

# Jugando sólo con la micro:bit

- [Software : El entorno de trabajo de MakeCode](#)
- [Nuestro primer programa](#)
- [Retos básicos](#)
- [Retos avanzados](#)
- [Música y sonido con micro:bit](#)
- [Inteligencia artificial Create AI](#)

# Software : El entorno de trabajo de MakeCode

“ El **punto de partida** para el estudio de un lenguaje de programación pasa inevitablemente por los **lenguajes gráficos** o de bloques que facilitan su estudio al mostrar de forma visual las instrucciones del algoritmo programado. (Morales, 2023)

Vamos a utilizar **MakeCode** ofrece un entorno integrado, simple e intuitivo de **codificación por bloques**. Hay que decir que en el entorno de MakeCode también están disponibles los lenguajes **Java** y **Python**, con la ventaja de que un programa desarrollado mediante bloques puede ser traducido instantáneamente a cualquiera de los otros dos lenguajes.

Aquí tienes un esquema de dónde se sitúa Makecode, encaja dentro de los programas de bloque se se carga en la placa

[2025-11-30 10\\_24\\_40-ESQUEMA LENGUAJES PROGRAMACIÓN - PowerPoint.png](#)

Para acceder al entorno basta con abrir en un navegador cualquiera el enlace:

<https://makecode.microbit.org/>. Si nuestro navegador tiene desactivadas las cookies o navega en modo incógnito, aparecerá un mensaje de advertencia:

[Advertencia guardado automático.jpg](#) Pulsando sobre el botón de **continuar** aparecerá la ventana principal de MakeCode. Para iniciar el entorno de programación habremos de pulsar sobre el botón de **nuevo proyecto**:

[Nuevo proyecto.jpg](#) MakeCode nos pedirá seguidamente **dar un nombre** cualquiera al proyecto y pulsar el **botón de crear**, tras lo cual se abrirá la ventana del editor.

[Entorno MakeCode.jpg](#)

*Entorno Microsoft MakeCode. Captura de pantalla*

Para trabajar con MakeCode no se requiere registro. No obstante, pulsando sobre el icono de la esquina superior derecha, situado en la barra de herramientas, es posible acceder a

MakeCode desde una cuenta de Google o de Microsoft. Esta característica permite guardar en la nube los diseños realizados y acceder a ellos desde cualquier ordenador.

Podemos ajustar el **idioma de la interfaz** pulsando sobre la rueda dentada de la esquina superior derecha. Están disponibles 33 idiomas, incluyendo el español.

El panel de la izquierda representa el anverso de una placa micro:bit. Se trata de un **simulador**, que mostrará en tiempo real el efecto del programa en desarrollo. Nótese que no es necesario disponer de una placa real para realizar los primeros pasos en programación con micro:bit.

Con el puntero del ratón se puede interactuar con la placa y pulsar sus botones, agitarla o cambiar las condiciones simuladas de luz o de ruido. Por otro lado, los controles situados bajo la placa nos permiten:

#### Controles simulador.jpg

- Apagar o encender el simulador.
- Reiniciar el simulador, es decir, el programa en ejecución.
- Entrar en modo de depuración para localizar y solucionar errores.
- Activar o desactivar el altavoz del ordenador.
- Tomar una fotografía de la placa.
- Mostrar el simulador en pantalla completa.

El panel central es una **caja de herramientas** en forma de menú desplegable con todos los **bloques de programación** disponibles. Los bloques pueden ser arrastrados directamente con el ratón desde el menú. Cada color del menú se identifica con un tipo de acción o de estructura de control. Por ejemplo, el color verde se reserva para los bloques que permiten ejecutar acciones repetitivas o bucles.

#### Menú bucles.jpg

La **caja de búsqueda** colocada sobre el menú desplegable es muy útil para encontrar rápidamente el bloque más adecuado para realizar la acción deseada en cada momento.

El panel de la derecha es el **área de programación**. Aquí se escriben los programas arrastrando y encajando los bloques de código. Cuando se abre el navegador aparecen por defecto dos bloques azules de eventos del **menú básico**:

- **Al iniciar**, es la estructura que contendrá el algoritmo que será ejecutado por la placa sólo una vez, tras ser encendida o reiniciada.
- **Para siempre**, es el bucle que contendrá el algoritmo que se ejecutará indefinidamente, una y otra vez, cuando acabe la ejecución del algoritmo de inicio.



Pulsando con el botón derecho del ratón sobre un bloque se desplegará un **menú contextual** que dará varias opciones, como duplicar, borrar u obtener ayuda sobre el uso del bloque.[C](#)

[Menú contextual bloques.jpg](#)

Al colocar el puntero del ratón sobre un bloque, el editor mostrará una **etiqueta de ayuda** con una breve explicación sobre la función del bloque.

[Hints.png](#)

<https://makecode.microbit.org/reference> contiene explicaciones detalladas de los bloques de MakeCode.

En la parte inferior de la ventana de MakeCode se dispone la **barra de herramientas**, con botones para realizar acciones como descargar el programa en un placa micro:bit conectada al ordenador, guardar el programa en nuestro ordenador o en un repositorio en línea, hacer y deshacer y ampliar o reducir la imagen.

# Nuestro primer programa

El primer programa que se realiza cuando se comienza a aprender un nuevo lenguaje consiste casi siempre en presentar un **saludo por pantalla**. Micro:bit es más original, puesto que propone como primer programa un **corazón palpitante**.

Vamos a combinar el saludo inicial con el corazón. De esta forma ilustraremos el funcionamiento de los bloques de eventos **al inicio** y **para siempre**.

Se trata de seguir los pasos explicados en la página anterior y crear un programa llamado **Hola**. Tras la apertura del editor pulsaremos sobre el menú **Básico** y arrastraremos y encajaremos los bloques para configurar el siguiente programa: [Hola.jpg](#)

Para escribir el texto ¡Hola! basta con hacer clic sobre la caja blanca del bloque **mostrar cadena**. De igual forma, los iconos de los bloques **mostrar icono** y los números de los bloques **pausa (ms)** pueden seleccionarse pulsando sobre las flechas, acción que despliega menús de selección.

[Desplegable de bloque.jpg](#)

Como en el caso de la caja de texto, también puede teclearse directamente el número de milisegundos del bloque **pausa (ms)**.

¿Qué hace el programa? Cuando se enciende la placa, el **algoritmo de inicio mostrará una vez** mediante la matriz de LED la cadena de texto "¡Hola!". Como la pantalla es muy pequeña, micro:bit deslizará el texto horizontalmente, de derecha a izquierda.

Seguidamente, el **algoritmo principal** mostrará en pantalla el icono de un corazón grande y entrará en pausa 500 milisegundos, es decir, medio segundo. Pasado este tiempo, micro:bit mostrará un corazón pequeño y esperará otro medio segundo. Este algoritmo se ejecutará indefinidamente hasta que se apague la placa, se pulse el botón de reinicio o se acaben las pilas. El efecto visual será el de un **corazón palpitante**.

Nada más terminar de colocar el último bloque, el **simulador de MakeCode** mostrará en pantalla el resultado del programa en ejecución. Para reiniciar el programa hay que pulsar sobre el icono de reinicio, situado justo bajo el dibujo de la placa micro:bit.

[Saludo.gif](#)

## Cargar el programa en micro:bit

Hay que conectar primero la placa al ordenador por medio del cable USB. Después de unos instantes, el sistema operativo, (Linux, Windows o MacOS) reconocerá a la tarjeta micro:bit conectada como si se tratase de una unidad USB con el nombre MICROBIT.

*Microbit USB Mint.jpg* La unidad Micro:bit en el escritorio de Linux Mint

Si la tarjeta es nueva, llevará cargado de fábrica un programa de demostración que empezará a ejecutarse nada más ser conectada al puerto USB. El programa mostrará mensajes, iconos en movimiento y sonidos y pedirá al usuario que calibre el compás magnético.

## METODO RÁPIDO: EMPAREJAMIENTO (recomendable)

La primera vez que le damos a descargar, nos pide que conectemos la MICRO:BIT

[2025-11-02 19\\_03\\_44-WhatsApp.png](#)

La primera vez nos muestra la opción de EMPAREJAR

[2025-11-02 19\\_04\\_58-Greenshot.png](#)

Nos pregunta por la tarjeta que tiene que estar conectada al USB

[2025-11-02 19\\_06\\_36-Greenshot.png](#)

Una vez emparejado, ya sale la opción de descargar

[2025-11-02 19\\_08\\_18-Greenshot.png](#)

Empieza a cargarse

[2025-11-02 19\\_09\\_02-Greenshot.png](#)

### Y YA ESTA

□□□□□ La siguiente vez que utilices esa micro:bit automáticamente pasa a descargar de forma muy rápida.

## METODO QUE NO FALLA : DESCARGAR COMO ARCHIVO Y COPIAR Y PEGAR EN LA NUEVA UNIDAD USB CREADA

Seguidamente hay que pulsar sobre el botón **Descargar**, acción que abrirá el diálogo de guardado de archivos del sistema operativo. Bastará con seleccionar la unidad MICROBIT y confirmar la descarga.

[Descargar.jpg](#)

Si tienes Vitalinux, te recomendamos visitar <https://libros.catedu.es/books/vitalinux-y-robotica/page/r8-microbit>

Mientras dure la transferencia de datos, el **LED ámbar** situado junto al conector USB de micro:bit **parpadeará rápidamente**, quedando fijo cuando el programa haya sido transferido.

La transferencia de archivos a micro:bit lleva algo de tiempo, especialmente si el programa es largo. Tras la transferencia, el programa empezará a ejecutarse inmediatamente sobre la tarjeta micro:bit.

La desventaja es que es un método LENTO y no se reduce la velocidad en los siguientes intentos, no como en el método anterior

## ¿Y ahora qué?

Pues simplemente la placa ejecutará el programa cargado cuando se le alimente.

Sólo es posible cargar un programa a la vez en micro:bit, por lo que cada nuevo programa se reescribe sobre el anterior.

Si se desea que micro:bit funcione desconectado del ordenador, simplemente hay que alimentarlo por medio de la caja de pilas o mediante una batería externa o un cargador USB. El programa se ejecutará de igual forma, pues se encuentra guardado en la memoria de la placa.

[embedded-image-aT6h3zXI.jpeg](#)

# Retos básicos

## INDICE DE RETOS EN

<https://libros.catedu.es/books/microbit-car/chapter/retos-con-microbit>

[Panel LED](#)

[Jugando con los LED: bloques básicos de código](#)

[Sensor temperatura](#)

[Sensor luminosidad y sonido](#)

[Pulsadores](#)

[Acelerómetro](#)

[Brújula](#)

[Radio](#)

[Pines entrada y salida](#)

# Retos avanzados

Los tienes en <https://libros.catedu.es/books/microbit-car/chapter/retos-avanzados-con-microbit>

[Mejorando el termómetro](#)

[Aprender con un led la diferencia entre analógico y digital](#)

[Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código](#)

[Nivel de burbuja usando el giroscopio](#)

[Lectura puerto USB por MakeCode o Coolterm](#)

# Música y sonido con micro:bit

Lo tienes en <https://libros.catedu.es/books/microbit-car/chapter/musica-y-sonido-con-microbit>

[Una pequeña melodía](#)

[Grabadora de sonidos](#)

[Fray Santiago](#)

[Controles de volumen, tempo y tono](#)

# Inteligencia artificial Create AI

Lo tienes en <https://libros.catedu.es/books/microbit-car/chapter/create-ai>

## Qué es CreateAI

Es una aplicación web en [a https://createai.microbit.org/](https://createai.microbit.org/) ideada para realizar estos pasos: R...

## Vestibles

Para proyectos con CreateAI es recomendable fijar la micro:bit a una pulsera. Fuente Ro-botic...

## Paso 1 Collect data por Bluetooth y 1 micro:bit

Si trabajamos con un PC sin bluetooth no puedes usar esta página, mira esta página Conectamos nu...

## Paso 2 Test model

Grabando una acción Vamos a darle una acción, elegimos un icono, uno nombre y una acción, en es...

## Paso 3 Code

Una vez testado nuestro modelo pulsamos a Edit in MakeCode AHORA ES NUESTRO MOMENTO DE PROGRA...

## Sin Bluetooth y 2 micro:bit

La idea es utilizar una micro:bit que haga de "puente inalámbrico" utilizando la conexión vía rad...

## Entendiendo a mi peluche

Vamos a a reconocer diferentes gestos de un muñeco peluche. Extraído de <https://microbit.org/proj...>

## Más proyectos CreateAI



Con el mismo principio de detectar con la AI movimientos con el acelerómetro, se pueden hacer div...