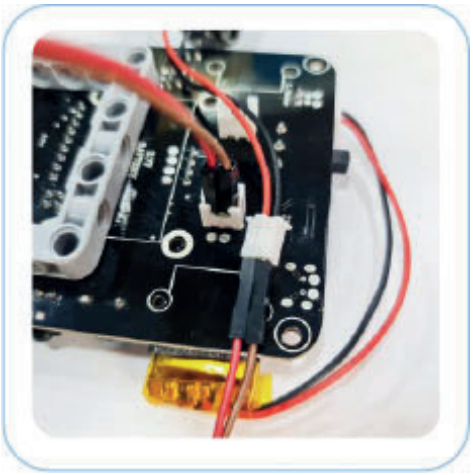


En caso de requerir cables adicionales para aumentar la longitud del cable del motor, busque conectar un cable hembra-macho rojo con el cable rojo del motor, y uno gris, negro o café con el cable negro del motor, tal y como se muestra a continuación.



Luego, verifique que conecta los cables hembra-macho en el puerto de motor de la tarjeta de expansión en el mismo sentido que lo haría si usara el conector original del motor.

Conexión en el Building:bit (cara inferior de la placa):



Conexión en el GamePad:



Conexión en el Arduino:

Para controlar un motor DC directamente desde el Arduino es necesario utilizar un driver para motor que se encargue de brindarle al actuador la energía necesaria para moverse, pero al mismo tiempo permita que las señales de baja potencia del Arduino controlen su comportamiento. El driver para motor es un dispositivo intermedio entre circuitos de baja y alta potencia, permitiendo que los motores consuman mayores cantidades de corriente sin sobre exigir los circuitos de baja potencia como el Arduino ni quemarlos.

Para realizar la conexión de estos elementos debes tener en cuenta que los motores necesitarán su propia fuente de alimentación ya que son muy exigentes con la energía, por lo que tendrás que usar la pila de 9V para alimentar el driver para motor que le transferirá esa energía a los motores y el Arduino tendrás que alimentarlo con el computador, con otra pila de 9V o con pilas de 6V por el puerto Vin, en pocas palabras, se necesitan dos fuentes de alimentación, una para los motores y otra para el Arduino.

Cuando se usa más de una fuente de alimentación es muy importante conectar las tierras de todas las fuentes entre sí para asegurar un correcto funcionamiento de todo el sistema.

Para realizar correctamente las conexiones del motor DC al driver para motor y de ahí al Arduino, apóyate de las marcas que tiene el driver en su placa

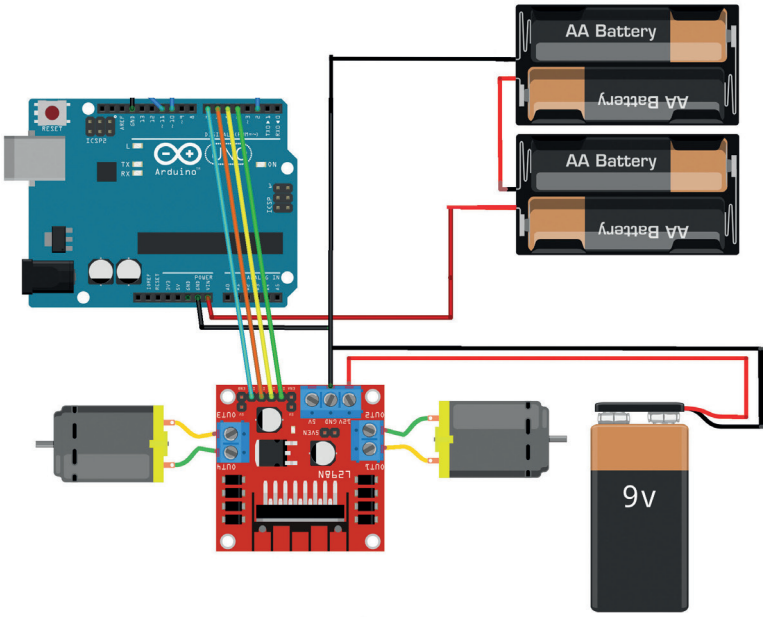


para identificar cada puerto, revisa tanto la parte superior como inferior de la placa para encontrar las marcas y sigue los pasos que se mencionan a continuación.

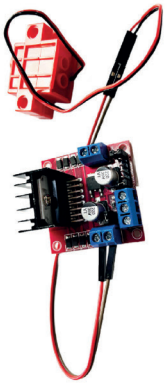
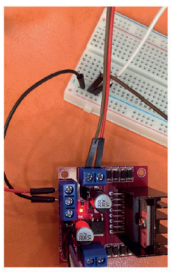
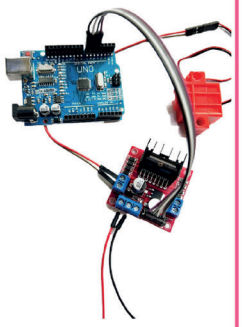
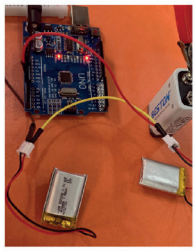
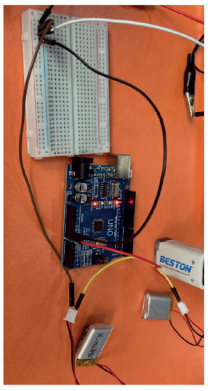
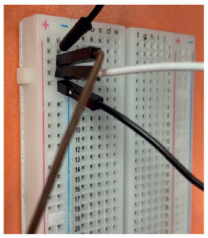
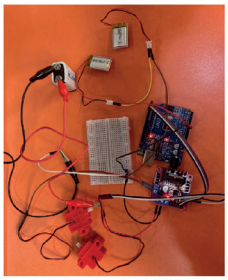
- Conecta la pila de 9V en los pines 12V y GND. Asegúrate de conectar el pin positivo (+ o rojo) en 12V y el pin negativo (- o negro) en GND.
- Para controlar un primer motor, conecta el motor en OUT1 y OUT2 y conecta los pines de control del Arduino en IN1 e IN2, ya que IN1 enciende o apaga OUT1, e IN2 enciende o apaga OUT2.
- Para controlar un segundo motor, conecta el motor en OUT3 y OUT4 y conecta los pines de control del Arduino en IN3 e IN4, ya que IN3 enciende o apaga OUT3, e IN4 enciende o apaga OUT4.
- Recuerda unir la tierra (GND) del Arduino con la tierra (GND) del driver para motor.

Ten en cuenta que para mover un motor en un sentido tienes que encender uno de sus pines y apagar el otro (por ejemplo, OUT1 encendido y OUT2 apagado), y para cambiar el sentido de giro debes invertir los pines (por ejemplo, OUT1 apagado y OUT2 encendido) y para detenerlo debes apagar ambos pines (por ejemplo, OUT1 apagado y OUT2 apagado).

Como puedes observar, es posible controlar dos motores de forma independiente entre sí: el uno no depende del otro y, en caso de que alguno de los motores no gire en el sentido deseado, simplemente debes intercambiar los cables en los puertos OUT para invertir su polaridad, sin necesidad de modificar el código.



Las conexiones físicas de estos elementos se realizan de la siguiente forma:

Conexión del motor al Driver	Conexión de la pila de 9V al Driver	Conexión del Arduino al Driver	Unión de tierras entre las fuentes de 9V y 6V	Conexión completa con dos motores DC
	1. Conecte caimanes a la pila 9V (rojo en positivo y negro en negativo), y conecte el caimán negro al cable blanco  2. Conecte el caimán rojo al cable rojo del Driver  3. Conecte el cable blanco a la protoboard junto con el cable negro del Drive 		1. Conecte el voltaje de las pilas de 3V en Vin del Arduino  2. Conecte la tierra de las pilas de 3V en la protoboard junto con el cable de GND del Driver  Cables de tierra conectados: 	

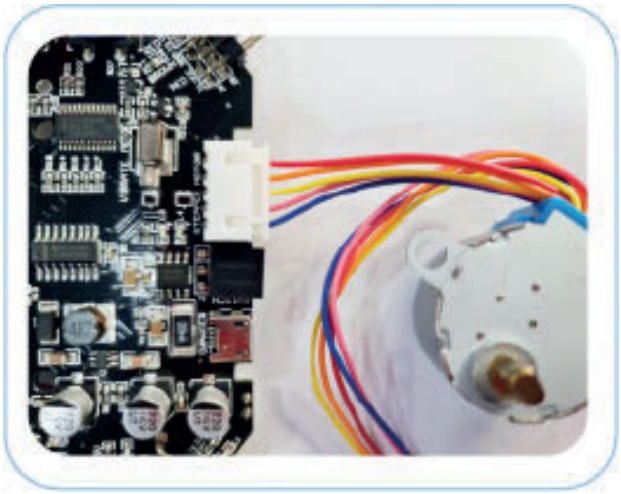
NOTA

Ten en cuenta que necesitarás un destornillador para ajustar y desajustar las borneras del driver donde conectarás los cables.



Motor paso a paso

Estos motores están diseñados para controlar la rotación de su eje, moviéndolo cada cierta cantidad de grados (predefinida por su estructura interna) de forma constante, de tal manera que cada "paso" que da el motor es de un grado específico. Este motor tiene 5 cables y un puerto de conexión que impide que se conecte de forma incorrecta, por lo que solo es necesario conectar el motor al puerto destinado para el mismo en el GamePad, como se muestra en la siguiente imagen:



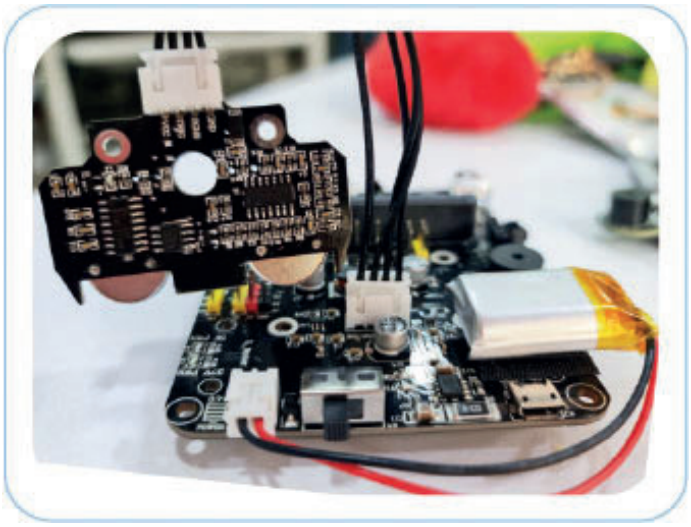
En caso de requerir cables adicionales para aumentar la longitud del cable del actuador, busca conectar un cable hembra-macho rojo con el cable rojo del motor, uno café con el cable naranja del motor, uno blanco con el cable amarillo del motor, uno gris con el cable rosado del motor y uno morado con el cable azul del motor. Luego, verifica que conectas los cables hembra-macho en el puerto del actuador de la tarjeta de expansión en el mismo sentido que lo haría si usara el conector original del actuador, como se muestra en la siguiente imagen:



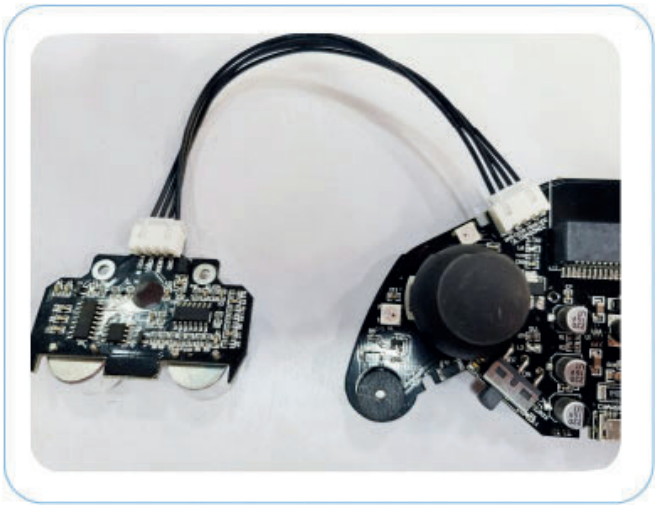
Sensor de ultrasonido

Este sensor usa ondas de ultrasonido para medir distancias. Para usarlo se requiere de una fuente de alimentación (voltaje y tierra) y dos pines programables, uno de salida (Trig) y otro de entrada (echo). Este sensor tiene 4 cables y un puerto de conexión que impide que se conecte de forma incorrecta, por lo que solo es necesario conectar el sensor al puerto destinado para el mismo en el GamePad o el Building:bit, como se muestra en las siguientes imágenes:

Conexión en el Building:bit:

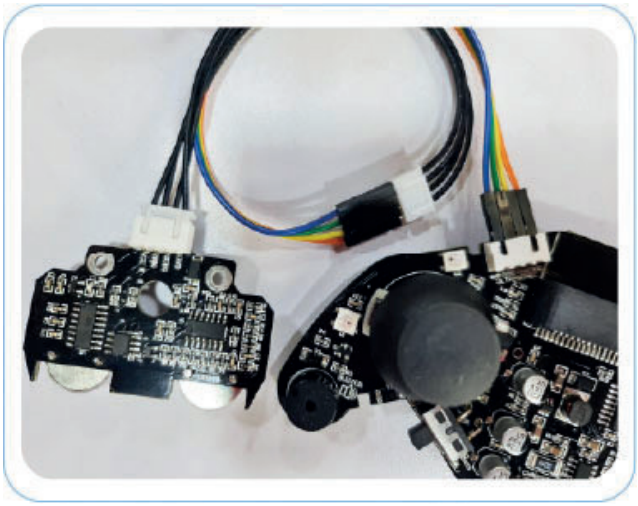


Conexión en el GamePad:

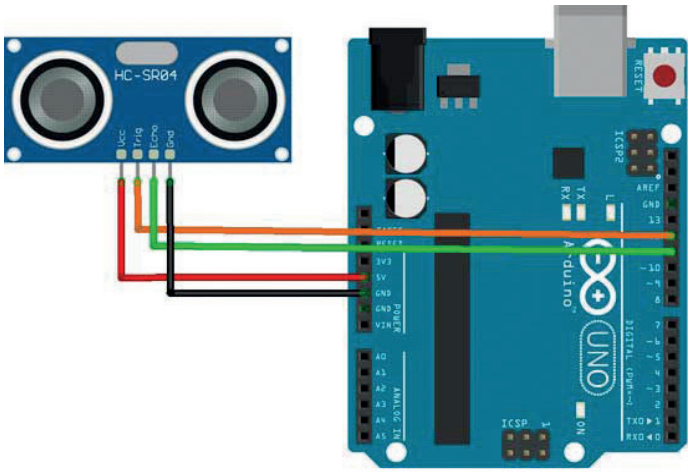


En caso de requerir cables adicionales para aumentar la longitud del cable del sensor, busque conectar un cable hembra-macho azul con el cable GND del sensor, uno verde con el cable ECHO del sensor, uno amarillo con el cable TRIG del sensor, y uno naranja con el cable VCC del sensor. Luego, verifique que conecta los cables hembra-macho en el puerto del sensor de la tarjeta de expansión en el mismo sentido que lo haría si usara el conector original del sensor, como se muestra en las siguientes imágenes (verifique que conecta GND del sensor con GND de la tarjeta, ECHO del sensor con ECHO de la tarjeta, TRIG del sensor con TRIG de la tarjeta y VCC del sensor con VCC de la tarjeta).

Conexión en el Building:bit:



Conexión en el Arduino:



Conecta el cable de voltaje VCC del ultrasonido en el pin de voltaje del Arduino 5V, el cable de tierra GND del ultrasonido en el pin de tierra del Arduino GND, y los cables Trig y Echo en los pines que se hayan escogido en la programación para cada uno de ellos (para el ejemplo, Trig en 12 y Echo en 11).

Potenciómetro del GamePad

Este elemento es una resistencia variable, la cual cambia la posición rotacional de su eje en un cambio de resistencia que, al pasar por el módulo que tiene integrado a sus pines, se convierte en una variación de voltaje, la cual puede ser medida e interpretada por el micro:bit como si fuera un sensor analógico. Este elemento tiene 3 pines, dos de alimentación (voltaje VCC y tierra GND) y uno de salida (IO) donde se puede leer el cambio de voltaje al cambiar la posición rotacional de su eje.



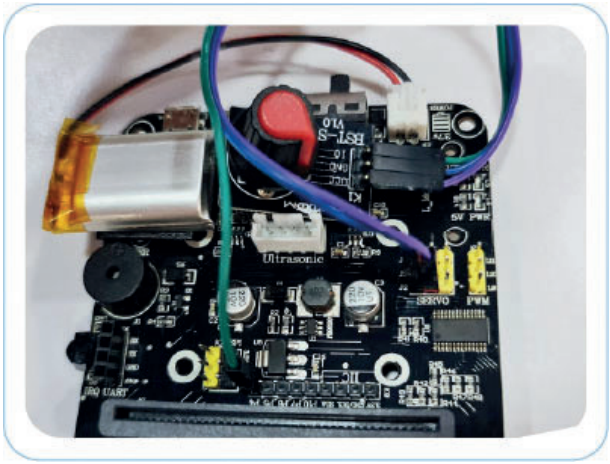
Para conectar este elemento en el GamePad debe ubicarlo bocarriba como se muestra en la imagen:

Si desea usar cables para ubicar el sensor más lejos del GamePad asegúrese de conectar VCC en el orificio derecho del puerto de conexión del GamePad, IO en el izquierdo y GND en el centro (use cables hembra-macho), como se muestra en la imagen:

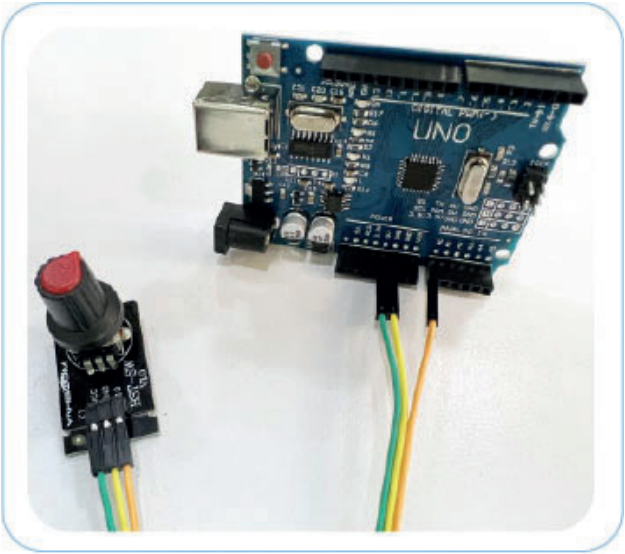




Para conectar este elemento en el Building:bit, debe conectar sus pines de alimentación en los pines de voltaje y tierra de la tarjeta de expansión del robot (VCC con voltaje o pines rojos y GND con tierra o pines negros), y el pin de salida IO en P4 o P10 (use cables hembra-hembra), como se muestra en la imagen:



Para conectar este elemento en el Arduino, debe conectar los pines de alimentación del potenciómetro en los pines de voltaje y tierra del Arduino (VCC con voltaje o pin 5V, y GND con tierra o pines GND), y el pin de salida IO en cualquiera de los pines de A0 a A5 (use cables hembra-macho), como se muestra en la imagen:



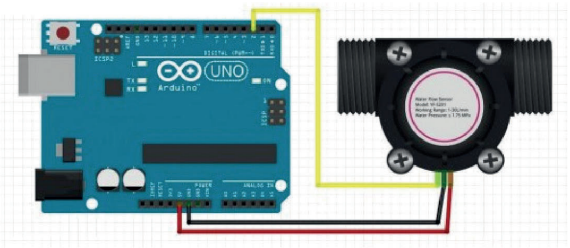
Sensor de caudal y sensor de calidad de agua.

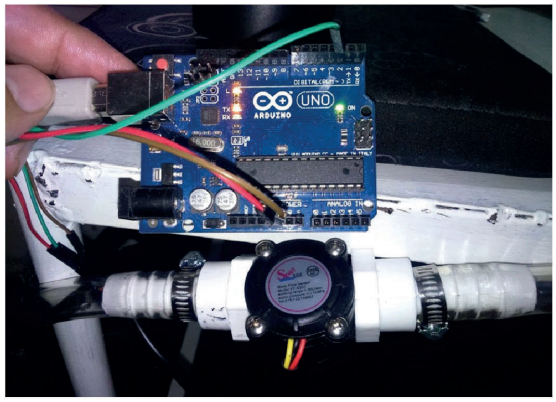
Estos sensores se caracterizan por entregar como resultado un voltaje que representa numéricamente el nivel de la variable física medida. En el caso del sensor de caudal, se genera una señal digital que representa el nivel de caudal de una tubería o del líquido que pasa a través del sensor, mientras que el sensor de calidad de agua usa un valor analógico para representar las partes por millón (PPM) de una muestra de agua con el fin de indicar su calidad.

Ambos sensores se pueden usar tanto con micro:bit como con Arduino, sin embargo, en las rutas didácticas se enfoca su uso al Arduino debido a la complejidad de las operaciones matemáticas que se deben realizar por código para convertir la información recibida por el sensor en una magnitud física real.

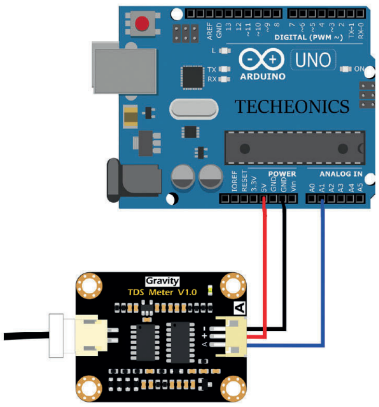
La conexión de ambos sensores es muy similar entre sí, el cable rojo (pin VCC, +, V) se conecta en los pines de voltaje de la tarjeta, el cable negro (pin GND, -, G) se conecta en los pines de tierra, y el cable amarillo (pin S, Sig, A) se conecta en los pines de señal (el sensor de calidad de agua se debe conectar en los pines de entrada analógica marcados con la letra “A”, mientras que el sensor de caudal se debe conectar en los pines digitales marcados solo con números).

Conexión sensor de caudal





Conexión sensor de calidad de agua

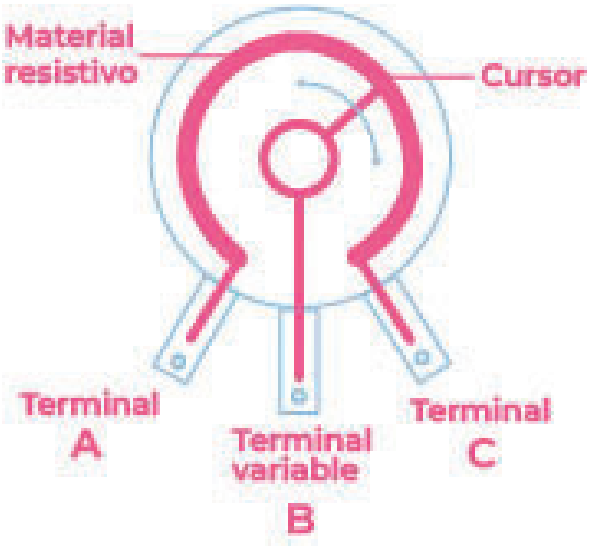




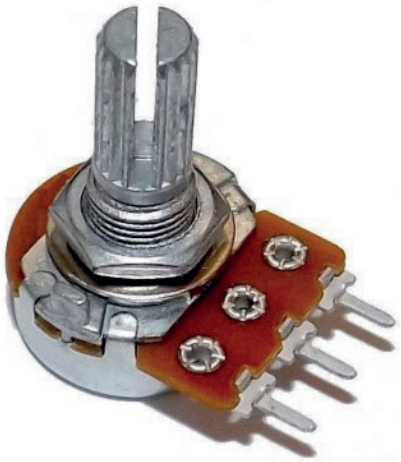
Resistencia variable Potenciómetro

El potenciómetro consta de una pista de carbón o cermet sobre la cual se ubica un cursor conectado a un patín que se desliza por la resistencia, haciendo que varíe el valor de salida. La forma de medir la resistencia variable de un potenciómetro es tomando el valor entre uno de sus terminales fijos y la terminal variable. Si se mide la resistencia entre los dos terminales fijos el valor siempre será igual a la resistencia máxima del potenciómetro.

Estructura interna del potenciómetro:



El potenciómetro:



Para conectar el potenciómetro a una tarjeta programable, ya sea el Arduino o el micro:bit, se debe:

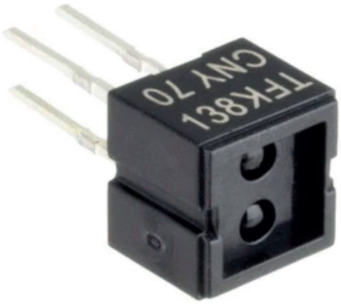
- Conectar un terminal fijo en voltaje (3V para micro:bit o 5V para Arduino)
- Conectar el otro terminal fijo en tierra (GND)
- Conectar el terminal variable en un pin de señal.

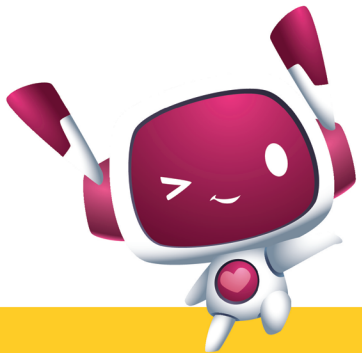


Sensor CNY

El CNY70 es un sensor óptico infrarrojo de corto alcance muy utilizado para aplicaciones de robótica en el desarrollo de aplicaciones como: seguidores de línea, robots laberinto, robots sumo y velocistas.

Este sensor tiene integrados un emisor y un receptor infrarrojo (IR), enfocados en la misma dirección, donde el emisor genera una luz infrarroja y el receptor la recibe cuando esta rebota contra un objeto, lo que permite diferenciar los colores blanco y negro, donde el blanco refleja la luz y el negro la absorbe, impidiendo que regrese al receptor. Dependiendo de si el receptor recibe o no la señal infrarroja emitida por el emisor, la salida del sensor cambia de 0 a 1 o viceversa.





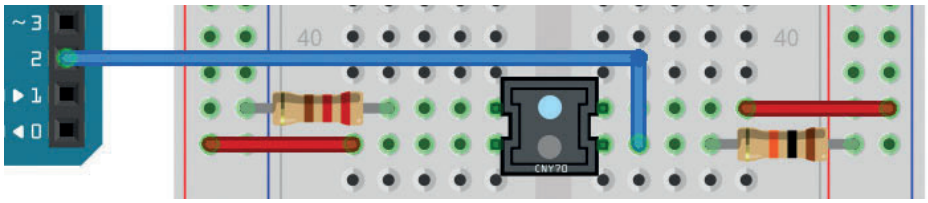
Este tipo de sensor se debe conectar con un conjunto de resistencias específico para utilizarlo como detector de blanco y negro.

La conexión debe realizarse de la siguiente manera:

- Conecta los dos pines del sensor que se muestran en la imagen (inferior izquierdo y superior derecho) en voltaje (cable rojo).

¡Importante! Ten en cuenta la orientación del sensor, mira dónde está marcado el nombre "CNY70" y orienta tu sensor de la misma manera. Si la conexión no se realiza exactamente igual que la imagen, el sensor fallará.

- En los pines restantes, conecta una resistencia en cada uno y el otro extremo de la resistencia conéctalo en tierra.
- En el pin superior izquierdo conecta una resistencia de 220 ohmios.
- En el pin inferior derecho conecta una resistencia de 10 kilo ohmios.
- Usa un cable para conectar la salida del sensor con la tarjeta programable. Ten en cuenta que la salida del sensor es el pin que conectaste a la resistencia de 10 kilo ohmios, es decir, el pin inferior derecho.



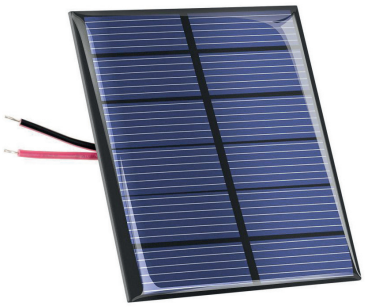
Con esta conexión, el sensor será capaz de diferenciar los colores blanco y negro, siempre y cuando la distancia de detección sea menor a 2cm y el entorno esté iluminado con luz artificial, sin incidencia de luz natural.

Paneles solares

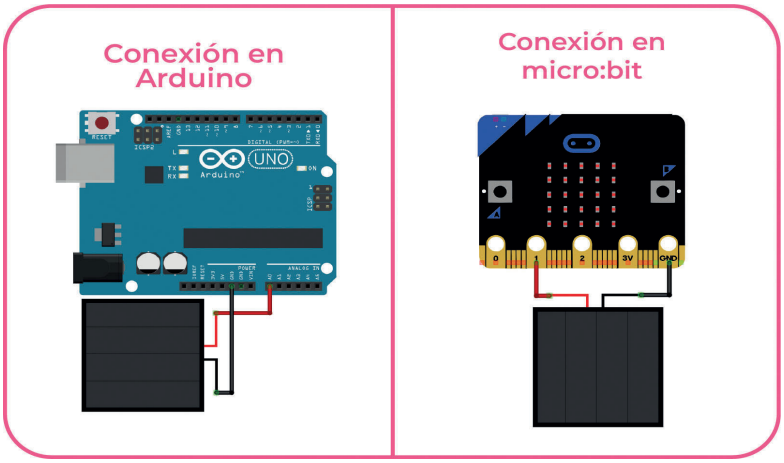
Los paneles solares son dispositivos electrónicos que aprovechan la energía del sol para generar calor o electricidad, y constan de módulos fotovoltaicos individuales que son los encargados de captar la energía del sol y convertirla en electricidad.

Los módulos fotovoltaicos contienen células fotovoltaicas hechas de materiales semiconductores, como el silicio, que se encargan de transformar la luz del sol en energía eléctrica. Este proceso se da debido a que, cuando

la luz del sol incide sobre el panel, se separan los electrones de las células fotovoltaicas formando una capa de carga positiva y otra de carga negativa, lo que produce una diferencia de potencial que genera una corriente eléctrica.



Para conectar el panel solar a cualquiera de las tarjetas programables, debes conectar el cable negro (-, tierra) en los pines de tierra (GND) y el cable rojo en un pin de señal de lectura analógica, lo que permitirá leer el nivel de voltaje que está generando el panel por la incidencia del sol.



Fotoresistencia

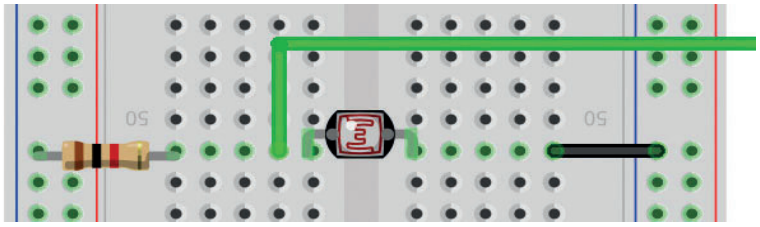
La fotoresistencia es un dispositivo electrónico conformado por una resistencia que varía según la intensidad de luz. Está hecho con sulfuro de cadmio, un material semiconductor que tiene la capacidad de variar su resistencia según la cantidad de luz del entorno.





Para usar este sensor, se debe conectar una resistencia fija de valor conocido en serie con el fotoresistor, alimentar ambas resistencias y tomar la lectura del cambio de voltaje que se genera sobre una de ellas debido al cambio de resistencia provocado en el sensor. La conexión debe realizarse como se muestra a continuación:

- Conecta un pin de una resistencia de 1 kilo ohmio a voltaje y el otro pin a la fotoresistencia.
- Conecta un pin de la fotoresistencia a la resistencia de 1 kilo ohmio y el otro pin a tierra.
- Conecta un cable en medio de las dos resistencias y de ahí conéctalo a un pin de señal de cualquiera de las tarjetas programables. Esto te permitirá leer los cambios de voltaje generados por la variación de luz ambiente en el sensor.



Software

Programación en MakeCode – Generalidades

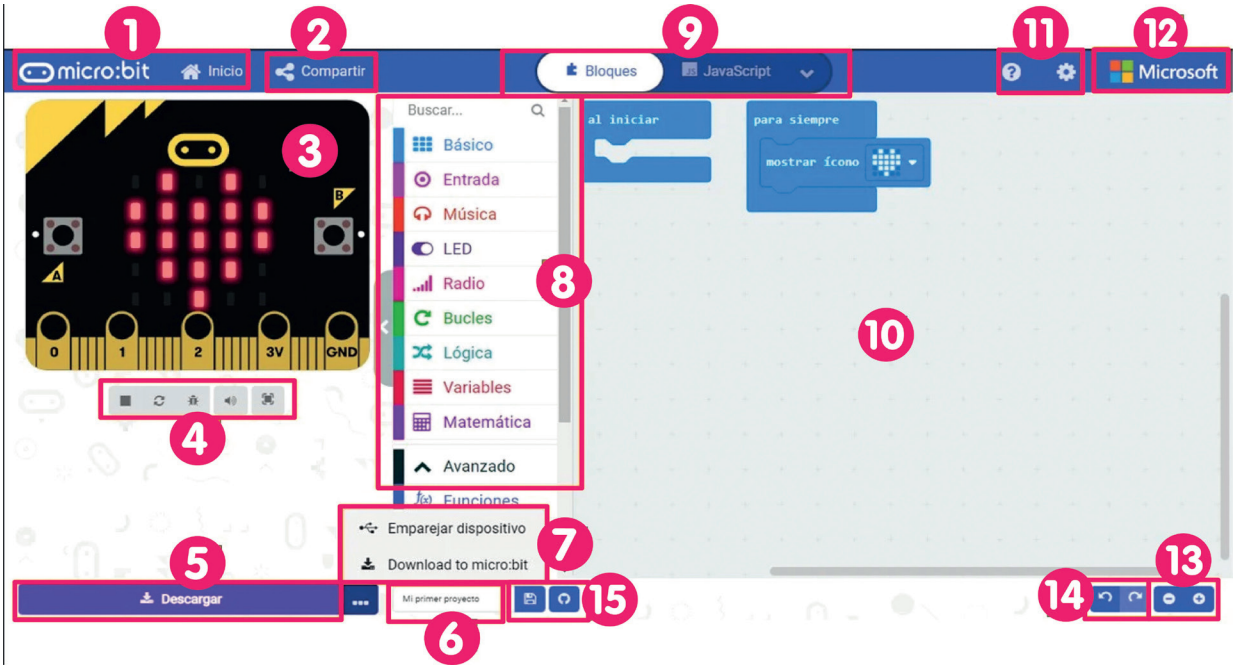
MakeCode es una plataforma de programación por bloques, donde cada bloque representa una instrucción o herramienta de programación que puede conectarse con otros bloques para formar un algoritmo, sin necesidad de conocer un lenguaje de programación específico ni de memorizar funciones que permitan realizar distintas tareas.

Esta plataforma cuenta con:

- Un emulador al costado izquierdo donde se puede simular, en un micro:bit virtual, todo lo que se ha programado para verificar la lógica y el funcionamiento del algoritmo.
- Un listado de categorías en el centro dentro del cual se encuentran todos los bloques de programación. Esta división de bloques por categorías facilita la búsqueda de las diferentes instrucciones de programación que se pueden usar en cualquier proyecto, y vienen divididas por colores para un reconocimiento visual mucho más rápido.

- Un espacio de trabajo al costado derecho donde se colocan y se unen todos los bloques de programación para formar un algoritmo.

Esta plataforma de programación está compuesta por los siguientes elementos:



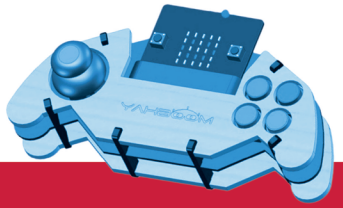
1. Haga clic en cualquiera de estos dos botones para explorar la página principal de MakeCode.
2. ¡Muestre su destreza! Con este botón puede compartir su proyecto con sus amigos y profesores.
3. Emulador, acá es donde ocurre la magia. Todo lo que programe será visualizado previamente en el emulador.
4. Barra de herramientas del Emulador. Conozca cada botón para mejorar su habilidad en MakeCode mientras programas.
5. Botón para descargar proyecto en el dispositivo o cargar al Micro:bit con el programa realizado.
6. Haga clic y cambie el nombre del Proyecto, ponga un nombre genial a su proyecto.
7. Una el mundo virtual con el real: Empareje Micro:bit con MakeCode y descargue su proyecto directamente al Micro:bit conectado. Se sorprenderá.
8. ¡El centro de mando! Esta paleta presenta todas las categorías de programación. No tenga miedo, ¡haga clic en todos!



9. ¡Sea un experto! Haga clic y visualice su programa en Bloques o lenguaje JavaScript. Aprenda un nuevo lenguaje.
10. Arrastre sus bloques a esta zona. Área de trabajo y de magia.
11. Haga clic si tiene dudas: Ayuda y Configuración general de MakeCode.
12. ¿Quiere más? Haga clic en la página de soluciones educativas Microsoft.
13. ¿Códigos largos? Haga clic en las herramientas de Zoom para el área de trabajo (alejar, acercar).
14. Si comete un error con un bloque, no se preocupe: Herramientas para deshacer o rehacer escritura.
15. Haga clic en Guardar avance y conviértase en un verdadero explorador visitando el repositorio de librerías GitHub.

A continuación, se listan las categorías de bloques que tiene MakeCode para programar y se da una breve descripción de ellos y el tipo de tareas que realizan:

- **Básico:** Contiene bloques que realizan tareas básicas en el micro:bit, como escribir en la matriz led, mostrar números, letras, borrar pantalla, pausas, entre otros.
- **Entrada:** Contiene bloques que leen las diferentes entradas del micro:bit, desde los botones físicos, hasta los sensores que tiene integrados como el Touch, el acelerómetro, el de luz, entre otros.
- **Música:** Contiene bloques que generan sonidos musicales, desde notas individuales hasta melodías, silencios y demás.
- **Led:** Estos bloques controlan específicamente cada led del micro:bit, ya sea para encenderlo, apagarlo, cambiar su intensidad de brillo, entre otros.
- **Radio:** Contiene bloques que permiten establecer una comunicación por radio, enviar y recibir información.
- **Bucles:** Estos bloques están diseñados para repetir instrucciones de programación, ya sea definiendo un número fijo de repeticiones o estableciendo una condición que detenga la repetición.
- **Lógica:** Contiene bloques que permiten programar condicionales, donde el programa debe tomar una decisión frente a una condición y realizar acciones en caso de ser verdadera o falsa la condición.
- **Variables:** En este espacio el programador puede crear variables para guardar y modificar información ya sea numérica, de texto, imágenes, entre otros.
- **Matemáticas:** Contiene bloques que realizan operaciones matemáticas, desde sumas y restas, hasta raíces cuadradas, redondeos y demás.

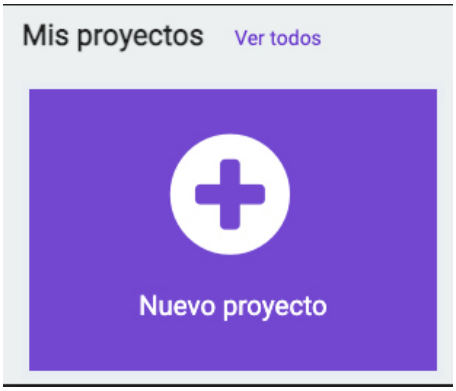


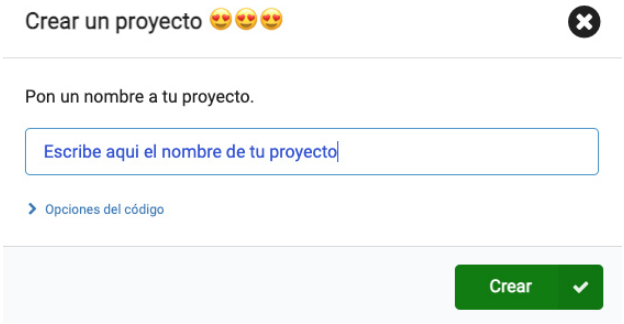
- **Extensiones:** En esta sección, el programador puede ingresar para buscar más bloques de programación que le permitan programar tareas adicionales en el micro:bit o controlar algunos de sus accesorios, como robots, sensores, actuadores y demás.
- **Funciones:** En este espacio el programador puede crear funciones, establecer sus parámetros de entrada y ubicarlas en cualquier parte del programa.
- **Arreglos:** Contiene bloques que permiten guardar más de un dato de información en una misma variable.
- **Texto:** Contiene bloques que se encargan de escribir y evaluar cadenas de texto.
- **Juego:** Contiene bloques para la programación de videojuegos que incluye llevar un conteo de las vidas, la puntuación, entre otros.
- **Imágenes:** Contiene bloques que permiten crear imágenes grandes, imágenes en movimiento, flechas en movimiento, entre otros.
- **Pines:** Contiene bloques que permiten programar los pines de entrada/salida del micro:bit.
- **Serial:** Contiene bloques que permiten enviar y recibir información por el puerto serial. Estos bloques son muy utilizados para mostrar información del micro:bit en el monitor serial del MakeCode.
- **Control:** Contiene bloques especializados para programar diferentes elementos internos del micro:bit. Estos bloques requieren conocimientos de programación intermedia-avanzada para usarlos.

Para crear y programar un algoritmo en esta plataforma se deben realizar los siguientes pasos básicos:

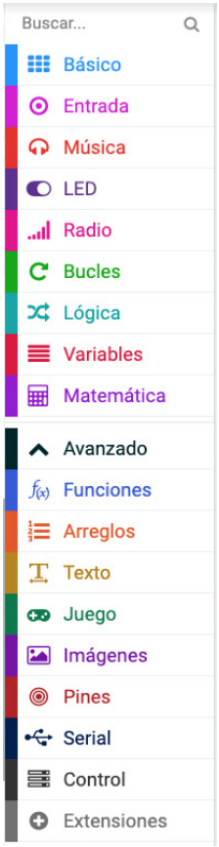


1. Cree un nuevo proyecto y asigne un nombre relacionado con lo que desea programar.

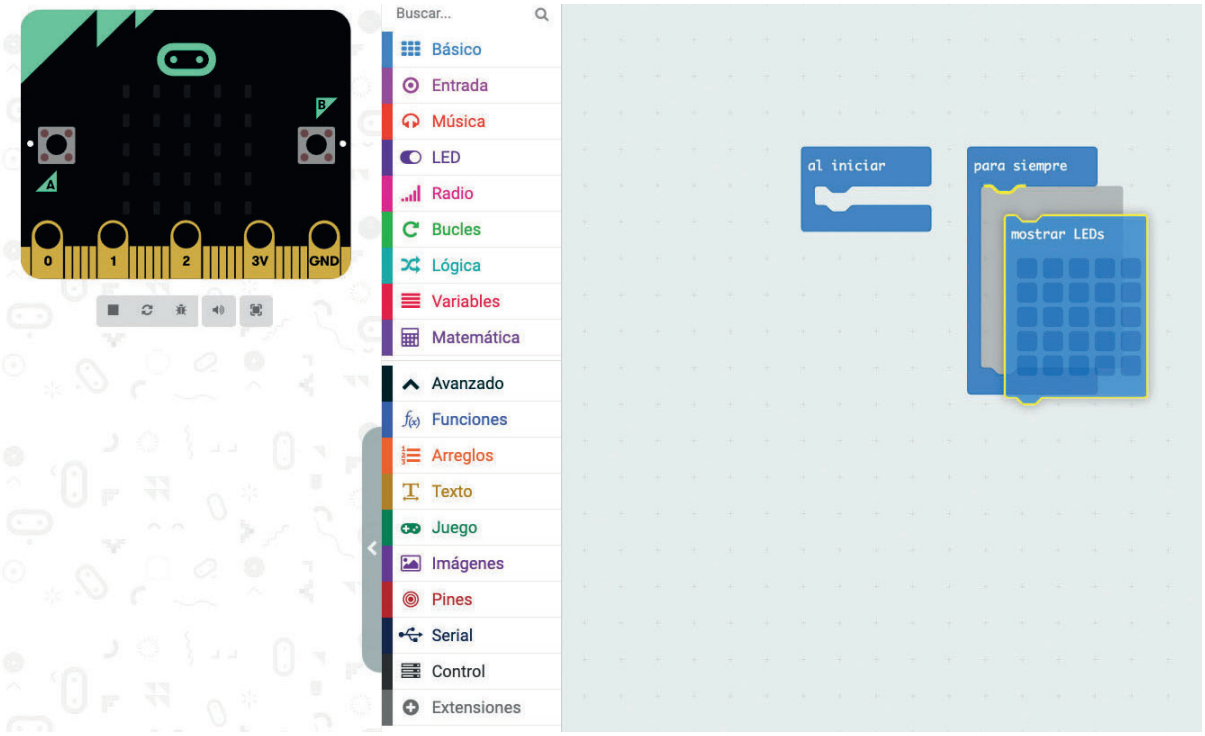




2. Busque en las categorías del centro los bloques que necesita para realizar el algoritmo.



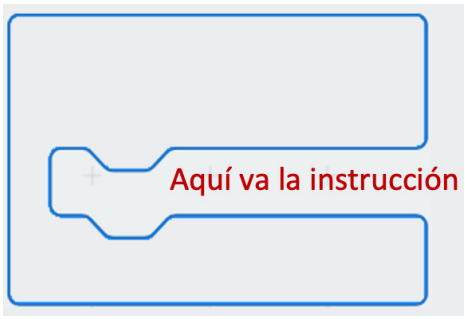
3. Haga clic sostenido en el bloque que desee usar, arrástrelo al espacio de trabajo y suéltelo, teniendo en cuenta que cada bloque encaja con los demás dependiendo de su forma geométrica.



4. Los tipos de bloques que encontrará en MakeCode según su forma geométrica y cómo programarlos son los siguientes:

a. Bloques de ejecución:

- No encajan dentro de otros bloques de programación. Debe arrastrarlos hasta el área de trabajo y soltarlos allí para que funcionen.
- Están diseñados para que agregue instrucciones en su interior y estas se ejecuten en el micro:bit.



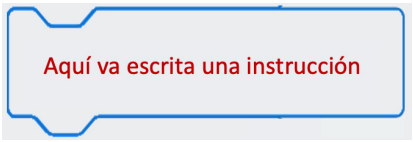


- Por ejemplo, el bloque “Para siempre” un bloque de ejecución dentro del cual se puede programar una tarea que desee que se repita una y otra vez, indefinidamente, como lo muestra su nombre:

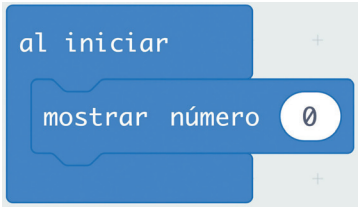


b. Bloques de instrucción:

- Están diseñados para arrastrarlos hasta los bloques de ejecución o los bloques condicionales, y soltarlos allí para ejecutar tareas.
- Poseen instrucciones escritas en su interior, que se ejecutan en el micro:bit.

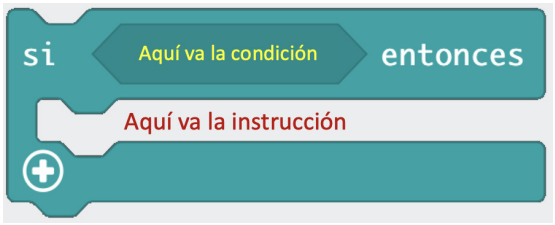


- Por ejemplo, el bloque “Mostrar número” un bloque de instrucción con el cual se programa la matriz de ledes del micro:bit:

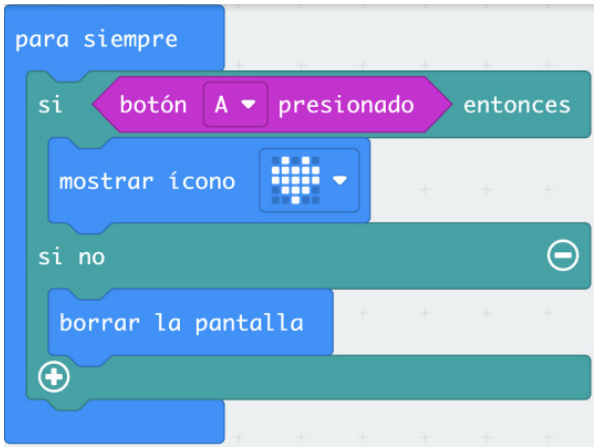


c. Bloques condicionales:

- Están diseñados para arrastrarlos hasta los bloques de ejecución y soltarlos ahí para tomar decisiones.
- Poseen un espacio con lados triangulares para agregar una condición, y otro rectangular para agregar bloques de instrucción y realizar tareas.



- Por ejemplo, el bloque “Si … entonces … sino” es un bloque condicional con el cual se puede tomar una decisión como mostrar un corazón cuando se oprime un botón y borrarlo cuando no se presiona nada:



d. Bloques de lectura:

- Están diseñados para arrastrarlos hasta los bloques de instrucción, bloques de condición o bloques condicionales, y soltarlos ahí para obtener un tipo de información específica y usarla en cualquier parte del algoritmo.
- Estos bloques poseen información numérica, lógica o de texto.



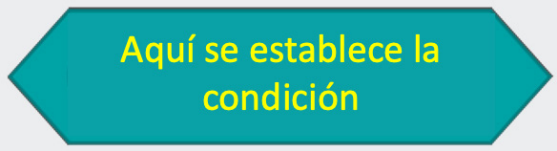
- Por ejemplo, el bloque “Temperatura (°C)” es un bloque de lectura con el cual puede conocer la temperatura ambiente medida por el micro:bit y usarla para realizar alguna tarea, como mostrarla en la matriz led del micro:bit:



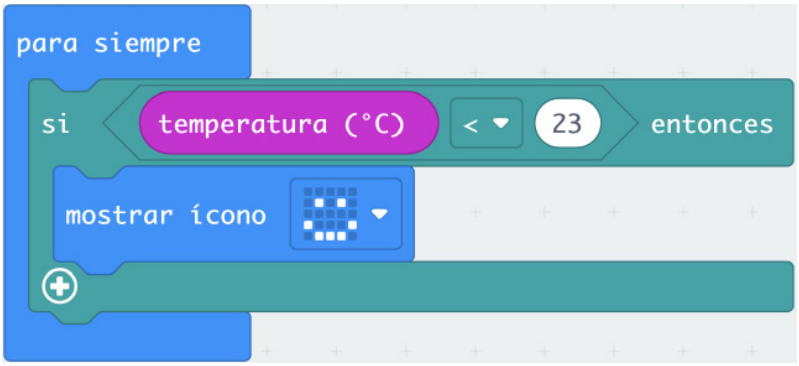


e. Bloques de condición:

- Están diseñados para que los arrastre hasta los bloques condicionales y los suelte allí para definir la condición a partir de la cual se tomarán decisiones.
- Dentro de estos bloques se define una condición que puede ser de tipo mayor (>), menor (<), igual (=), verdadero, falso, entre otros.



- Por ejemplo, el bloque "< (menor a)" es un bloque de condición con el cual puede comparar dos valores, como lo son la temperatura registrada del micro:bit con un valor numérico seleccionado por nosotros, y mostrar una carita feliz si la condición se cumple:



5. Ten en cuenta que debes ubicar los bloques en orden según lo que desea que se realice primero en el proyecto, pues las tareas programadas en el micro:bit se ejecutan de forma secuencial, realizando primero las tareas de arriba y luego las de abajo. Es importante considerar que las instrucciones programadas en los bloques:

a. "Para siempre" se ejecutan de forma indefinida, repitiéndose por siempre.
b. "Al iniciar" se ejecutan una sola vez al iniciar el programa y no vuelven a realizarse en el resto del algoritmo a menos que se reinicie el micro:bit.
c. "Al presionar" o "si agitado" se ejecutan cada vez que se realiza la acción descrita en el bloque (presionar un botón, un pin, realizar un gesto en el micro:bit, entre otros).

6. Al terminar de programar el algoritmo, puede probarlo en el emulador haciendo clic en los botones, agitando o inclinando la tarjeta micro:bit con el ratón, o variando los medidores de luz y temperatura, entre otras acciones más, dependiendo de las tareas que se haya programado en el proyecto.

7. Después de verificar el funcionamiento del algoritmo en el simulador puede pasar a programarlo en el micro:bit físico.

Librerías del kit STEM

Algunos elementos del kit STEM, como el GamePad y el robot móvil Building:bit, tienen librerías que permiten controlar los diferentes elementos que están integrados en sus tarjetas (como sensores, actuadores y ledes), y los dispositivos electrónicos adicionales que se pueden conectar a ellos, como los accesorios del GamePad o los motores del Building:bit.

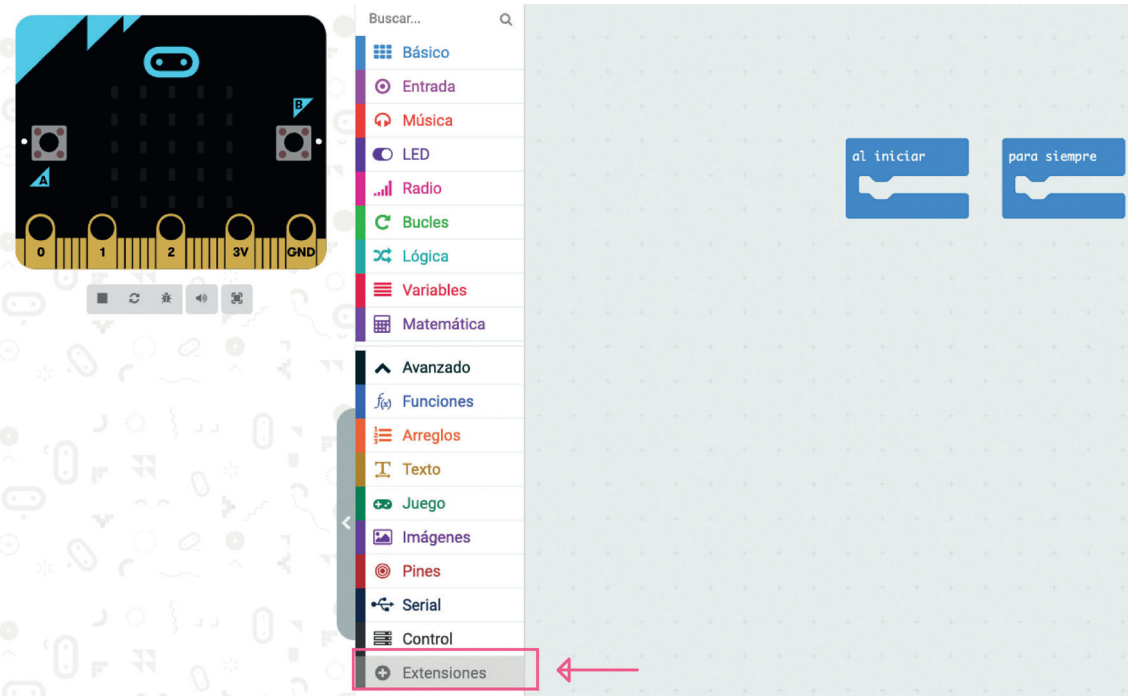
Una librería, o extensión, es un paquete de categorías nuevas que poseen bloques de programación adicionales para el MakeCode, los cuales están diseñados para realizar tareas específicas de los kits que se integran al micro:bit. Por ejemplo, la librería del GamePad tiene bloques que permiten programar tareas y dispositivos electrónicos del GamePad y los elementos que se conectan en él, mientras que la del Building:bit tiene bloques enfocados a programar las tareas y los dispositivos electrónicos del Building:bit y los elementos que se conectan en él.

MakeCode Online

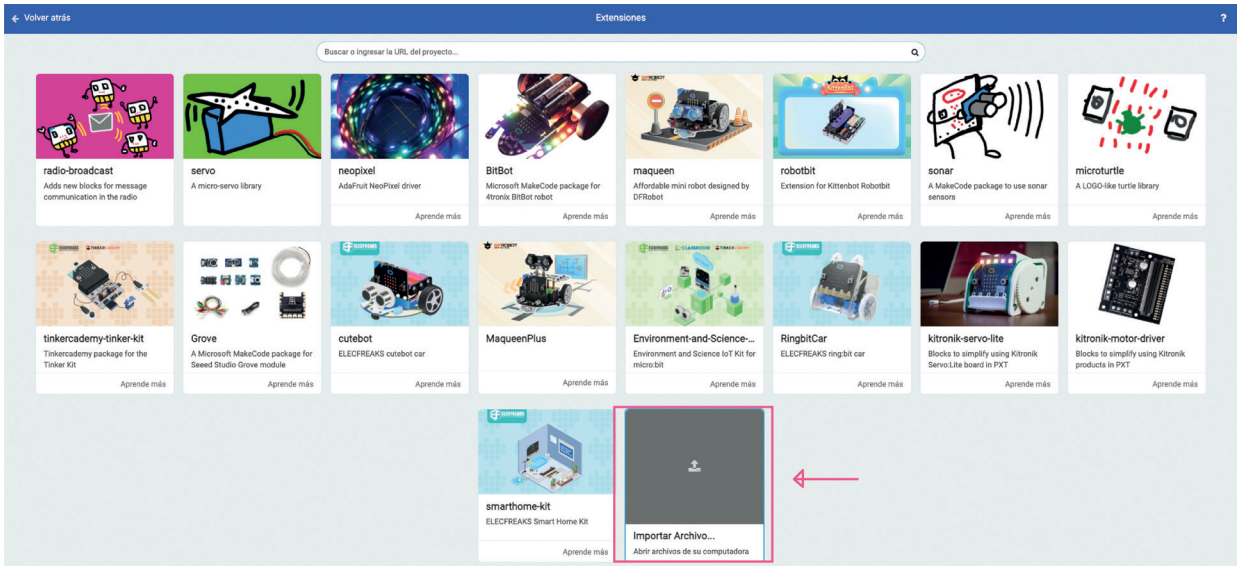
Para cargar las librerías del kit STEM al proyecto teniendo acceso a la versión de MakeCode Online en un navegador, deberá realizar los siguientes pasos:

1. Abra el proyecto donde necesita las librerías, o cree uno nuevo, despliegue las categorías avanzadas y escoja “Extensiones”:

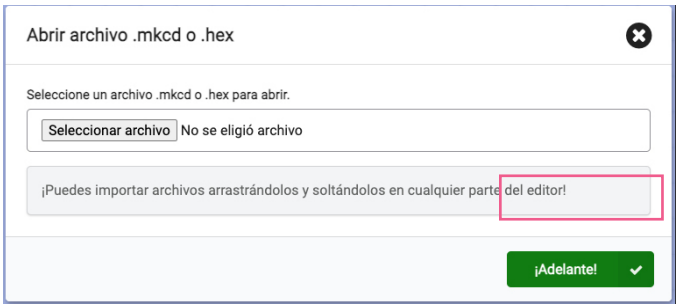


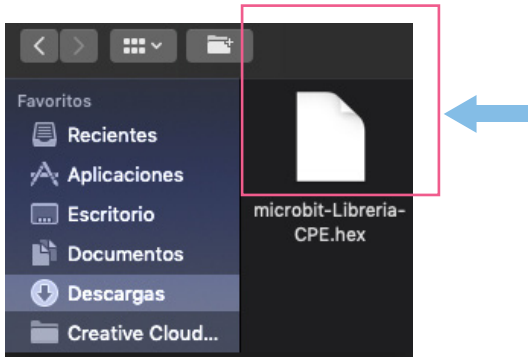


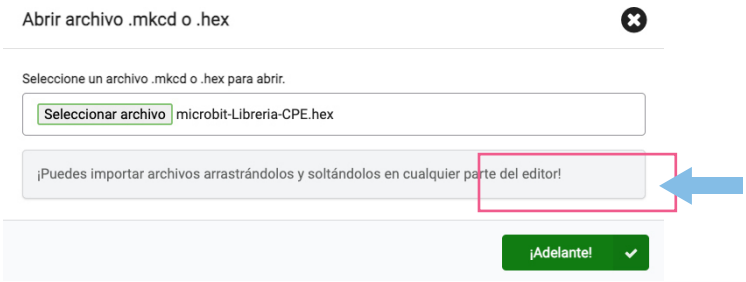
2. Vaya hasta la parte de inferior de la página que se acaba de abrir y elija la opción “Importar archivo...”:

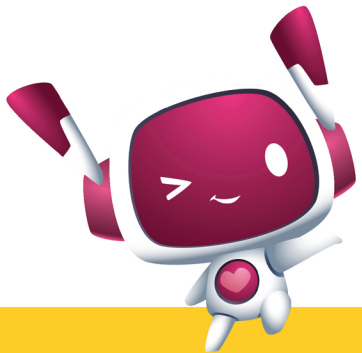


3. Seleccione la opción “seleccionar archivo”, busque en sus documentos el archivo llamado “microbit-Libreria-CPE.hex” si no lo tiene, puede descargarlo en el siguiente enlace: **https://drive.google.com/file/d/15c85yS9LiRxPlXTN9Mrcg3UknD3CcVmv/view?usp=sharing**, selecciónelo y haga clic en “Adelante” para cargarlo al proyecto. Este archivo contiene las librerías tanto del GamePad como del Building:bit incorporadas.

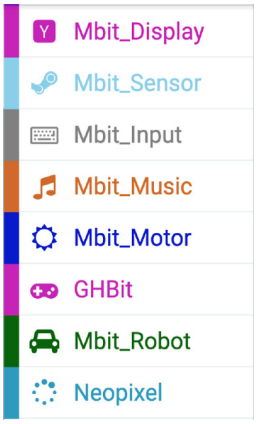








4 ¡Listo! Se debieron haber cargado las siguientes categorías en el proyecto con nuevos bloques destinados a programar el GamePad y el Building:bit:



a. Las categorías del GamePad son:
- GHBit
- Neopixel

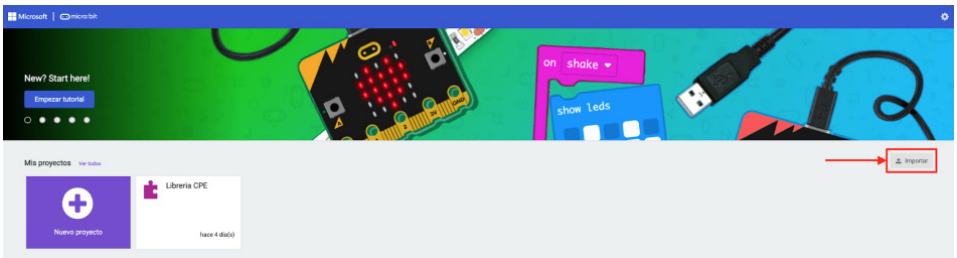
b. Las categorías del Building:bit son:
- Todas las que empiezan con “Mbit”
- Neopixel

MakeCode Offline

Si está trabajando desde la aplicación de MakeCode Offline y necesita cargar la librería, debe realizar los siguientes pasos:

Opción 1: Creando un nuevo proyecto

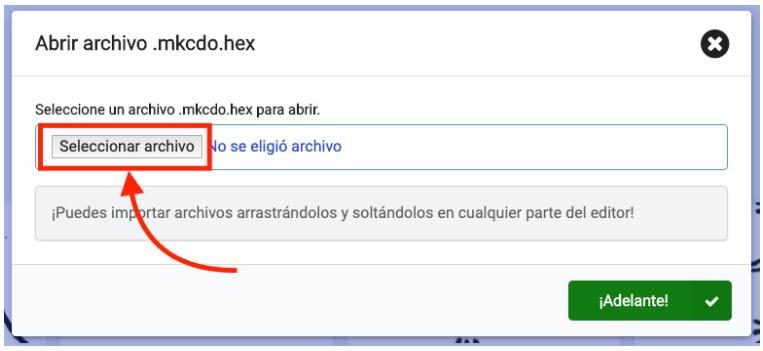
1. Si es la primera vez que va a cargar la librería de robótica en MakeCode Offline, asegúrese de estar conectado a internet.
2. Entre en MakeCode Offline y haga clic en “Importar”.



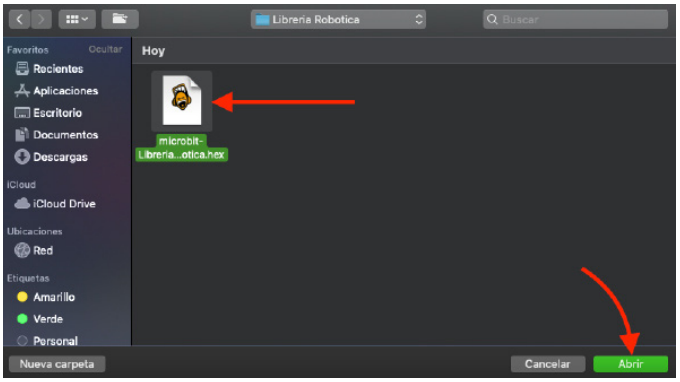
3. Seleccione “Importar archivo...”.



4. Haga clic en “Seleccionar archivo...”.

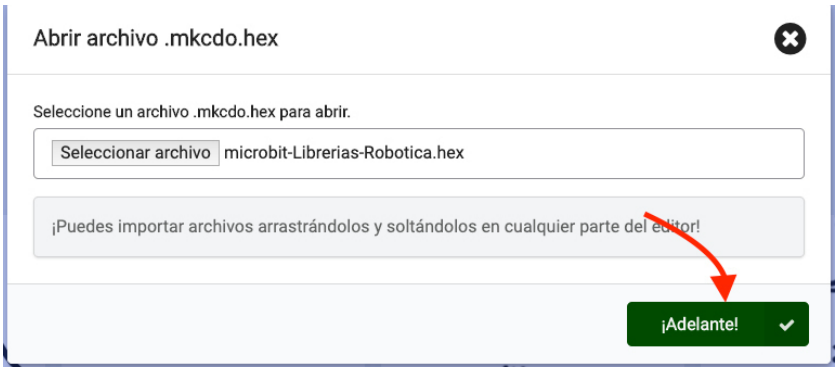


5. Busque y seleccione el archivo “Librerías robótica” (que puede descargar del “Repositorio público”) y por último haga clic en “Abrir”.

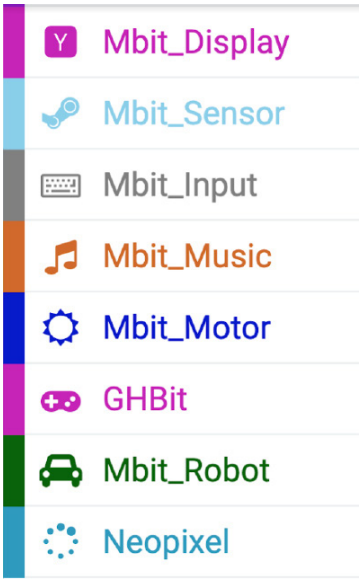




6. Seleccione la opción “¡Adelante!”.



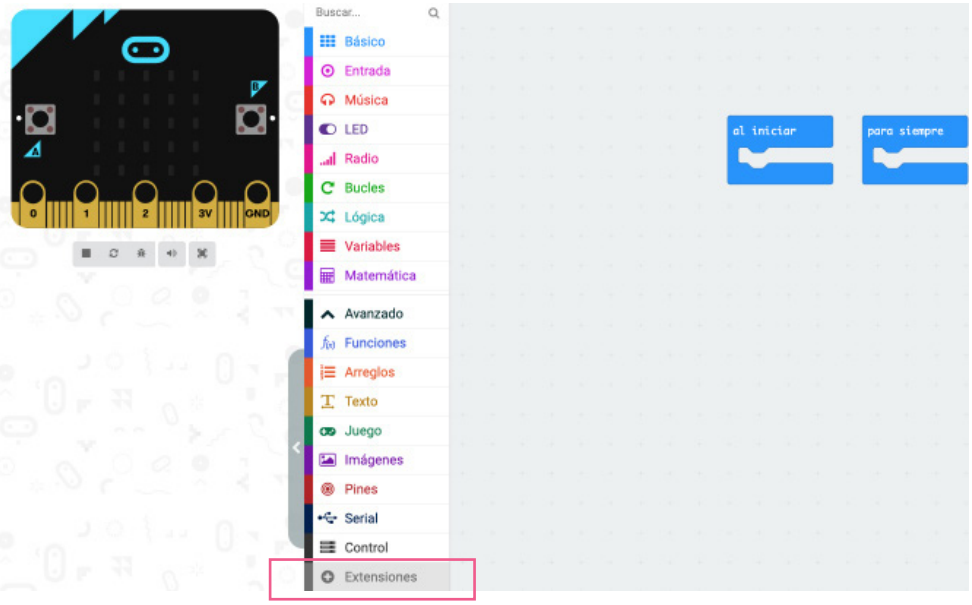
7. Automáticamente, se debe cargar un nuevo proyecto en MakeCode con todas las librerías que necesitará en los cursos.



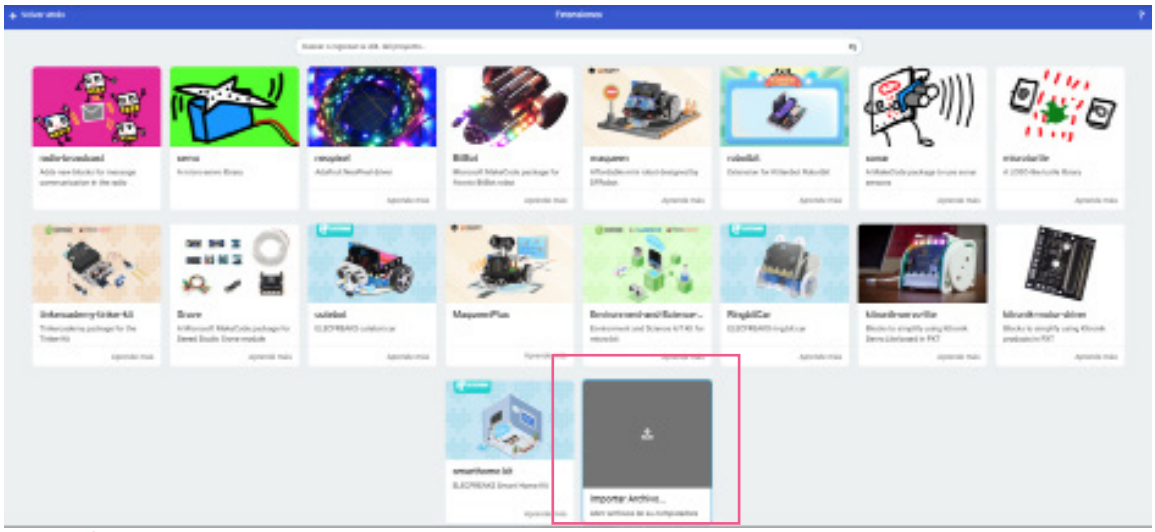
8. Para volver a cargar la misma librería en cualquier otro proyecto, no es necesario tener conexión a Internet.

Opción 2: A partir de un proyecto existente

1. Si es la primera vez que va a cargar la librería de robótica en MakeCode Offline, asegúrese de estar conectado a Internet.
2. Entre al proyecto que creo, haga clic en “Avanzado” y luego seleccione la opción “Extensiones”.

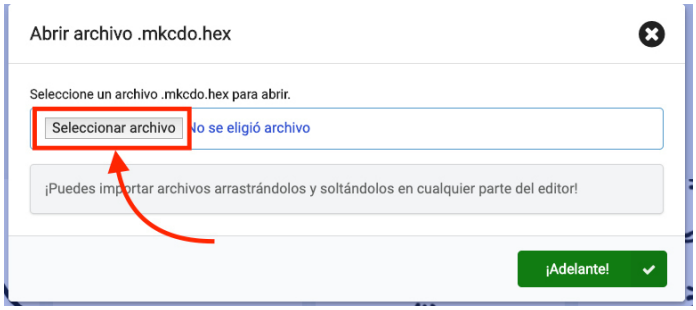


3. Busque la opción “Importar archivo...” y haga clic en ella.

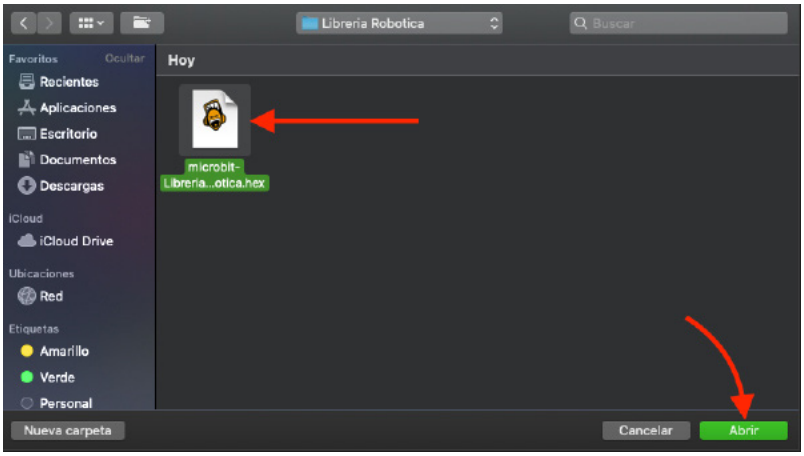




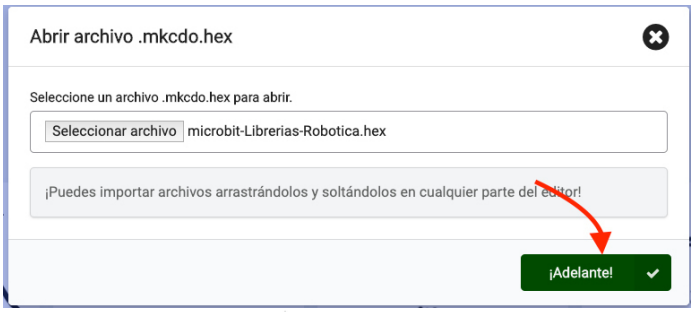
4. Haga clic en “Seleccionar archivo...”.



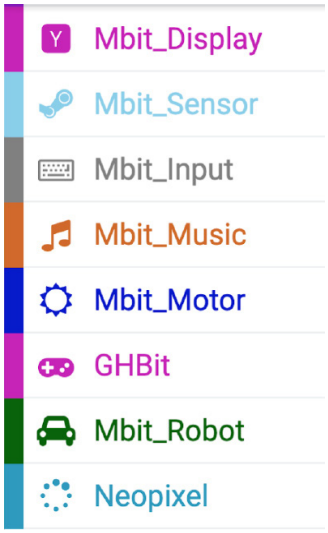
5. Busque y seleccione el archivo “Librerías robótica” (que puede descargar del “Repositorio público”) y haga clic en “Abrir”.



6. Seleccione la opción “¡Adelante!”.



7. Automáticamente, se deben cargar todas las librerías que necesitará en los cursos.



8. Para volver a cargar la misma librería, no es necesario tener conexión a Internet.

Programación de sensores

Los sensores son dispositivos electrónicos que se encargan de convertir una magnitud física, química o ambiental en electricidad, la cual puede ser registrada y medida por el micro:bit para interpretar cada variable externa.

En el Kit de ingeniería STEM que ha recibido tiene una enorme variedad de sensores que puede utilizar en sus proyectos para medir diferentes variables ambientales y crear tecnología aplicable a la vida real. Algunos de estos sensores son:

1. Sensores del micro:bit: Están integrados en la tarjeta del micro:bit y se programan con los bloques de la categoría “Entrada”. Entre estos sensores se encuentran:

a. Sensor de luz: Mide cambios de intensidad lumínica.





b. Sensor de temperatura: mide el nivel de temperatura ambiente.



c. Acelerómetro: mide la aceleración.



d. Brújula: detecta la posición geográfica del norte terrestre.



e. Micrófono: mide cambios en la intensidad sonora del entorno.



2. Sensores integrados en la tarjeta del GamePad: Están integrados en la tarjeta del GamePad y se programan con los bloques de la categoría “GHBit”. Entre estos sensores se encuentran:

a. Sensor de luz: mide cambios de intensidad lumínica.



b. Joystick: mide cambios de inclinación en la palanca del dispositivo.



3. Sensores integrados en la tarjeta del Building:Bit: Están integrados en la tarjeta del Building:bit y se programan con los bloques de la categoría "Mbit_ Robot". Entre estos sensores se encuentran:

a. Sensor de ultrasonido: Mide distancias usando ondas de ultrasonido.



b. Sensor infrarrojo detector de obstáculos: Detecta la presencia de obstáculos frente al robot.



c. Sensores infrarrojos seguidores de línea: Detecta y diferencia los colores blanco y negro para construir un sistema de seguidor de línea.



4. Accesorios del GamePad. Entre los accesorios que incluye el GamePad hay sensores adicionales que se pueden conectar a la tarjeta de expansión de este y programarse con los bloques de su librería. Entre ellos se encuentran:

a. Sensor de ultrasonido: Mide distancias usando ondas de ultrasonido



b. Potenciómetro: Mide ángulos de rotación en su eje.





5. Sensores externos: Como parte del kit STEM se entregó un paquete de sensores externos que se pueden conectar al micro:bit a través de la tarjeta de expansión para sensores y usarlos para realizar diferentes mediciones y aplicaciones tecnológicas. Algunos de sus usos pueden ser: medir la intensidad de luz ambiente, determinar la altura de nivel de agua de un recipiente, censar el nivel de humedad en la tierra, medir la temperatura ambiente, entre otros. Por ejemplo, en la ruta 10 sesión 5 se usaron 3 tipos diferentes de sensores para controlar las condiciones ambientales de una huerta escolar con el fin de optimizar el crecimiento de los cultivos, se usó un sensor de humedad en la tierra para controlar la humedad de la planta, un sensor de luz para regular la cantidad de luz solar que llega a la huerta y un sensor de temperatura para refrescar la huerta cuando la temperatura se eleva demasiado.

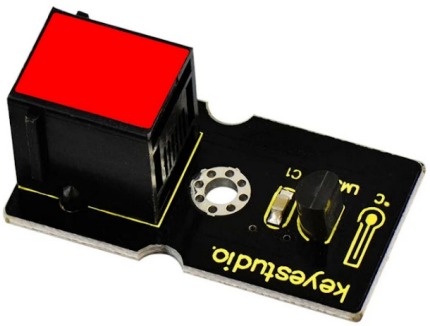
Estos sensores se dividen en dos categorías, analógicos y digitales, donde cada uno de ellos se programa de la siguiente manera:

a. Analógicos: Son aquellos sensores que miden variaciones en magnitudes físicas y tienen la capacidad de determinar el grado de intensidad en que se detecta cada magnitud. Le entregan al micro:bit un código de números enteros que varía entre 0 y 1023. Todos los sensores analógicos se programan de la misma manera, usando el bloque de “lectura analógica” que se encuentra en la categoría “Pines” del MakeCode.

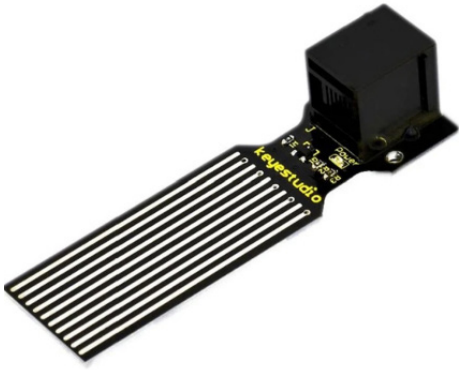


Los sensores analógicos que tiene en su kit STEM son:

a. Sensor lineal de temperatura LM35: Mide el nivel de temperatura ambiente.



b. Sensor de nivel de agua: Mide el nivel de agua en un recipiente.



c. Sensor de luz (fotorresistencia): Mide cambios de intensidad lumínica.



d. Micrófono: Mide cambios en la intensidad sonora del entorno.





e. Sensor de humedad en la tierra: Mide cambios en el nivel de humedad de la tierra.



b. Digitales: son aquellos sensores que detectan cambios bruscos en magnitudes físicas y determinan si se está censando o no un fenómeno físico y le entregan al micro:bit un código de 0 o 1 dependiendo si se detecta o no el fenómeno. Todos los sensores digitales se programan de la misma manera, usando el bloque de lectura digital que se encuentra en la categoría Pines del MakeCode.

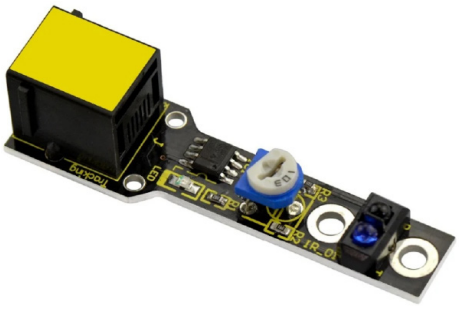


Entre ellos se encuentran:

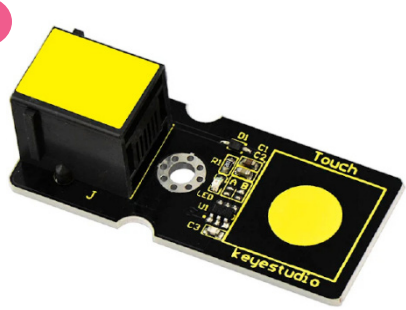
a. Sensor IR seguidor de línea. Detecta y diferencia los colores blanco y negro para construir un sistema de seguidor de línea.

b. Sensor capacitivo touch. Detecta si se está tocando o no la zona táctil del sensor.

a



b



Recuerde revisar en qué número de pin conecta los sensores que va a usar para que lo programe en el bloque de lectura digital/analógica, de tal manera que, si conecta el sensor de temperatura en P1, debe ajustar el menú desplegable del bloque de "lectura analógica pin" para que escoja el pin P1, como se muestra en el ejemplo:



Programación de actuadores

Los actuadores son aquellos elementos que convierten la energía eléctrica, hidráulica o neumática en movimiento y se utilizan para accionar sistemas. En el kit de ingeniería STEM que ha recibido encontrará varios actuadores eléctricos rotacionales, los cuales convierten la electricidad en movimiento rotacional, ya sea para hacer girar su eje a diferentes velocidades, o controlar su ángulo de rotación y sostenerlo.

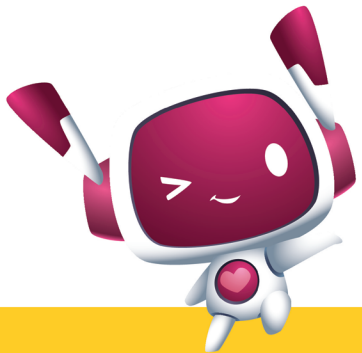
Por ejemplo, el servomotor puede controlar el ángulo de rotación de su eje y sostenerlo en el ángulo deseado; el motor DC puede girar 360° a diferentes velocidades y en uno de dos sentidos: con las manecillas del reloj o en sentido contrario; y el motor paso a paso también puede girar 360°, pero lo hace paso por paso, ya sea dando pasos pequeños o pasos grandes.

En la ruta 9 sesión 4 se usaron 2 servomotores para controlar el movimiento de un prototipo de delfín impreso en 3D con el fin de emular las inclinaciones X, Y del acelerómetro de un micro:bit emisor que, según la problemática, iría pegado a un collar de un delfín real para evaluar su movimiento en el agua. Por otra parte, en la ruta 11, sesión 2 se usó un servomotor y un motor DC para construir una banda transportadora dispensadora de alimentos, donde se usa el servomotor para abrir una puerta y el motor DC para mover la banda, entregar la comida y volver a cerrar la puerta con el servomotor.

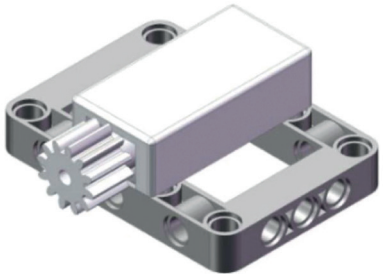
Algunos de los actuadores que puedes encontrar en este kit son:

1. Motores DC.

a. Motor reductor: Este es un motor diseñado para controlar la velocidad

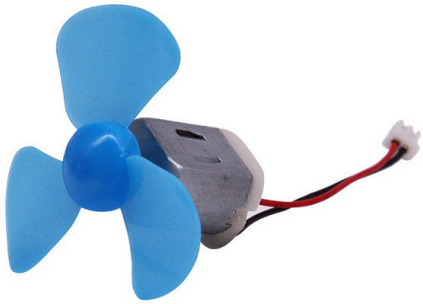


y dirección de giro de su eje. Cuenta con una caja reductora de engranajes conectada a su eje principal, que incrementa su fuerza reduciendo la velocidad, con el fin de controlar los diferentes modelos de robots que se pueden construir con el Building:bit sin perder fuerza.



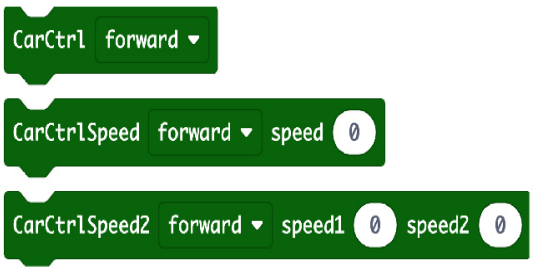


b. Motor DC con ventilador. Este es un motor diseñado para controlar la velocidad y dirección de giro de su eje. Gira más rápido que el motor reductor y tiene menos torque, debido a que no cuenta con una caja reductora. Está diseñado para usarse en aplicaciones que no requieran mucha fuerza, como ventiladores o similares.



Ambos motores se programan de la misma manera y, dependiendo de dónde los conecte (ya sea en la tarjeta de expansión del Building:bit o del GamePad) deberá usar alguno de los siguientes bloques de programación:

a. Conectado en el Building:bit. Puede usar cualquiera de los 3 bloques que se muestran a continuación, donde el primero hace girar el motor a su máxima velocidad, el segundo permite controlar la velocidad de giro del motor (independiente del puerto de conexión donde se conecte), y el tercero permite controlar la velocidad de giro del motor dependiendo de en qué puerto de conexión se conecte, el derecho o el izquierdo. En el espacio blanco donde está el número cero “0” deberá escribir una velocidad entre 0 (detenido) y 255 (máxima velocidad), y en el menú desplegable deberá seleccionar la dirección de desplazamiento.



b. Conectado en el GamePad. Para programar un motor DC conectado al GamePad debe usar el siguiente bloque, donde se puede ajustar la velocidad del motor en un rango de 8 niveles (primero a octavo) siendo el primero (first) el de más baja velocidad y el octavo (eighth) el de más alta velocidad.



2. Servomotores. Este es un motor diseñado para rotar su eje a un determinado ángulo y sostenerlo. A diferencia de los motores DC, este motor no controla su velocidad ni dirección de giro, sino que controla la posición de su eje. Para programarlo debe usar uno de los siguientes bloques, dependiendo de en qué tarjeta de expansión lo conecte:

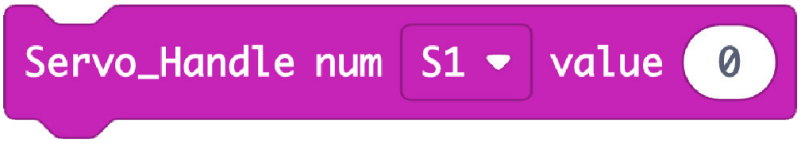


a. Conectado en el Building:bit. Si conectas el servomotor en J2 deberá programar el bloque "Servo_Car num" de la categoría "Mbit_Robot" con la opción S1, si lo conecta en J3 deberá usar la opción S2, y si lo conecta en J4 deberá usar la opción S3. En el espacio blanco donde está el número cero "0" deberá escribir un ángulo entre 0° y 180°:

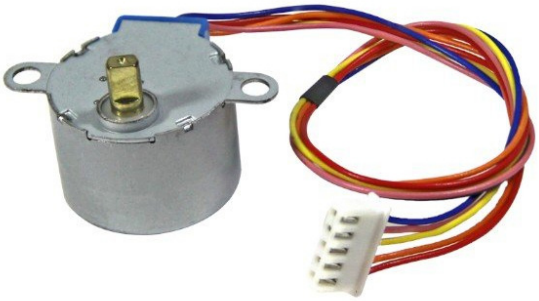




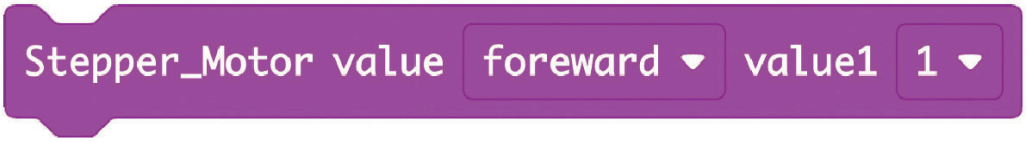
b. Conectado en el GamePad. Si conecta el servomotor en J5 deberá programar el bloque "Servo_Handle num" de la categoría "GHBit" con la opción S1, si lo conecta en J6 deberá usar la opción S2, si lo conecta en J7 deberá usar la opción S3, y si lo conecta en J8 deberá usar la opción S4. En el espacio blanco donde está el número cero "0" deberá escribir un ángulo entre 0° y 180°:



3. Motor paso a paso. Este es un motor diseñado para rotar su eje cada cierta cantidad de grados, dependiendo de su estructura interna y de cómo se programe. A diferencia de los motores DC, su eje no rota libremente, sino que lo hace "paso a paso" cambiando su posición de un grado a otro.



Para programarlo deberá usar el siguiente bloque, donde:



a. Use el primer menú desplegable para escoger si el motor se mueve hacia adelante (forward), hacia atrás (reversa), o si se detiene (stop).

b. Use el segundo menú desplegable para escoger cada cuantos "pasos" se mueve el motor, ya sea dando pasos pequeños (1), pasos medianos (2) o pasos grandes (3).

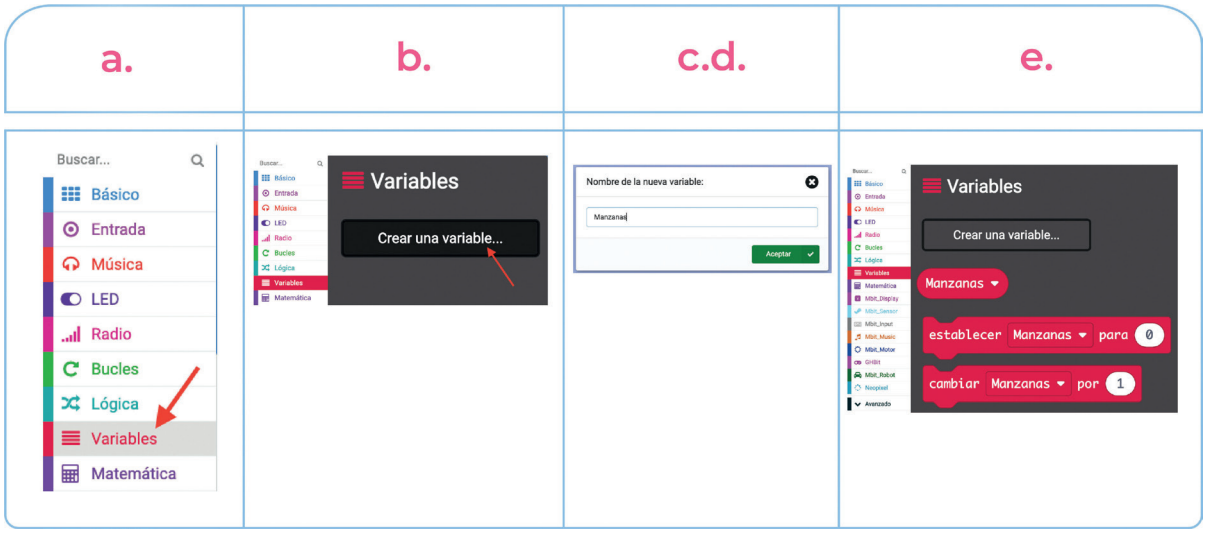
¿Cómo crear y programar variables?

Las variables son espacios de memoria de una tarjeta programable en las que se puede almacenar información de todo tipo como texto, números, sonidos, entre otros, para usarlos en cualquier parte del algoritmo, modificarlos, borrarlos o cambiarlos según como sea necesario.

El micro:bit tiene una categoría especial para crear variables y programarlas en MakeCode, la cual funciona de la siguiente manera:

1. Crear variables

a. Ingrese en el proyecto o cree uno nuevo y seleccione la categoría "Variables".
b. Haga clic en "Crear una variable...".
c. Escriba el nombre de su variable, una palabra relacionada con el tipo de información que desee guardar en ella, por ejemplo, "Manzanas" para indicar la cantidad de manzanas que se registran en el programa.
d. Seleccione la opción "Aceptar".
e. En la categoría "Variables" debieron haber aparecido nuevos bloques, como los que se muestra en el ejemplo:



2. Programar variables

Las variables creadas se pueden programar de 3 maneras distintas, según el bloque que se utilice:

a. Mostrar el valor de la variable Con este bloque se puede mostrar el valor almacenado en una variable o guardarlo en otra.



b. Asignarle un valor a una variable. Con este bloque se le puede asignar un nuevo valor a una variable en cualquier instante y en cualquier lugar del programa.



c. Cambiar el valor de una variable. Con este bloque se puede incrementar o reducir gradualmente el valor de una variable.



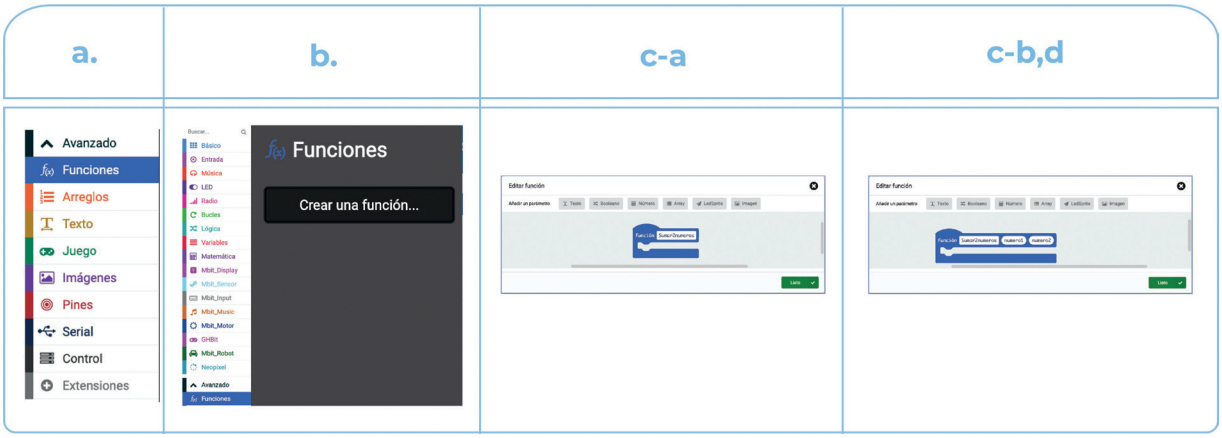
¿Cómo crear y programar funciones?

Las funciones en la programación son como pequeños fragmentos de código que realizan una tarea específica, la cual suele repetirse en diferentes partes de un mismo algoritmo. Las funciones permiten reducir la longitud de un algoritmo; al guardar toda una secuencia de líneas de código dentro de ellas para solo llamarlas (usando el nombre de la función) en el programa sin tener que reescribir todas las líneas de nuevo.

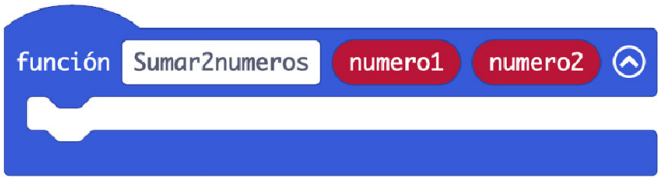
El micro:bit tiene una categoría especial para crear funciones y programarlas en MakeCode, la cual funciona de la siguiente manera:

1. Crear funciones

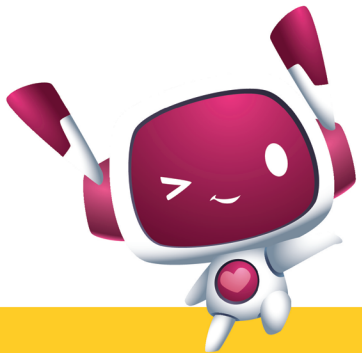
a. Ingrese en el proyecto o cree uno nuevo, haga clic en “Avanzado” y clic en la categoría “Funciones”.
b. Seleccione “Crear una función...”.
c. En la ventana que se acaba de abrir debe realizar las siguientes acciones:
a. Dale un nombre a tu función. Donde dice “doSomething” escriba el nombre de la función, algo relacionado con la tarea que va a cumplir dentro del programa, por ejemplo “Sumar 2 números”.
b. Defina los parámetros de entrada de la función. Al hablar de “parámetros” se hace referencia a los datos de entrada que recibirá la función, con los cuales va a realizar las tareas que se programarán dentro de ella. Haga clic en el tipo de parámetro de entrada para determinar si se ingresarán letras, números, texto, imágenes, u otro tipo de información a la función, y luego escriba un nombre para cada parámetro. Para el ejemplo, se definen los parámetros numéricos “Número 1” y “Número 2”.
d. Seleccione la opción “Listo” para crear la función que acaba de diseñar en el paso anterior.



e. Como resultado se debe crear un bloque de ejecución similar al siguiente:



f. ¡La función está lista para programarse!



2. Programar funciones

a. Una vez creada la función, se puede programar todo tipo de tareas dentro de ella usando cualquiera de los bloques de MakeCode, como si se tratara de un bloque de ejecución cualquiera. En el ejemplo a continuación, se pueden ver los números ingresados usando la instrucción de "Mostrar número" y los parámetros de entrada "numero1" y "numero2" (estos parámetros solo existen dentro de la función, no se pueden usar fuera de ella).



b. En la categoría "Funciones" van a aparecer 2 bloques de programación: "devuelve" y "llamada", donde el primero se encarga de entregar un resultado a partir del algoritmo programado en la función, y el segundo, llama la función en cualquier parte del algoritmo para que se ejecute.





c. Luego se debe usar el bloque "devuelve" para entregar como resultado la suma de los números.

a. Nota: El bloque "devuelve" solo se puede usar dentro de una función.

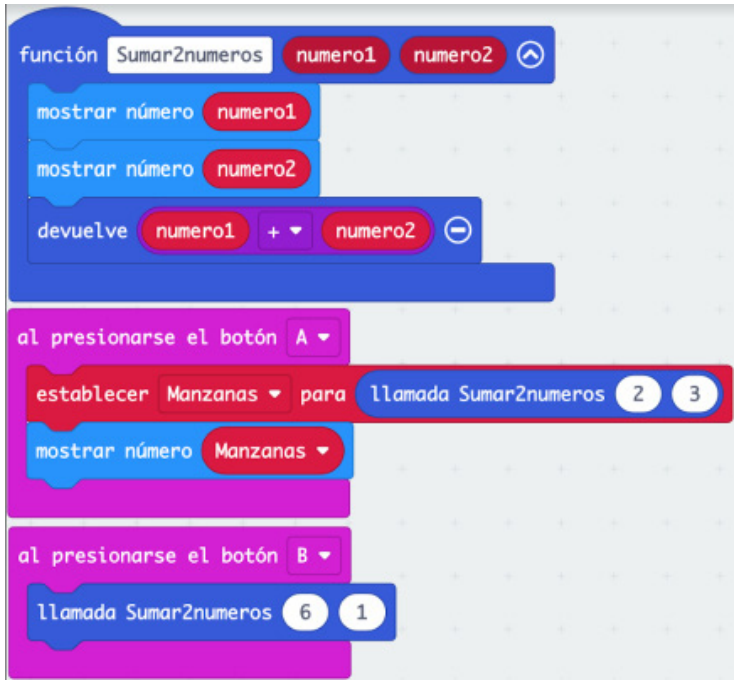


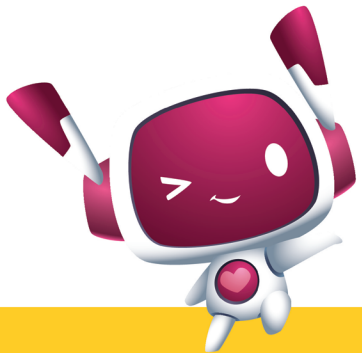
d. No es obligatorio usar el bloque “devuelve”, este bloque se usa principalmente cuando se realizan operaciones matemáticas y se desea entregar el resultado de una operación como salida de la función. Es posible programar todo tipo de funciones sin usar el bloque “devuelve”.

e. Cuando se usa el bloque “devuelve”, aparece un nuevo bloque en la categoría “Funciones” con forma ovalada y el nombre “llamada”. Este bloque realiza la misma función que el bloque de instrucción “llamada”, solo que entrega directamente el resultado de la función, lo que permite guardarlo en una variable; para el ejemplo, entregaría la suma de los dos números.



f. En el ejemplo, se guarda el resultado de la función “Sumar2numeros” en una variable y luego se muestra la variable en la pantalla del micro:bit al presionar el botón A, mientras que al presionar el botón B solo se llama la función para que muestre los valores ingresados.





a. Algoritmo de ejemplo:
https://makecode.microbit.org/_WUARDReyRVxP

¿Cómo programar el micro:bit físico?

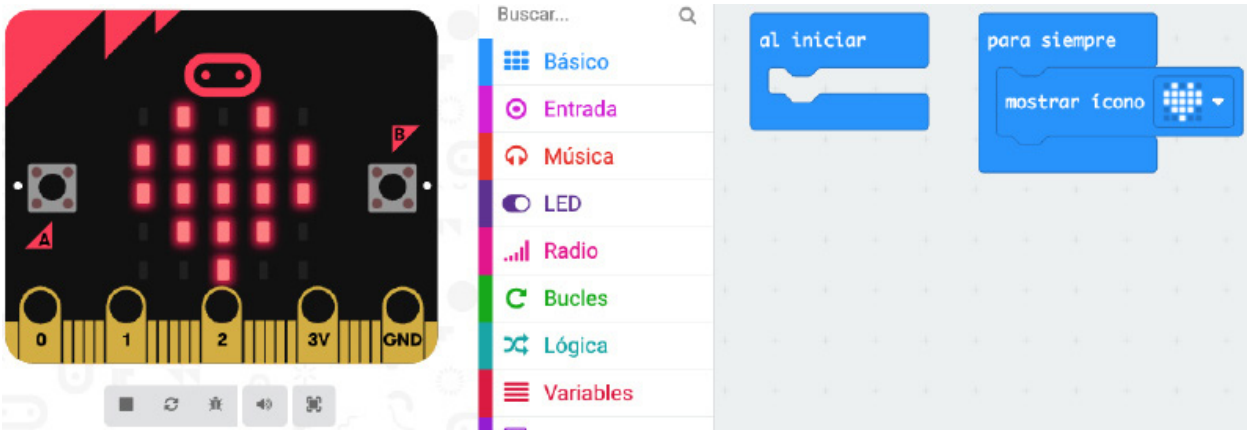
Existen tres formas de programar el micro:bit físico con el algoritmo realizado. La primera es a través del navegador Google Chrome, donde se puede enlazar el micro:bit al navegador y hacer una programación directa con solo presionar un botón. La segunda es a través del explorador de archivos, donde se debe programar el micro:bit como si fuera una memoria USB arrastrando el algoritmo hasta él, y la tercera es a través de la aplicación MakeCode Offline, y solo deberá hacerlo está trabajando directamente en esta aplicación, de lo contrario realice las formas 1 o 2.

Google Chrome

Para enlazar y programar el micro:bit con el navegador Google Chrome debe asegurarse de tener una versión reciente del navegador y luego realizar los siguientes pasos:

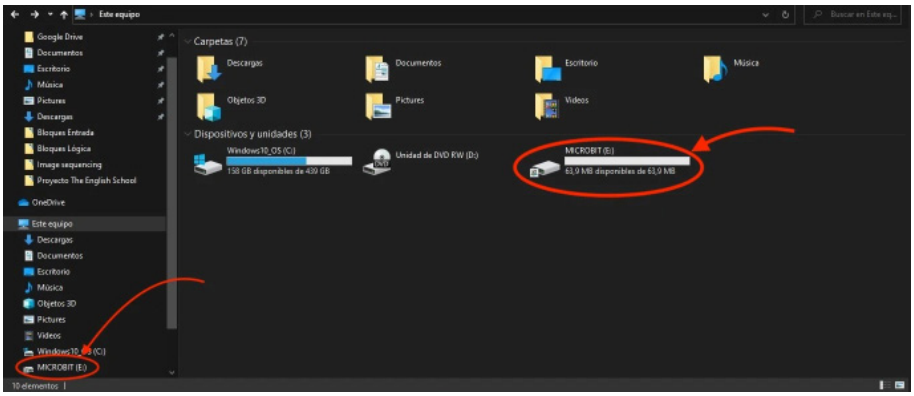
1. Ingrese a Google Chrome, posteriormente ingrese a la plataforma de programación MakeCode y abra el proyecto que desee programar en el micro:bit. Para el ejemplo, se usará un proyecto en el que se muestra un corazón.

https://makecode.microbit.org/#

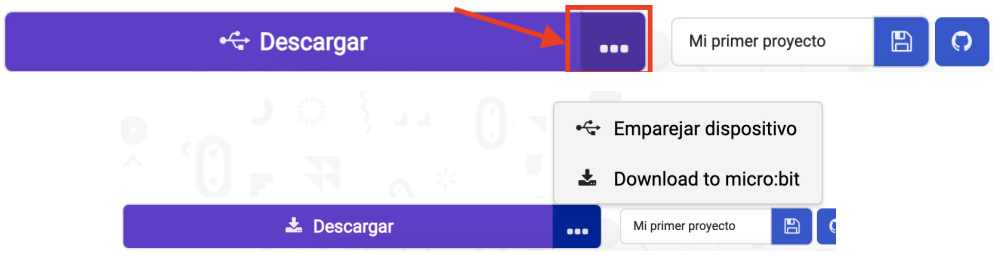


2. Conecte el micro:bit al computador mediante el cable USB, y asegúrese de que el computador lo reconoce como un dispositivo USB.

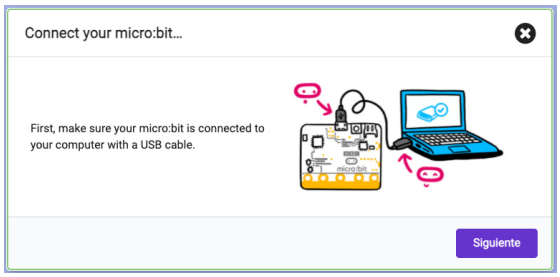


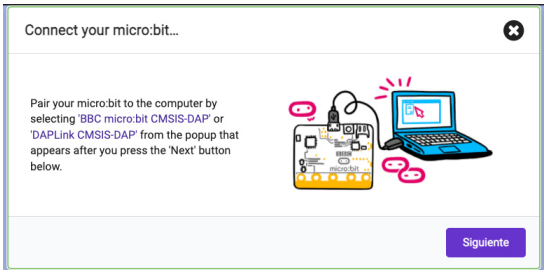


3. Dentro del proyecto, haga clic en los tres puntos y elija la opción “Emparejar dispositivo”.

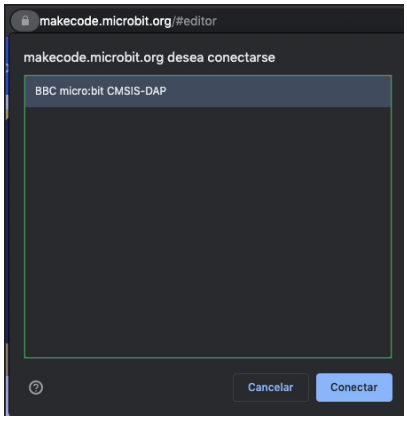


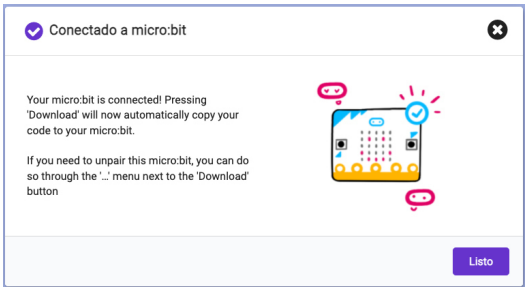
4. Haga clic en “siguiente” y “siguiente”, escoja el micro:bit que aparece en la ventana emergente, haga clic en conectar y luego haga clic en “Listo”. ¡El micro:bit ha quedado emparejado al navegador!



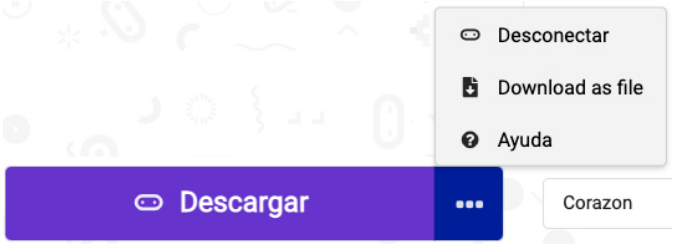






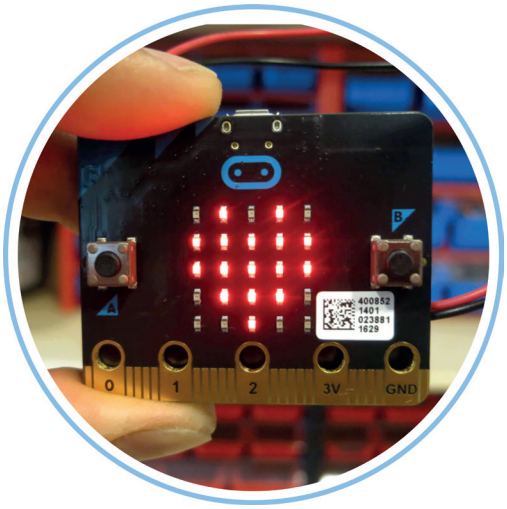


5. Si el micro:bit se emparejó correctamente, el botón de descargar debe aparecer con el logo del micro:bit y, al hacer clic en los 3 puntos, debe aparecer la opción de “desconectar”.



6. Una vez enlazado el micro:bit al computador, debe hacer clic en el botón “Descargar” y asegurarse de que el micro:bit se mantenga conectado al computador hasta que el proceso finalice. El algoritmo se descargará automáticamente en el micro:bit y al terminar, el botón de descarga dirá “Downloaded” y el micro:bit comenzará a ejecutar el programa.



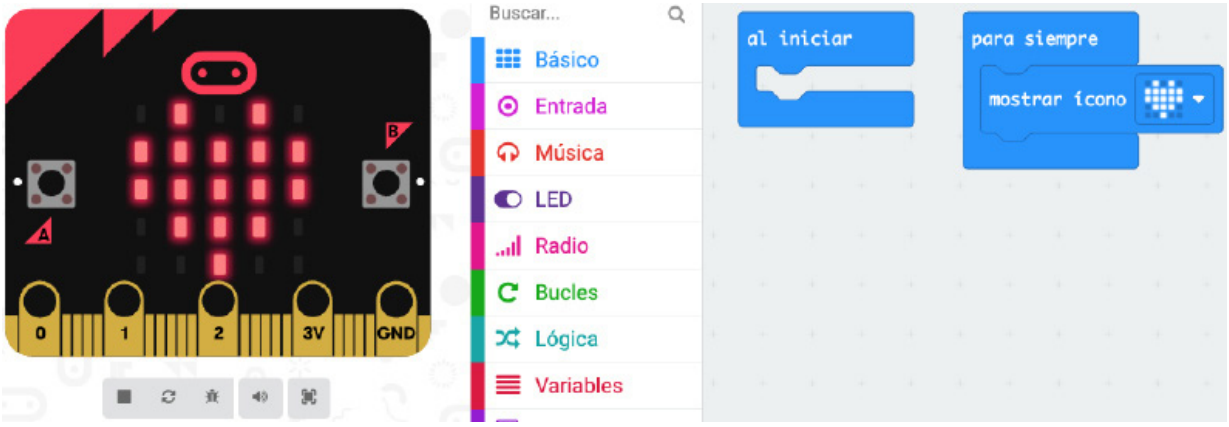


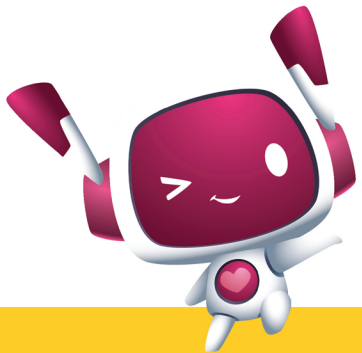
Archivo .hex

Si no cuenta con el navegador Google Chrome actualizado o su dispositivo no lo tiene instalado, puede programar el micro:bit como si fuera un dispositivo USB arrastrando el algoritmo de su proyecto al micro:bit, siguiendo los pasos que se muestran a continuación:

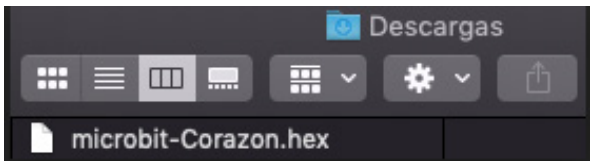
1. Abra el navegador, ingrese a la plataforma de programación MakeCode y abra el proyecto que desee programar en el micro:bit. Para el ejemplo, se usará un proyecto en el que se muestra un corazón.

https://makecode.microbit.org/#

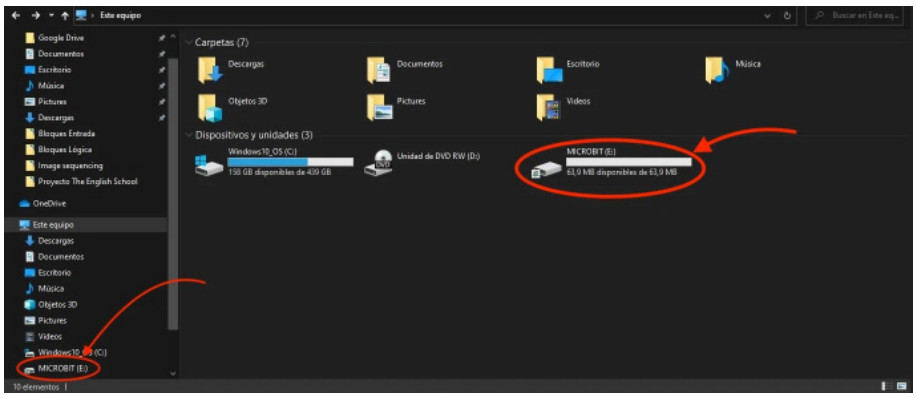




2. Haga clic en el botón “Descargar”. Se guardará en la carpeta de descargas un archivo con extensión “.hex” y el nombre “microbit-“ seguido por el nombre del proyecto.







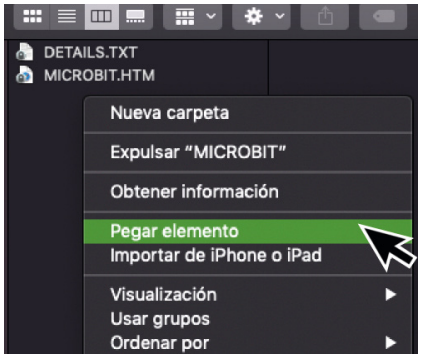
3. Conecte el micro:bit al computador mediante el cable USB y asegúrese de que el computador lo reconoce como un dispositivo USB.

4. Haga clic sostenido en el archivo que se acaba de descargar, arrástrelo hasta el micro:bit y suéltelo, o haga clic derecho en el archivo, cópielo, abra el micro:bit como si abriera una USB y pegue el archivo. Con cualquiera de estos dos métodos programará el micro:bit físico.

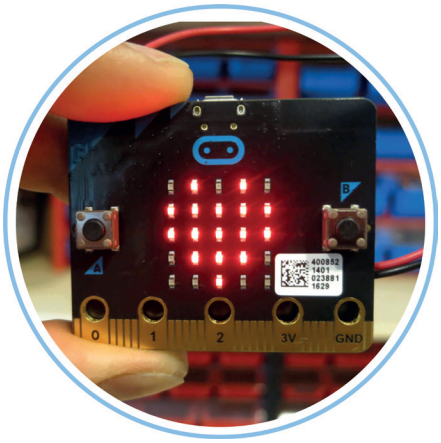
a. Método 1 – Arrastrar el archivo



b. Método 2 – Copiar y pegar el archivo

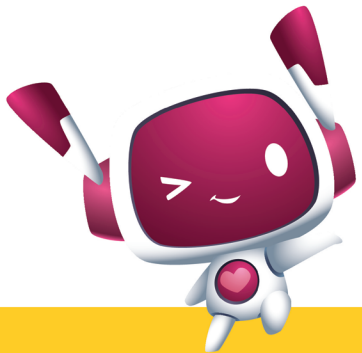


5. Mantenga conectado el micro:bit al computador hasta que finalice el pegado del documento. Sabrá que terminó, cuando el led naranja del micro:bit deje de parpadear y cuando el computador indique que el archivo se ha copiado correctamente. El micro:bit comenzará a ejecutar el programa y podrá desconectarlo para usarlo con las pilas AAA.



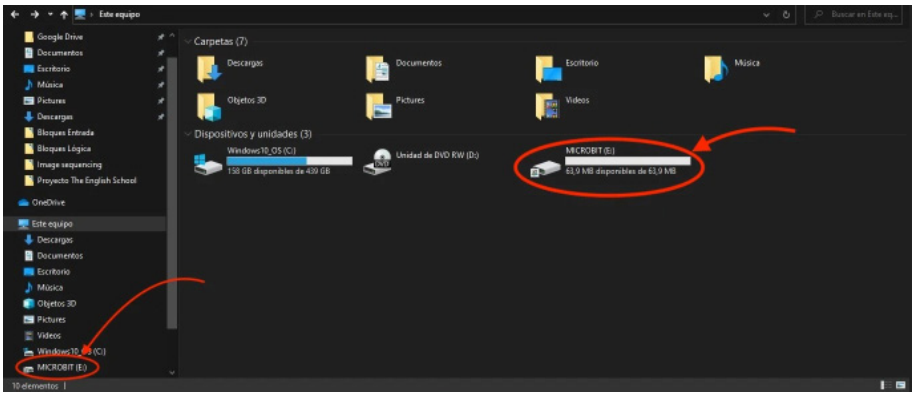
MakeCode Offline

Si está trabajando desde la aplicación de MakeCode Offline, siga los pasos que se mencionan a continuación para descargar los proyectos en la tarjeta programable micro:bit.

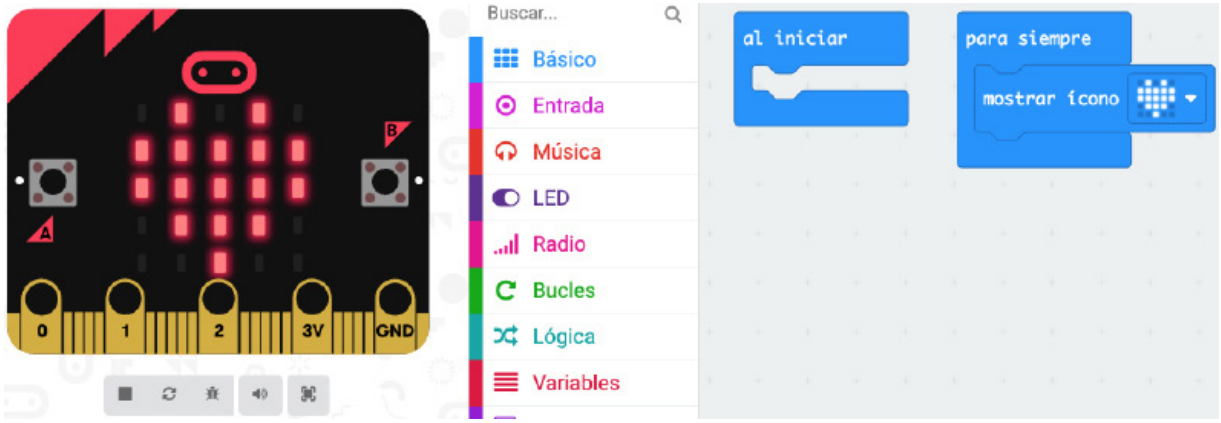


1. Conecte el micro:bit al computador y espere a que lo reconozca como si fuera una memoria USB.

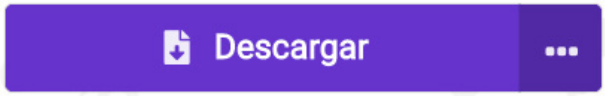




2. Ingrese al proyecto que desea descargar en la tarjeta. Asegúrese de estar dentro de la aplicación MakeCode Offline donde realizó el proyecto.

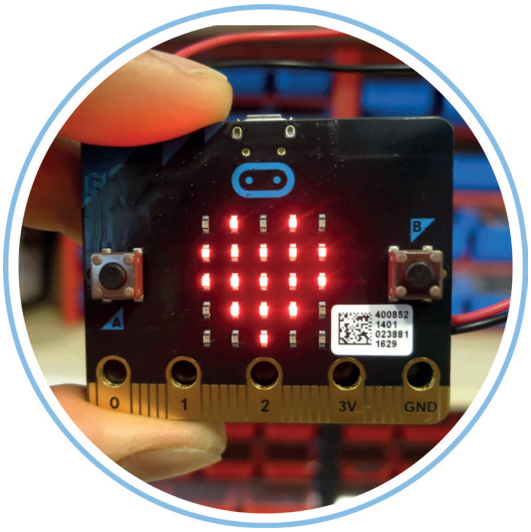


3. Haga clic en “Descargar”.



4. El micro:bit se debe programar directamente, tal y como sucede con la programación desde Google Chrome, con la diferencia de que no es necesario emparejar el micro:bit a la aplicación, él se empareja automáticamente una vez que está conectado al computador.

5. Mantenga conectado el micro:bit al computador hasta que finalice la programación. Sabrá que terminó, cuando el led naranja del micro:bit deje de parpadear, o cuando el computador indique que el archivo se ha copiado correctamente o que el micro:bit se ha desconectado sin previo aviso. El micro:bit comenzará a ejecutar el programa y podrá desconectarlo para usarlo con las pilas AAA.



Programación en Arduino – Generalidades

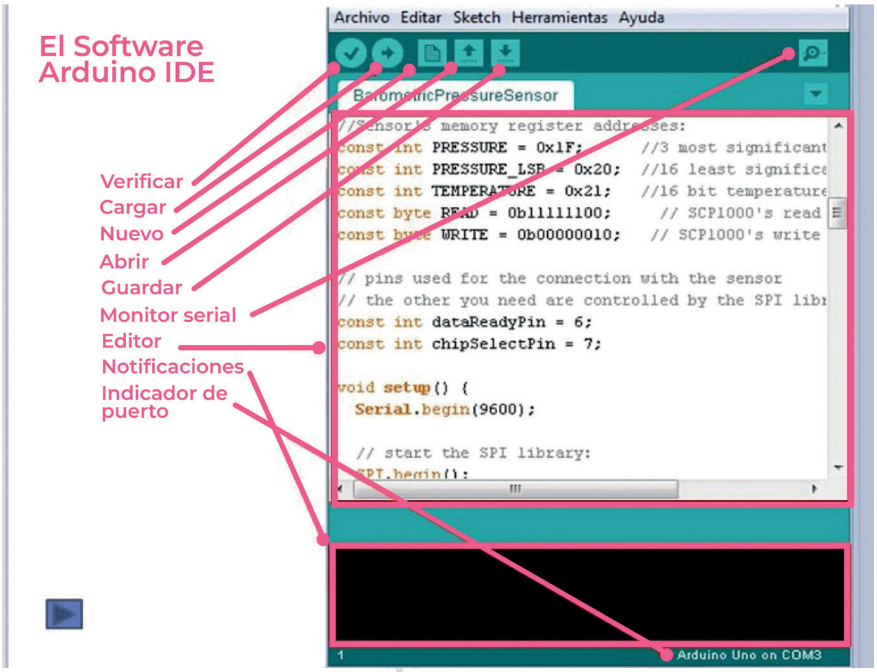
La programación en Arduino consta de líneas de código basadas en C++ donde se requiere de un conocimiento mínimo de la sintaxis y el lenguaje de programación Arduino para programar de forma correcta las diferentes tareas que se pueden realizar con la tarjeta. Esta tarjeta programable cuenta con las herramientas necesarias para compilar y quemar el programa realizado en el microcontrolador del Arduino, todo mediante el software de programación de la tarjeta, el cual es de acceso libre y se puede descargar en la página oficial de Arduino o del Gestor de Contenidos en la dirección **"Repositorio Técnico NSTs. Fase 3 > Instaladores".**



Partes del software Arduino

El software de programación Arduino tiene las siguientes partes:

- Verificar: con este botón se compila el programa. La compilación permite identificar si las instrucciones de programación están correctamente escritas o si existe algún error de sintaxis, no asegura que la lógica de programación sea correcta.
- Cargar: con este botón se carga el programa al Arduino, solamente se puede cargar después de verificar que la sintaxis es correcta.
- Nuevo: con este botón se puede crear un nuevo proyecto.
- Abrir: con este botón se puede abrir un proyecto de Arduino existente.
- Guardar: con este botón se guarda el algoritmo realizado.
- Monito serial: con este botón se abre la ventana del monitor serial, la cual permite visualizar datos registrados por el Arduino mediante sensores, información recibida por serial, o escribir texto que se haya programado para mostrarse por el puerto serial.
- Editor: en esta sección se escriben todas las instrucciones de programación.
- Notificaciones: en esta sección se muestra toda la información relacionada con la compilación del algoritmo, se especifican los errores de sintaxis que puedan existir y si la carga del programa al Arduino fue correcta.
- Indicador de puerto: en esta sección se especifica el puerto USB en el que se conectó el Arduino.



Estructura de un sketch de programación

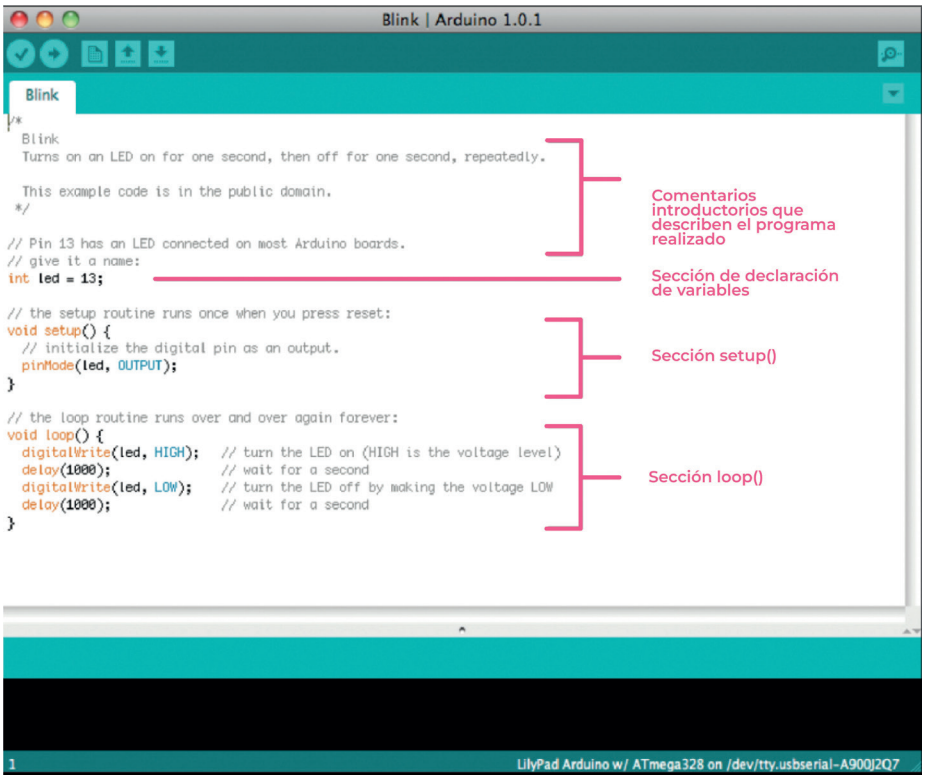
Un sketch es el nombre que se le da a un proyecto de Arduino y tiene la extensión ".ino". El sketch debe estar siempre dentro de una carpeta o directorio con el mismo nombre que el proyecto para que pueda funcionar. Cada vez que se crea un nuevo proyecto, el Arduino crea una carpeta con el mismo nombre del Sketch de forma automática para asegurar su correcto funcionamiento.
El sketch tiene una estructura básica compuesta por dos elementos obligatorios que encierran las instrucciones de programación de cualquier algoritmo que se realice:

- **Void setup().** En esta sección se escriben las instrucciones que definen qué puertos del Arduino serán entradas y salidas, se inicializa la comunicación serial, se definen los estados iniciales de los pines del Arduino y se inicializan otros elementos de programación como las variables. Las instrucciones de programación escritas dentro de esta sección se ejecutan una única vez al iniciar el programa.
- **Void loop().** En esta sección se escriben las instrucciones de programación que se ejecutan de forma indefinida, repitiéndose ininterrumpidamente, se usa para escribir las instrucciones que realizarán las tareas de automatización deseadas.

```
1  void setup() {
2   // put your setup code here, to run once:
3  }
4
5  void loop() {
6   // put your main code here, to run repeatedly:
7  }
```

También se pueden escribir líneas de programación fuera de las dos estructuras mencionadas, antes del "void setup()", con el fin de declarar variables globales que se puedan usar a lo largo de todo el algoritmo (útiles para guardar información y verla o modificarla en cualquier momento), para asignarle un nombre a los pines del Arduino que se van a utilizar, para llamar librerías las cuales contienen nuevas instrucciones de programación que facilitan la realización de tareas específicas, o para escribir comentarios que ayuden a recordar cómo funciona el programa realizado.





Aspectos básicos de programación en Arduino

Los lenguajes de programación requieren de una sintaxis adecuada para poder asegurar una correcta comunicación con el compilador, ya que el compilador es el encargado de leer el código y traducirlo a lenguaje de máquina que es el lenguaje que entienden los microcontroladores. La sintaxis no puede tener ningún error, de lo contrario el programa no se podrá ejecutar.

En el caso de la programación con Arduino, es importante usar el punto y coma (;) al final de cada instrucción de programación para que el programa interprete que la instrucción ha finalizado. También se deben cuidar las mayúsculas y minúsculas para que el programa entienda las instrucciones escritas, y se debe verificar que los corchetes ({) que se abran también se cierren para identificar el inicio y fin de funciones o secciones de programación como setup y loop.

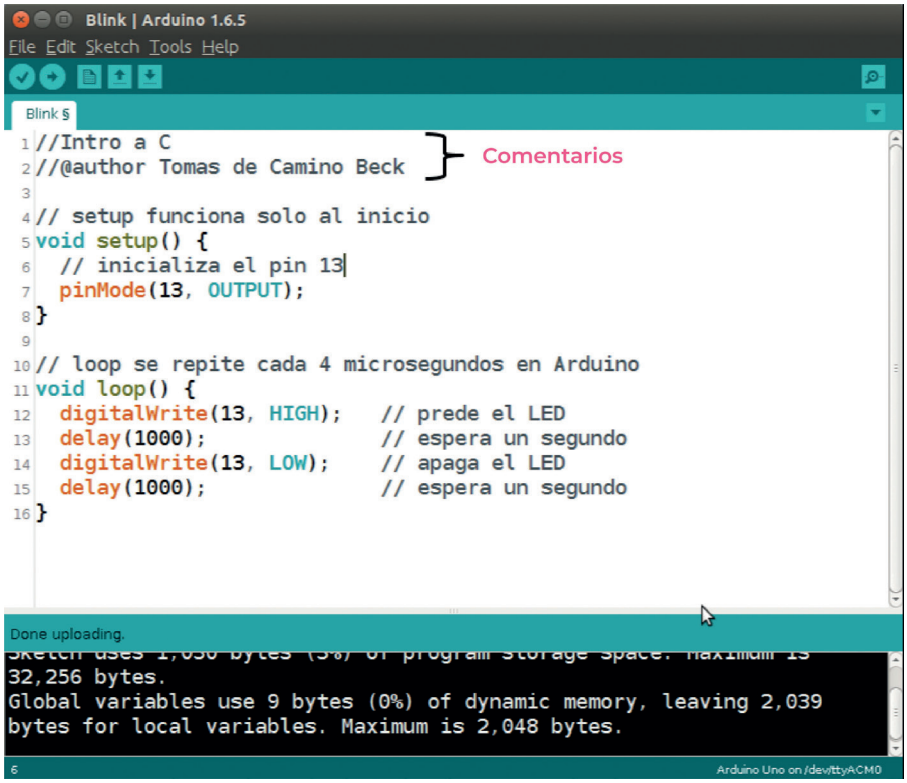
También es importante tener en cuenta que hay palabras reservadas en Arduino que no se pueden usar para fines distintos que para las que fueron

creadas, como HIGH, LOW, true, false, entre otros.

Los elementos básicos de programación en Arduino son:

- **Comentarios.** Son fragmentos de texto escritos por el programador que no representan ninguna instrucción de programación, sino que se usan para describir el comportamiento de alguna variable, el funcionamiento del algoritmo, o algún otro dato relevante del programa. Constan básicamente de notas de texto que sirven como guía al momento de leer el algoritmo.

  - **Una línea de comentario.** Para escribir solo una línea de comentario se deben poner dos "slash" (//) y a continuación escribir el texto. El comentario finalizará cuando se agregue un salto de línea.

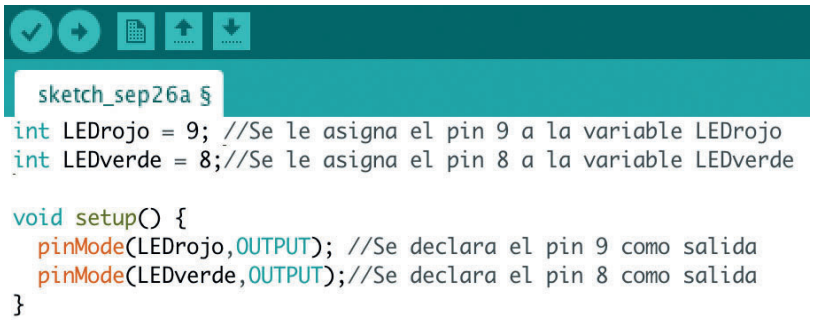


  - **Varias líneas de comentario.** Para escribir varias líneas de texto de comentario se debe marcar el inicio y el final de los comentarios usando un "slash" y un asterisco, de tal manera que abrir el paquete de comentarios se representa con /* y cerrarlo se representa con */.





- **Asignación de pines.** Todos los pines del Arduino se pueden asignar a un nombre o variable al inicio del código, con el fin de programarlos usando el nombre asignado en lugar de usar el número del pin, lo que le permitirá al programador evitar la necesidad de recordar todos los números de pines usados y recurrir únicamente al nombre asignado a cada pin. Esto se realiza escribiendo el tipo de información que se almacenará en la variable (en este caso se debe asignar como un valor entero int), definiendo un nombre para la variable (ten en cuenta que el algoritmo distingue mayúsculas y minúsculas) seguido del signo igual (=) y el número del pin seleccionado, como se muestra en el ejemplo, donde el pin 9 se asignó a la variable “LEDrojo” y el pin 8 se asignó a la variable “LEDverde”, de tal manera que resultará muy sencillo recordar qué pin controla el led rojo y cuál controla el led verde.



- **Escritura y lectura de pine.** Cada vez que se desea programar alguno de los pines del Arduino, sea digital o analógico, se debe definir desde el inicio del programa qué pines serán de entrada y cuáles de salida. Esta asignación no se puede cambiar más adelante en el programa y es fundamental realizarla, ya que sin ella no será posible usar los pines. La definición de entradas y salidas se realiza en el setup y se debe hacer solo una vez para cada pin. Dentro del loop se programa la lectura o escritura digital o analógica de cada pin, usando:

  - **digitalWrite:** escritura digital, nos permite encender o apagar ledes o parlantes, y se usa principalmente para controlar dispositivos digitales, escribiendo un estado de alto o encendido (HIGH) y bajo o apagado (LOW).
  - **digitalRead:** lectura digital, nos permite leer los valores digitales de cualquier sensor digital, detectando si se registra un 1 o HIGH, o un 0 o LOW.
  - **analogWrite:** escritura analógica, nos permite controlar el encendido y apagado gradual de dispositivos analógicos como ledes RGB y motores, y se usa principalmente para programar dispositivos analógicos que se pueden controlar de forma gradual, escribiendo el valor analógico con el que se desea encender o apagar (un número entero entre 0 y 1023).
  - **analogRead:** lectura analógica, nos permite leer los valores analógicos de cualquier sensor analógico, detectando valores enteros entre 0 y 1023.

En el siguiente ejemplo se muestra una lectura analógica y una escritura digital, junto con la definición de pines de entrada y salida, donde se enciende o apaga un led verde dependiendo del valor analógico de un sensor, registrado por el Arduino.

sketch_sep26a §

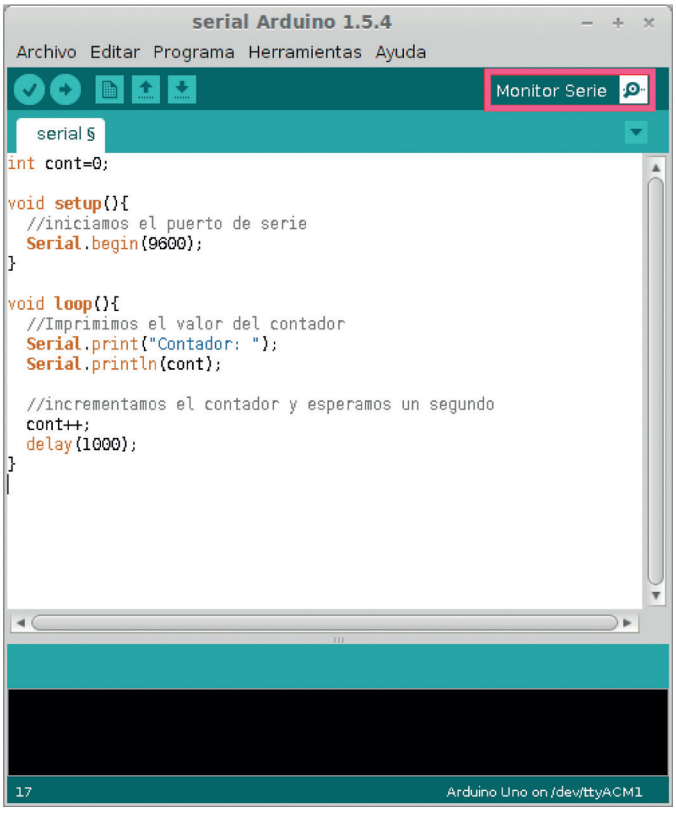
```
int LEDrojo = 9; //Se le asigna el pin 9 a la variable LEDrojo
int LEDverde = 8;//Se le asigna el pin 8 a la variable LEDverde

void setup() {
  pinMode(LEDrojo,OUTPUT); //Se declara el pin 9 como salida
  pinMode(LEDverde,OUTPUT);//Se declara el pin 8 como salida
}
```



- **Monitor serial:** Esta es una herramienta muy útil para ver la información que está recibiendo, enviando y manipulando el Arduino. Con unas instrucciones sencillas de programación es posible usar este monitor serial para mostrar desde texto hasta números, y solo se debe tener en cuenta inicializar la comunicación serial en el setup para activar esta ventana de información y asegurarse de coincidir el número de Baudios programado por código con los de la ventana Monitor Serial.

  - **Serial.begin:** permite inicializar la comunicación serial y establecer los baudios de comunicación, los cuales se dejarán por defecto en 9600.
  - **Serial.print:** permite escribir en el monitor serial. Se usan comillas dobles (“) para escribir texto y sin comillas para escribir números.
  - **Serial.println:** permite escribir en el monitor serial y agregar un salto de línea.

En el siguiente ejemplo se muestra un algoritmo que escribe en el monitor serial la palabra “Contador:” y muestra el valor actual de la variable “cont” cada segundo, incrementándolo de uno en uno.



- **Condicionales.** La estructura de un condicional en Arduino consta de las instrucciones “if”, “else” y “else if” que significan “si”, “sino” y “sino si”, donde la primera evalúa una condición y determina qué hacer si la condición se cumple, la segunda determina qué hacer si la condición no se cumple, y la tercera determina qué hacer si la condición no se cumple, pero sí se cumple una segunda condición. Este tipo de estructuras usan corchetes ({) para abrir y cerrar las instrucciones que se deben ejecutar en cualquiera de los casos, lo que significa que habrá un par de corchetes para las instrucciones del “if” otros para el “else” y así sucesivamente. Es importante abrir y cerrar correctamente los corchetes para evitar errores de sintaxis. También es importante usar paréntesis para escribir las condiciones a evaluar. Dentro de las condiciones se pueden usar comparadores de igualdad (==), desigualdad (<, >, <=, >=, !=) y conectores tipo y, o (&&, ||). Las condiciones que más se usan son para evaluar números con respecto a alguna referencia.

En el siguiente ejemplo se muestra un condicional tipo “if, else if, else”, allí la primera condición activa un encendido intermitente de un led, la segunda enciende un led y la tercera lo apaga. Observa el uso de los paréntesis y corchetes en toda la estructura y cómo el valor de la variable “Sensor” determina el comportamiento del led:

```
void loop() {
  if (analogRead(Sensor) > 800) {
    digitalWrite(LEDverde,HIGH);
    delay(500);
    digitalWrite(LEDverde,LOW);
    delay(500);
  }
  else if (analogRead(Sensor) > 400 && analogRead(Sensor) <= 800) {
    digitalWrite(LEDverde,HIGH);
  }
  else {
    digitalWrite(LEDverde,LOW);
  }
}
```

- **Llamar librerías.** Las librerías son extensiones que se pueden agregar a cualquier programa de Arduino con el fin de introducir nuevas instrucciones de programación creadas para realizar tareas específicas, por ejemplo, controlar servomotores, displays y sensores complejos de programar. La forma correcta de llamar la librería es agregando la



instrucción “#include” al inicio de todo el programa seguido del nombre de la librería con extensión “.h” y encerrado entre “< >” como se muestra en el ejemplo, donde se llama la librería del servomotor diseñada para programar servomotores.

En el siguiente ejemplo, se usa la librería del servomotor para mover un servomotor conectado en el pin 9.

```
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // twelve servo objects can be created on most boards

int pausa=100;

int sensorValue = 0;  // variable to store the value coming from the sensor
int outputValue = 0;        // value output to the PWM (analog out)

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}
```

NOTA

Si deseas agregar una librería completamente nueva que no esté incluida entre las librerías del Arduino, debes asegurarte de guardar la librería en la misma carpeta donde creaste el proyecto.

Cómo crear un nuevo proyecto en Arduino y descargarlo en la tarjeta.

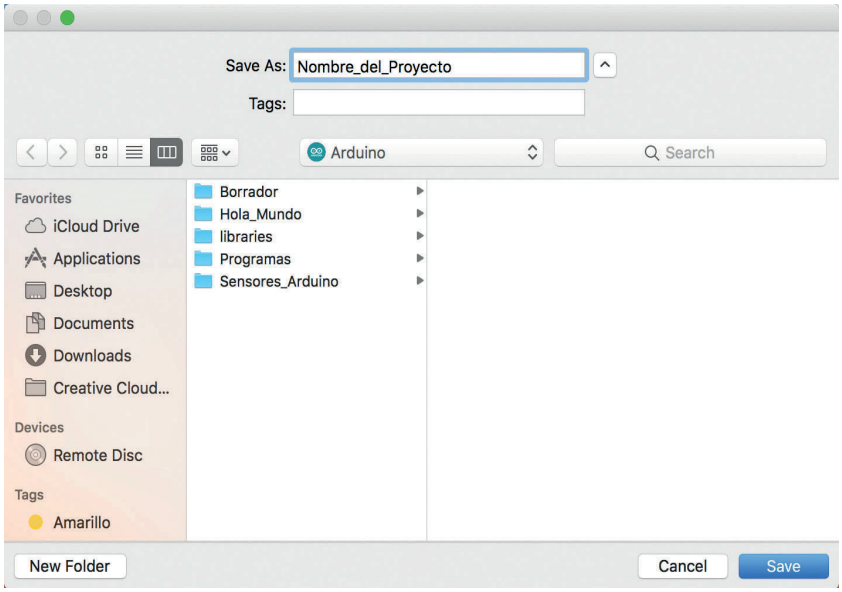
Para crear un proyecto nuevo en Arduino debes realizar los siguientes pasos:

1. Cuando se abre la aplicación de Arduino se crea automáticamente un nuevo proyecto.

a. Por otro lado, si ya hay un proyecto de Arduino abierto y se desea crear uno nuevo, se debe hacer clic en el icono con forma de hoja de papel llamado “nuevo” y se abrirá una nueva ventana.



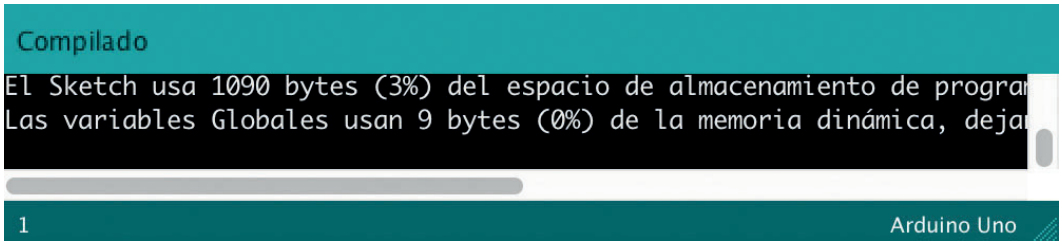
2. Para guardar el nuevo proyecto, se debe hacer clic en el icono de la flecha señalando hacia abajo llamada “salvar”, se debe escoger el directorio donde se guardará el programa y darle un nombre sin espacio, tildes, “ñ”, ni símbolos.



3. Después de guardar el proyecto, comience a escribir el algoritmo deseado en el espacio de trabajo, use el botón con forma de “OK” llamado “Verificar” para comprobar que no haya errores de sintaxis en el algoritmo, y prepárese para programar la tarjeta. Sabrá que no hubo errores de sintaxis si en la consola le aparece el mensaje “Compilado” y todos los textos que se muestran son de color blanco. En caso de que esto no suceda, lea los mensajes de color naranja que hayan aparecido y trate de solucionar el problema, compilando las veces que sea necesario hasta asegurarse de que el algoritmo es correcto. Recuerde que el software no le dejará programar el Arduino si se detecta



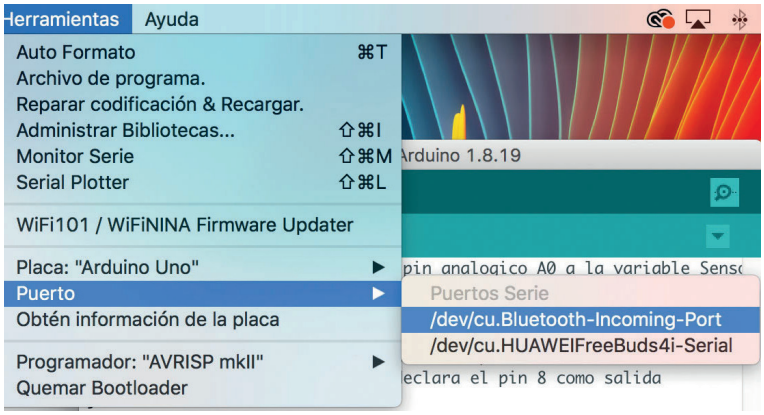
cualquier error de sintaxis.



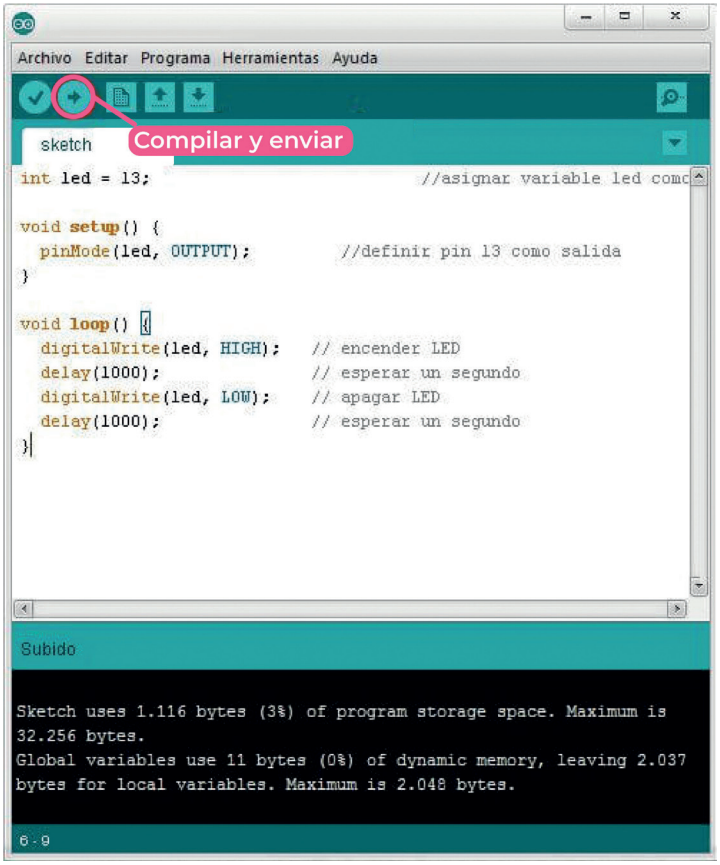
4. Una vez compilado el programa, proceda a conectar el Arduino al computador y realice los siguientes pasos:

a. En “Herramientas” seleccione la “placa: Arduino Uno”.

b. En “Herramientas” seleccione “Puerto” y dé clic en el puerto que diga “Bluetooth-incoming-port”, ya que este será el puerto en el que el computador reconoció el Arduino y será donde se cargará el programa.



5. Después de completar el paso anterior, asegúrese de que Arduino está encendido y proceda a hacer clic en el botón “Subir” representado con una flecha que señala hacia la derecha. Este botón se encargará de cargar el programa en la tarjeta Arduino. Espere a que se indique en la consola que se ha subido correctamente el programa antes de proceder a usar el Arduino (la consola debe decir “Subido”).



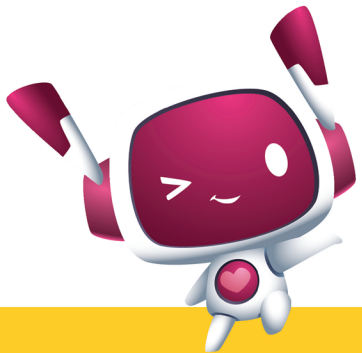
6. ¡Pruebe el programa y verifique si su programación fue correcta!

Instrucciones de mantenimiento

Limpieza de Gamepad

Para limpiar el exterior del gamePad, puede utilizar un cepillo de cerdas suaves para eliminar los depósitos alrededor de los botones y un paño de microfibra humedecido con un producto de limpieza para limpiar toda la carcasa, secándola inmediatamente después.

Si requiere de algún mantenimiento a las piezas internas, se sugiere recurrir a un técnico especializado para detectar el fallo correspondiente.



Micro:bit

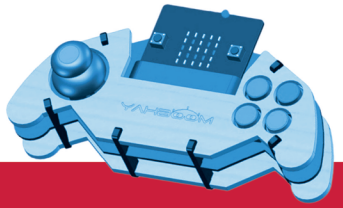
Problema	Motivo	Solución
La micro:bit no responde al programa que realicé	Es probable que la programación incluida en el archivo contenga algún error o que el archivo .hex no haya sido correctamente cargado en la micro:bit.	1. Revisar en MakeCode que el archivo esté correctamente programado. Si es necesario, corregirlo y volver a descargarlo en la micro:bit. 2. Si el archivo está correctamente programado, cárguelo nuevamente en la micro:bit y revise que el proceso finalice correctamente. 3. Si el error persiste, puede haber errores de conexión.
La micro:bit no enciende al conectar la batería	La batería funciona con dos pilas AAA. Es probable que se hayan agotado o que haya inconvenientes de hardware.	1. Conecte la micro:bit a un dispositivo por USB y revise si enciende la luz amarilla. Si esto no sucede, es probable que se trate de un error de hardware de la placa. 2. Si la luz sí enciende, prueba con nuevas pilas en el porta pilas. 3. Si luego del cambio de pilas sigue sin funcionar, es probable que se trate de un error físico de la batería o de la micro:bit.
No veo los archivos .hex en mi dispositivo	Los archivos .hex son descargados desde el sitio MakeCode de la misma forma que cualquier archivo de otro sitio	1. Revise la carpeta de descargas del dispositivo y buscar el archivo .hex. 2. Si no se encuentra, descargarlo nuevamente (identificando con un nombre el programa creado). 3. Si la descarga no es posible, revise la conectividad de internet.
Se cerró el navegador o quedé sin conexión mientras trabajaba en el sitio MakeCode	El sitio MakeCode aloja de forma local en el navegador los avances realizados en cada proyecto.	Al quedar sin conexión es posible seguir trabajando y descargar el proyecto. Si se cierra el navegador, se puede recuperar el último proyecto guardado y seguir guardando cambios. Si se borran datos de caché, la información se perderá.

Garantías

1. Si tiene algún inconveniente con alguna de las piezas de este equipo, podrá contactarse con la página **https://www.computadoresparaeducar.gov.co** en el chat en línea y el formulario PQRSDF, o al correo electrónico: **info@cpe.gov.co** donde se le brindará la asesoría correspondiente.

Adicionalmente, cuenta con otros canales de comunicación como:

- Líneas gratuitas nacionales: 018000919275 – 018000919273 – 018000949596 Directo de teléfono fijo o celular (operadores Movistar,



Tigo, Claro) sin indicativos.
- WhatsApp (solo mensajería): 3166585748
- Horarios de atención:

  - Lunes a viernes de 7am a 7pm.
  - Sábados de 8am a 5pm.

2. Para efectos de garantía por falla mecánica o electrónica, contáctese dentro de los primeros 10 días hábiles luego de la entrega a través del e-mail especificando las fallas presentadas . A través de este se le brindará respuesta pronta y solución.

3. Si después de hacer las correctivas necesarias su equipo sigue teniendo fallas, se le reemplazará el mismo por uno nuevo.

4. La garantía del producto no incluye el mal uso de este.

5. El equipo cuenta con una garantía de 3 años.

Sostenibilidad

La sostenibilidad implica un uso autónomo e intensivo de los recursos suministrados bajo los enfoques y finalidades que entrega el proyecto, durante el máximo lapso posible.

En contraste con la sostenibilidad, como se plantea acá, debemos recordar lo que ocurría antaño, cuando los recursos tecnológicos suministrados a las instituciones educativas permanecían en los almacenes de los colegios hasta cuando debían darse de baja por obsolescencia. ¿Por qué pasaba esto? Porque los rectores y profesores tenían miedo de dañarlos, de que se perdieran o porque no sabían cómo utilizarlos o porque temían que los estudiantes expusieran ante el grupo, porque los estudiantes tenían mejores habilidades en su uso (en su calidad de nativos digitales), y que esto “dejara mal parado” al docente como poseedor del conocimiento. Como vemos, muchas razones.

Hoy la situación ha cambiado, los docentes están mejor preparados para lidiar con la tecnología (la pandemia aportó lo suyo), las tecnologías digitales son



más amigables, los directivos docentes están mejor dispuestos y sensibilizados frente a la necesidad de emplear la tecnología para una educación de calidad, entre otros. Sin embargo, un factor determinante es que los diseñadores de los proyectos de innovación educativa han incrementado los aspectos pedagógicos, que son los que garantizan de verdad la apropiación efectiva de las nuevas tecnologías para una educación de calidad.

La sostenibilidad del proyecto "Laboratorios de Innovación Educativa" busca garantizar que los objetivos e impactos positivos del proyecto perduren de forma duradera después de finalizada la intervención de los gestores u operadores externos del proyecto, en este caso, CPE o la organización contratada para ejecutarlo.

Esta visión de la sostenibilidad busca ser alcanzada por varios elementos del proyecto que fueron introducidos con ese propósito y que forman parte del paquete pedagógico de este. A continuación, desarrollamos para los estudiantes, docentes y rectores los criterios de sostenibilidad incorporados a algunos de los componentes del proyecto.

Rol de las rutas didácticas en la sostenibilidad

Las 66 sesiones que conforman las rutas didácticas persiguen que los docentes y estudiantes de las sedes beneficiadas dispongan de un material que les permita ganar autonomía en el uso de las soluciones tecnológicas suministradas.

En ese sentido, las guías fueron diseñadas en un esquema sencillo, con un lenguaje al alcance de los estudiantes y docentes de las instituciones educativas oficiales de las zonas urbanas y rurales del proyecto.

Estos son talleres autocontenidos, es decir, no se necesitan recursos ni procedimientos por fuera de los suministrados por el proyecto para el diseño y el montaje de los prototipos por parte de estudiantes y docentes.

Hay seis talleres por grado educativo, lo que, por una parte, garantiza la cobertura a todos los estudiantes de la sede educativa y, por otra, posibilita que la experiencia de uso de las soluciones no se limite a un momento puntual, sino que cada curso puede experimentar por lo menos en seis momentos con dichas soluciones. Los docentes y estudiantes son los expertos para desarrollar las sesiones, no se requieren personas externas que deban guiar la implementación de los talleres.

El papel de los manuales y videos frente a la sostenibilidad

Desde la perspectiva técnica, la sostenibilidad ha sido tradicionalmente un factor negativo en los proyectos en la medida en que, por lo general, los proyectos con componentes tecnológicos crean una dependencia negativa hacia los agentes que poseen el conocimiento técnico. Si se requiere un cambio de disposición de los equipos, por ejemplo, o hacer una reinstalación de estos, usualmente los beneficiarios en la institución educativa se cohíben con lo cual se dejan de usar los equipos por el temor de dañarlos o no saber cómo reinstalarlos .

El proyecto suministra dentro del paquete pedagógico los manuales de los equipos (explicados en un lenguaje sencillo) en los cuales se describen sus componentes, así como los procedimientos paso a paso para su montaje y desmontaje.

Adicionalmente, los docentes y estudiantes disponen de videos que complementan los manuales técnicos con descripciones visuales y dinámicas de los procedimientos técnicos más importantes.

La formación de docentes y directivos docentes y la sostenibilidad

Pero, sin duda alguna, el elemento más importante de la sostenibilidad de este tipo de proyectos es la voluntad sincera y determinada de los líderes educativos, es decir, los rectores y docentes de las instituciones beneficiadas del uso de estos dispositivos con un sentido pedagógico. En esto desempeña un papel importante la conciencia y seguridad que se tenga, la utilidad de estos recursos para apalancar nuevos conocimientos, las nuevas posibilidades para la innovación educativa en las instituciones en aras de mejorar las perspectivas de futuro de los estudiantes que se educan en sus aulas.

Por consiguiente, el otro componente de la sostenibilidad del proyecto es la formación de los directivos docentes y de un grupo tanto de docentes como de estudiantes de las sedes, buscando no solo familiarizarlos con las soluciones tecnológicas, sino también mostrarles lo que el enfoque STEM representa en la comprensión del mundo del siglo XXI, y cómo estas tecnologías que sustentan este enfoque están involucradas en la mayoría de las dimensiones de la sociedad actual, de modo que desarrollar conocimientos y habilidades sobre sus características y usos, conlleva una ventaja competitiva para los



estudiantes frente a su futuro y lleva a la institución a cumplir su misión educadora de la manera más pertinente y cabal.

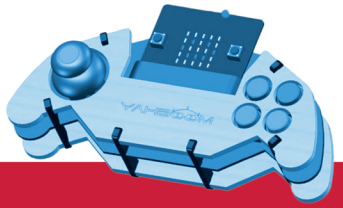
El kit de ingeniería STEM y la sostenibilidad del proyecto

Lo que en últimas hace que un objeto sea utilizable es que sea útil (elemento importante de la sostenibilidad).

Hoy por hoy, no hay nada más atractivo que la relevancia que ha tomado el enfoque STEM en todos los sistemas educativos del mundo. Los medios de comunicación han ayudado a posicionar el término STEM en el imaginario de los docentes y directivos docentes quienes leen STEM en clave de innovación educativa porque han entendido que cuando se define a la cuarta revolución industrial como la fusión de tecnologías del mundo físico, digital y biológico, lo que representa mejor esa articulación es que este enfoque que suma la ciencia, la tecnología, la ingeniería y las matemáticas para construir prototipos representa soluciones para muchos problemas cercanos e inmediatos en lo local, tal como lo han logrado en otros terrenos más complejos (como los que enfrentan las ciudades o el cambio climático).

Por ello, si algo aporta a la sostenibilidad del proyecto, es este ingrediente de las nuevas soluciones tecnológicas, ya que suscita la motivación de estudiantes y docentes, entusiasma a padres de familia cuando ven a sus hijos construir robots con los elementos del kit STEM y los directivos docentes ven en estos dispositivos la esperanza cierta de una educación de calidad.

Estos son los elementos más importantes que sustentan la sostenibilidad como se ha formulado en el proyecto “Laboratorio de Innovación Educativa”. Por esa razón, nuestro llamado es a que usen estos dispositivos pues están dadas las condiciones para un uso intensivo y prolongado en el tiempo de las nuevas posibilidades representadas por las nuevas soluciones tecnológicas.





LABORATORIOS
de Innovación Educativa



LABORATORIOS de Innovación Educativa

CISP
COMITATO INTERNAZIONALE PER LO SVILUPPO DEI POPOLI

Computadores para Educar