

## Semillitas de programación

Grado sugerido: Sexto

**Simón Felipe Velandia Valderrama**

*Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.*



Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](#)

Para contactar al autor/a de este recurso, escriba a: simon.velan@gmail.com

## PLANTILLA DE PROYECTO

Este documento presenta instrucciones paso a paso para el diseño, programación y montaje de un proyecto de computación física, domótica o robótica.

Tenga en cuenta que la plataforma solo recibirá recursos en formato **.pdf** cuyo tamaño no exceda los **10MB de peso y las 20 páginas de extensión**.

<b>Duración</b>	<i>Enero a diciembre de año escolar lectivo, calendario A o B, para aplicación de Guía de Actividades. Tiempo promedio por Encuentro: 2 semanas</i>
<b>Objetivo y descripción del proyecto</b>	<p><i>El objetivo principal del proyecto es integrar la tarjeta Micro:bit en los momentos pedagógicos para desarrollar habilidades y competencias del siglo XXI en estudiantes de secundaria, mediante la creación del producto "Semillitas de Programación".</i></p> <p><i>Este producto consiste en los imprimibles: Planeador por encuentros, guía de actividades y un kit de proyectos STEAM, enfocados en las áreas curriculares de Matemáticas, Tecnología e Informática, y Ciencias Naturales.</i></p> <p><i>La iniciativa busca mejorar el rendimiento académico, promover habilidades como la creatividad, resolución de problemas y pensamiento computacional, y evaluar el impacto en el desarrollo de competencias del siglo XXI.</i></p>
<b>Lista de materiales</b>	CANTIDAD / RUBRO (Para grupos de 40 estudiantes)  10      Kit Keyestudio sensores para Microbit 45 in 1 (Micro:bit no incluida)  4      Club Pack: Caja de 10 placas BBC Micro:bit V2, soportes para batería, cable micro USB, 20 pilas AAA  5      Kit surtido de componentes electrónicos  5      Protoboard para montaje de circuitos  5      Cables Dupont por paquete (Presentacion M-M) 5      Cables Dupont por paquete (Presentacion H-H) 5      Cables Dupont por paquete (Presentacion M-H) 10     Motor bomba sumergible (para proyectos acuáticos) 20     Motorreductor 10     Servomotor SG90 5      Multímetro digital (para medir voltaje, corriente, continuidad, resistividad, entre otros)

	10 Placa de expansión para ampliar funcionalidades en Microbit 10 Arduino Uno R3 (para grados 9,10,11) 10 Tarjeta ESP32 (para grados 9,10,11) 10 Base Board Expansión ESP32 para ampliar conexiones (para grados 9,10,11) 10 Raspberry Pi Pico para proyectos complejos (para grados 9,10,11) 4 Computadores para programar proyectos de electronica y robótica. 40 Software MakeCode, Python, Arduino IDE, etc.
<b>Características del problema para tener en cuenta en la solución.</b>	<p><i>El proyecto atiende a la problemática de la falta de oportunidades efectivas para que los estudiantes desarrollen habilidades y competencias del siglo XXI en el contexto escolar, especialmente en zonas vulnerables de la comuna 6 y 7 de la ciudad de San José de Cúcuta.</i></p> <p><i>La situación está relacionada con limitaciones en recursos tecnológicos, prácticas pedagógicas tradicionales que no fomentan la innovación, y una infraestructura educativa que requiere instrumentos didácticos modernos. Además, se considera que muchos docentes y estudiantes aún enfrentan dificultades para incorporar tecnologías digitales en el proceso de enseñanza-aprendizaje, limitando así el potencial de desarrollo integral del estudiantado.</i></p>

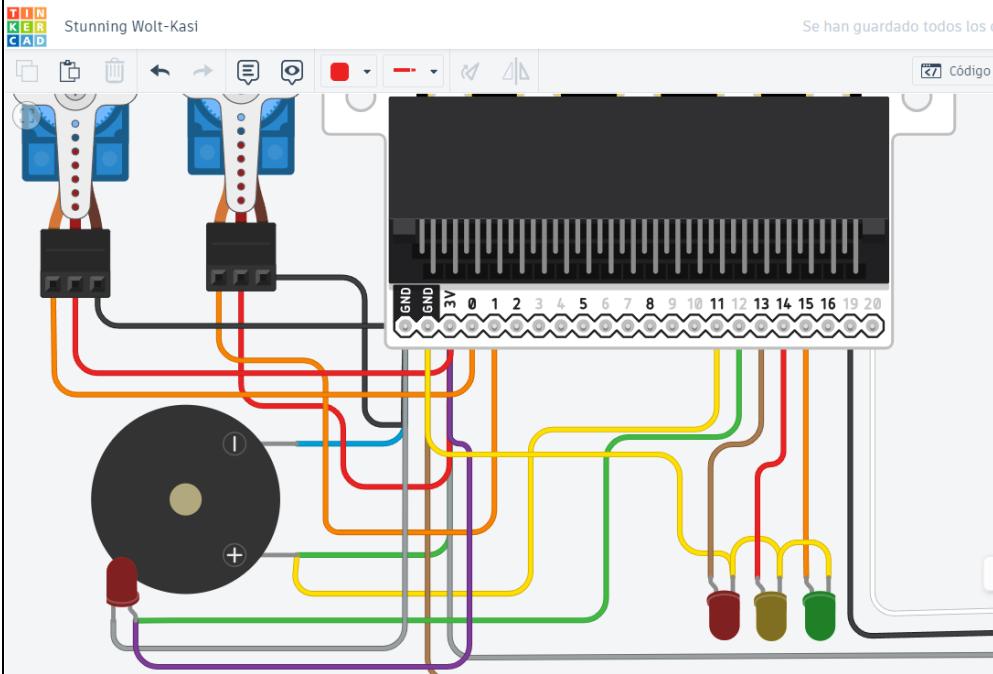
	<p>Para desarrollar el proyecto “<b>Semillitas de Programación</b>”, se aplica la <b>Guía de Actividades</b> tomando en cuenta la secuencia de actividades propuestas en el <b>Planificador de Encuentros</b>. Cada encuentro de la Guía de Actividades está estructurado en los siguientes pasos:</p> <p><b>Revisar</b> y comprender la sección de "Estructuración" para identificar los conceptos, objetivos y contenidos clave del encuentro.</p> <p><b>Organizar</b> los materiales y recursos necesarios según las instrucciones específicas de cada actividad.</p> <p><b>Verificar</b> el funcionamiento y la disponibilidad de todos los componentes electrónicos requeridos, incluyendo conexiones preliminares si es posible.</p> <p><b>Seguir</b> las instrucciones de conexión y montaje planteadas en la guía para integrar correctamente los componentes electrónicos con la Micro:bit.</p> <p><b>Elaborar</b> esquemas o diagramas de las conexiones para facilitar la comprensión y el montaje por parte de los estudiantes.</p> <p><b>Explicar</b> los conceptos teóricos relacionados con los componentes utilizados, enfocándose en su funcionamiento y finalidad.</p> <p><b>Preparar</b> el entorno de programación en MakeCode, abriendo y configurando el editor para facilitar la creación de los programas.</p>
<b>Pasos para desarrollar el proyecto</b>	<p><b>Diseñar</b> la lógica de programación en bloques en función de los objetivos del proyecto, incluyendo secuencias, condiciones y control de sensores o actuadores.</p> <p><b>Dialogar</b> con los estudiantes acerca de la relación entre las conexiones físicas y el código que se va a programar.</p> <p><b>Programar</b> en MakeCode las instrucciones necesarias, poniendo énfasis en que la lógica represente correctamente el funcionamiento del circuito.</p> <p><b>Probar</b> el funcionamiento del sistema completo, verificando que las conexiones y la programación se complementen correctamente.</p> <p><b>Ajustar</b> las conexiones o la programación según sea necesario para mejorar el funcionamiento y resolver errores detectados en las pruebas.</p> <p><b>Evaluar</b> la comprensión de los estudiantes mediante preguntas sobre el funcionamiento del circuito y el código programado.</p> <p><b>Promover</b> la reflexión acerca de los conceptos aprendidos y de posibles mejoras en el sistema.</p> <p><b>Proporcionar</b> actividades adicionales o desafíos creativos para profundizar en el aprendizaje y fomentar la innovación.</p> <p>A continuación, se muestra un encuentro que se propone en la <b>Guía de Actividades</b> del proyecto “<b>Semillitas de Programación</b>”:</p> <p><b>Encuentro 12: Proyecto Integrador con Micro:bit: Semáforo Inteligente con LCD y Servos. Duración: 2 semanas.</b></p> <ul style="list-style-type: none"> <li>• <b>Objetivo:</b> Integrar conocimiento de electrónica, programación y sensores para construir un semáforo inteligente que muestre información</li> </ul>

	<p>en pantalla LCD, controle servomotores y utilice sensores de temperatura y luz.</p> <ul style="list-style-type: none"> <li>• <b>Materiales:</b> Micro:bit, LEDs (rojo, verde), resistencias, cables, servomotores (2), botón, pantalla LCD compatible MakerBit, protoboard (opcional).</li> <li>• <b>Procedimiento:</b> <ol style="list-style-type: none"> <li>1. Diseño del circuito del semáforo con LEDs, servos para barreras y conexión de pantalla LCD.</li> <li>2. Conexión de LEDs, servomotores y botón a los pines de la Micro:bit, incluyendo la conexión de la pantalla LCD mediante el módulo MakerBit.</li> <li>3. Programación en MakeCode (Python o bloques) para:           <ul style="list-style-type: none"> <li>• Mostrar en la pantalla LCD la temperatura y nivel de luz ambiental de forma continua.</li> <li>• Controlar la secuencia del semáforo (verde y rojo) con tiempos definidos.</li> <li>• Activar servomotores para simular barreras en el cruce.</li> <li>• Reproducir sonidos durante la secuencia.</li> <li>• Implementar un botón peatonal que active una secuencia especial priorizando al peatón.</li> </ul> </li> </ol> </li> <li>• <b>Evaluación:</b> ¿Funciona correctamente la secuencia del semáforo con la integración de LCD, servos y sensores? ¿Se comprende la lógica del código y la interacción con los componentes?</li> <li>• <b>Extensión:</b> Añadir temporizadores para controlar los tiempos con mayor precisión, incorporar más sensores para ajustar la secuencia según condiciones ambientales, o agregar más efectos sonoros y visuales.</li> </ul> <p><b>Estructuración. <i>iSemáforo con Micro: bit!</i></b></p> <p><b>1. Materiales y conexiones</b></p> <p><b>Materiales:</b></p> <ul style="list-style-type: none"> <li>• 1 Micro:bit</li> <li>• 1 Placa expansora Keyestudio Sensor Shield V2</li> <li>• 1 Pantalla LCD (MakerBit, I2C)</li> <li>• 2 Servomotores</li> <li>• 3 LEDs (rojo, amarillo, verde)</li> <li>• 3 Resistencias (220Ω o 330Ω)</li> <li>• 1 Buzzer (zumbador)</li> <li>• Cables de conexión</li> <li>• <b>EXTENSIÓN MakerBit:</b> Esta extensión es fundamental para manejar la pantalla LCD 1602 compatible con Micro:bit y el módulo MakerBit que usas. Incluye bloques para conectar, limpiar y mostrar texto en la pantalla LCD. <b>Cómo agregar:</b> En MakeCode, ve a Avanzado &gt; Extensiones, busca "MakerBit" y agrégala.</li> <li>• <b>EXTENSIÓN Servos</b> (incluido en MakeCode por defecto): Para controlar los servomotores conectados a los pines P0 y P1. No necesitas una extensión adicional, ya que MakeCode tiene soporte nativo para servos.</li> </ul> <p><b>Conexiones principales:</b></p> <ul style="list-style-type: none"> <li>• <b>Servomotores:</b> <ul style="list-style-type: none"> <li>• Servo 1 → P0</li> <li>• Servo 2 → P1</li> </ul> </li> <li>• <b>LEDs:</b> puedes variar los pines según tu necesidad.</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>• Verde → P12</li> <li>• Amarillo → P13</li> <li>• Rojo → P14</li> <li>• <b>Buzzer:</b> <ul style="list-style-type: none"> <li>• P11</li> </ul> </li> <li>• <b>Pantalla LCD:</b> <ul style="list-style-type: none"> <li>• Conecta a los pines SDA y SCL de la placa expansora (usualmente P19 y P20 en Micro:bit)</li> </ul> </li> <li>• <b>GND y VCC:</b> <ul style="list-style-type: none"> <li>• Todos los componentes deben compartir GND y VCC con la placa expansora.</li> </ul> </li> </ul>
	<p><b>2. Estructura del programa</b></p> <p>El programa tiene dos partes principales:</p> <ul style="list-style-type: none"> <li>• Mostrar datos en la pantalla LCD (temperatura y luz, en tiempo real)</li> <li>• Secuencia del semáforo (con LEDs, servos y sonidos)</li> </ul>
	<p><b>3. Paso a paso para programar en MakeCode</b></p> <p><b>A. Mostrar datos en la pantalla LCD</b></p> <ol style="list-style-type: none"> <li>1. <b>Agrega la extensión MakerBit</b> en MakeCode (para controlar la pantalla LCD).</li> <li>2. <b>Inicializa la pantalla LCD</b> al iniciar el programa.</li> <li>3. <b>En un bucle infinito (forever):</b> <ul style="list-style-type: none"> <li>• Borra la pantalla LCD.</li> <li>• Muestra la temperatura en la primera línea.</li> <li>• Muestra el nivel de luz en la segunda línea.</li> <li>• Espera medio segundo antes de repetir.</li> </ul> </li> </ol> <p><b>Bloques MakeCode:</b> convertido a Python (en MakeCode)</p> <ul style="list-style-type: none"> <li>• makerbit.connect_lcd(39)</li> <li>• makerbit.clear_lcd1602()</li> <li>• makerbit.show_string_on_lcd1602()</li> <li>• input.temperature()</li> <li>• input.light_level()</li> </ul> <p><b>B. Secuencia del semáforo con servomotores y buzzer</b></p> <ol style="list-style-type: none"> <li>1. <b>En otro bucle (forever),</b> programa la secuencia de luces:       <ul style="list-style-type: none"> <li>• <b>Verde encendido:</b> <ul style="list-style-type: none"> <li>• Enciende LED verde (P12) y buzzer (P15).</li> <li>• Mueve los servos a la posición de "barrera abierta" (por ejemplo, Servo1 en 0°, Servo2 en 0°).</li> <li>• Espera 9 ciclos (puedes mostrar animaciones en la matriz de LEDs).</li> </ul> </li> <li>• <b>Amarillo encendido:</b> <ul style="list-style-type: none"> <li>• Apaga verde, enciende amarillo (P14).</li> <li>• Espera 2 segundos.</li> </ul> </li> <li>• <b>Rojo encendido:</b> <ul style="list-style-type: none"> <li>• Apaga amarillo, enciende rojo (P14).</li> <li>• Mueve servos a "barrera cerrada" (por ejemplo, Servo1 en 90°, Servo2 en 90°).</li> <li>• Espera 2 segundos.</li> </ul> </li> <li>• <b>Secuencia peatonal:</b> <ul style="list-style-type: none"> <li>• Enciende LED peatonal (P12).</li> <li>• Cuenta regresiva de 10 a 0 en la matriz de LEDs.</li> <li>• Reproduce sonidos de aviso.</li> <li>• Vuelve a la posición inicial de los servos.</li> </ul> </li> </ul> </li> </ol>

**Bloques MakeCode: convertido a Python (en MakeCode)**

- pins.digital\_write\_pin()
- servos.P0.set\_angle()
- servos.P1.set\_angle()
- music.play()
- basic.pause()
- basic.show\_number()
- basic.clear\_screen()



**Figura 12.** Prototipo de semáforo inteligente con Micro:bit, servomotores, buzzer, LEDs y pantalla LCD. Elaboración propia en Autodesk Tinkercad, 2025. Disponible en: <https://www.tinkercad.com/things/cLLZA6Zl0n6-semaforopeaje>

**iProgramando en MakeCode!**

1. Usa bloques o Python para:

- Mostrar en la LCD la temperatura y nivel de luz ambiental en tiempo real.
  - Controlar la secuencia del semáforo encendiendo y apagando LEDs con tiempos definidos (por ejemplo, verde 10 s, rojo 10 s).
  - Mover los servos para simular la apertura y cierre de barreras en el cruce.
  - Reproducir sonidos durante la secuencia para alertar a los usuarios.
2. Añade un evento "al presionar el botón" para activar una secuencia peatonal que priorice el cruce seguro, moviendo servos y cambiando luces.
3. A continuación, se presenta el código final (escrito en Python desde makecode.microbit.org)

```
n = 0
pins.digital_write_pin(DigitalPin.P11, 0)
makerbit.connect_lcd(39)
servos.P0.set_angle(0)
servos.P1.set_angle(0)

def on_forever():
```

```

makerbit.clear_lcd1602()
makerbit.show_string_on_lcd1602("Temperatura:",
    makerbit.position1602(LcdPosition1602.POS1),
    16)
makerbit.show_string_on_lcd1602("'" + str(input.temperature()),
    makerbit.position1602(LcdPosition1602.POS14),
    16)
makerbit.show_string_on_lcd1602("C", makerbit.position1602(LcdPosition1602.POS16), 16)
basic.pause(500)
makerbit.show_string_on_lcd1602("luz ", makerbit.position1602(LcdPosition1602.POS17), 16)
makerbit.show_string_on_lcd1602("'" + str(input.light_level())),
    makerbit.position1602(LcdPosition1602.POS21),
    16)
basic.forever(on_forever)

def on_forever2():
    global n
    while True:
        n = 10
        music.play(music.tonePlayable(880, music.beat(BeatFraction.WHOLE)),
            music.PlaybackMode.UNTIL_DONE)
        pins.digitalWritePin(DigitalPin.P15, 1)
        pins.digitalWritePin(DigitalPin.P12, 1)
        for index in range(9):
            basic.showLeds("""
                .#.
                .###.
                #####
                .....
                .....
                """)
            basic.showLeds("""
                .....
                .....
                ..#..
                .###.
                #####
                .....
                """)
            basic.showLeds("""
                .....
                ..#..
                .###.
                #####
                .....
                """)
        basic.clearScreen()
        pins.digitalWritePin(DigitalPin.P15, 0)
        pins.digitalWritePin(DigitalPin.P12, 0)
        pins.digitalWritePin(DigitalPin.P14, 1)
        basic.showLeds("""
            ..#..
            ..#..
            ..#..
            .....
            ..#..
            .....
            """)
        basic.pause(2000)

```

```

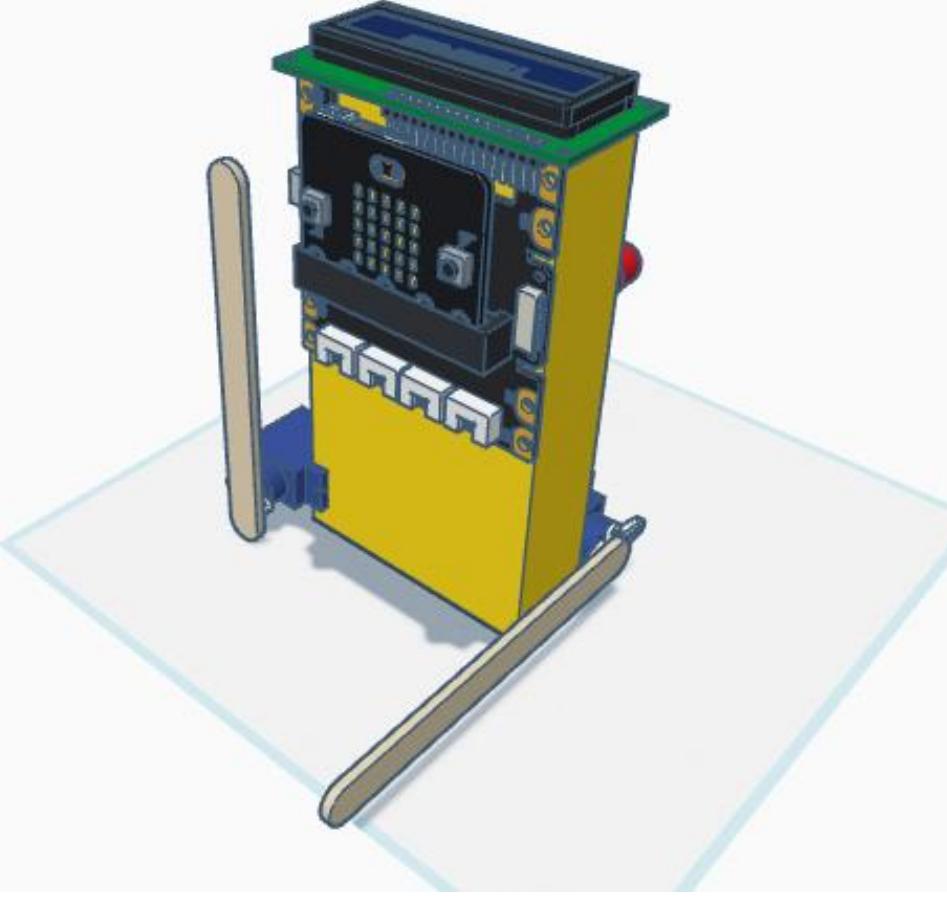
for index2 in range(5):
    basic.show_leds("""
        . # .
        . # .
        . # .
        ....
        . # .
    """)
    pins.digital_write_pin(DigitalPin.P14, 1)
    pins.digital_write_pin(DigitalPin.P11, 1)
    basic.pause(200)
    pins.digital_write_pin(DigitalPin.P14, 0)
    pins.digital_write_pin(DigitalPin.P11, 0)
    pins.digital_write_pin(DigitalPin.P14, 0)
    servos.P0.set_angle(90)
    basic.pause(200)
    servos.P1.set_angle(0)
    pins.digital_write_pin(DigitalPin.P13, 1)
for index3 in range(10):
    n += -1
    basic.show_number(n)
    music.ring_tone(131)
    basic.pause(500)
    music.ring_tone(0)
    basic.clear_screen()
    pins.digital_write_pin(DigitalPin.P13, 0)
    basic.pause(200)
    servos.P0.set_angle(0)
    basic.pause(500)
    servos.P1.set_angle(90)
basic.forever(on_forever2)

```

4. La programación en bloques disponible en  
[https://makecode.microbit.org/\\_0sMTdCaWmexg](https://makecode.microbit.org/_0sMTdCaWmexg) :

**¿Qué has aprendido?**

- Integrar sensores ambientales y mostrar datos en pantalla LCD.
- Controlar servomotores para simular mecanismos físicos.
- Programar secuencias complejas con sonidos y eventos.
- Diseñar un proyecto práctico y funcional combinando hardware y software.

	 <p><b>Figura 12.</b> Prototipo de semáforo inteligente con Micro:bit, barreras y pantalla LCD. Elaboración propia en Autodesk Tinkercad, 2025. Disponible en: [<a href="https://www.tinkercad.com/things/iuz69mIx18U-semaforoservos?sharecode=YbHm7XqcNzrr20QCeICGKKboJK-HSqdCRAPdiTW55L8">https://www.tinkercad.com/things/iuz69mIx18U-semaforoservos?sharecode=YbHm7XqcNzrr20QCeICGKKboJK-HSqdCRAPdiTW55L8</a> ]</p> <p><b>iDesafío extra!</b></p> <ul style="list-style-type: none"> <li>• Investiga cómo usar temporizadores para controlar los tiempos con mayor precisión.</li> <li>• Añade sensores adicionales para ajustar la secuencia según la temperatura o la luz ambiental.</li> <li>• Incorpora más efectos sonoros y visuales para mejorar la interacción.</li> </ul> <p><i>Enlace a demostración de dispositivo prototipo en funcionamiento:</i></p> <p><a href="https://youtube.com/shorts/tQ1P0tAiDdY">https://youtube.com/shorts/tQ1P0tAiDdY</a></p>
<b>Adaptaciones</b>	<p>Para diversos contextos, se sugieren las siguientes adaptaciones:</p> <ul style="list-style-type: none"> <li>• <b>Zona rural:</b></li> <li>• <i>Utilizar recursos tecnológicos disponibles localmente, como tablets o computadores simplificados.</i></li> </ul>

	<ul style="list-style-type: none"> <li>• <i>El diseño y aplicación de actividades que no requieren conexión a Internet, se usa la guía impresa y materiales offline. Para el caso del uso de extensiones en MakeCode, es necesario la conectividad a internet.</i></li> <li>• <i>Capacitar a docentes en metodologías adaptadas a entornos rurales y limitaciones de conectividad.</i></li> <li>• <b>Población con discapacidad:</b></li> <li>• <i>Los recursos son accesibles, ya que cada encuentro en la guía de actividades es adaptable e inclusiva, y permiten la participación activa de todos los estudiantes.</i></li> <li>• <b>Sin acceso a Internet:</b></li> <li>• <i>Cada encuentro se puede desarrollar con recursos de los kits físicos y actividades completamente offline.</i></li> <li>• <i>Capacitar en la gestión y mantenimiento de los materiales tecnológicos disponibles en el centro educativo.</i></li> <li>• <i>En caso del uso de extensiones en MakeCode, es necesario la conectividad a internet para previamente integrarlas en cada encuentro según sea el proyecto de aula a prototipar.</i></li> </ul>
<b>Referencias</b>	<p><i>Joint Research Centre (2019). Developing computational thinking in compulsory education: Implications for policy and practice. Comisión Europea.</i></p> <p><i>Ministerio de Educación Nacional, Colombia. (2025). Propósitos y lineamientos para la integración de tecnologías en la educación.</i></p> <p>Pérez, J., &amp; Ramírez, L. (2023). Uso de micro:bit para fomentar habilidades del siglo XXI en estudiantes de secundaria. <i>Revista Innovación Educativa</i>, 15(2), 34-45.</p> <p><i>Guía de recursos STEAM y micro:bit. (2022). Fundación Micro:bit.</i></p> <p><i>ISO 21001:2018. Sistemas de gestión para organizaciones educativas.</i></p>

## ANEXO(s)

*Incluya los anexos requeridos aquí. Si son videos, presentaciones u otros materiales, ingrese un enlace y/o un código QR que permita accederlos libremente.*

Sites (en construcción) para Proyecto Semillitas de Programación.

<https://sites.google.com/view/semillitas-de-programacion/inicio?authuser=0>

Drive para descargar.

[https://drive.google.com/drive/folders/1t3AzVsikxRzTdNSoMK5kEvcdpy4u3puK?usp=drive\\_link](https://drive.google.com/drive/folders/1t3AzVsikxRzTdNSoMK5kEvcdpy4u3puK?usp=drive_link)

Allí se encuentra:

- Planificador x Encuentros (incluye propuesta de Diagrama Gantt)
- Materiales para Guía de Actividades.
- Guía de Actividades (Laboratorios).

## OTRAS PIEZAS RELACIONADAS .

Algunas propuestas del Proyecto a instituciones: ProFuturo (Movistar), REDUCALIA (Red de Maestras y Maestros Investigadores de Norte de Santander) y Secretaría de Educación Municipal Cúcuta.

\* Construcción de un sistema de riego automático para plantas ornamentales, mediante motor eléctrico, sensor de humedad, tarjeta microcontroladora Micro:bit y el software de programación MakeCode. Experiencia expuesta en el Foro Educativo 2023 de la Secretaría de Educación Municipal Cúcuta.

Enlace: <https://www.youtube.com/watch?v=SvJ85e9RnYM>

\* Presentación del proyecto ante la REDUCALIA Norte de Santander: Integración de la tarjeta Micro:bit en los momentos pedagógicos para desarrollar competencias y habilidades del siglo XXI en estudiantes de secundaria. Esta propuesta se asocia con la categoría de Ciencia, Tecnología e Innovación y responde a una problemática del contexto escolar. El resultado esperado es el diseño de un producto concreto que contribuya al mejoramiento de la situación particular, en este caso, el desarrollo de habilidades y competencias en los estudiantes, mediante las fichas metodologías y el kit de proyectos STEAM: "Semillitas de Programación".

Enlace: <https://youtu.be/2FrWmCMYgxw>

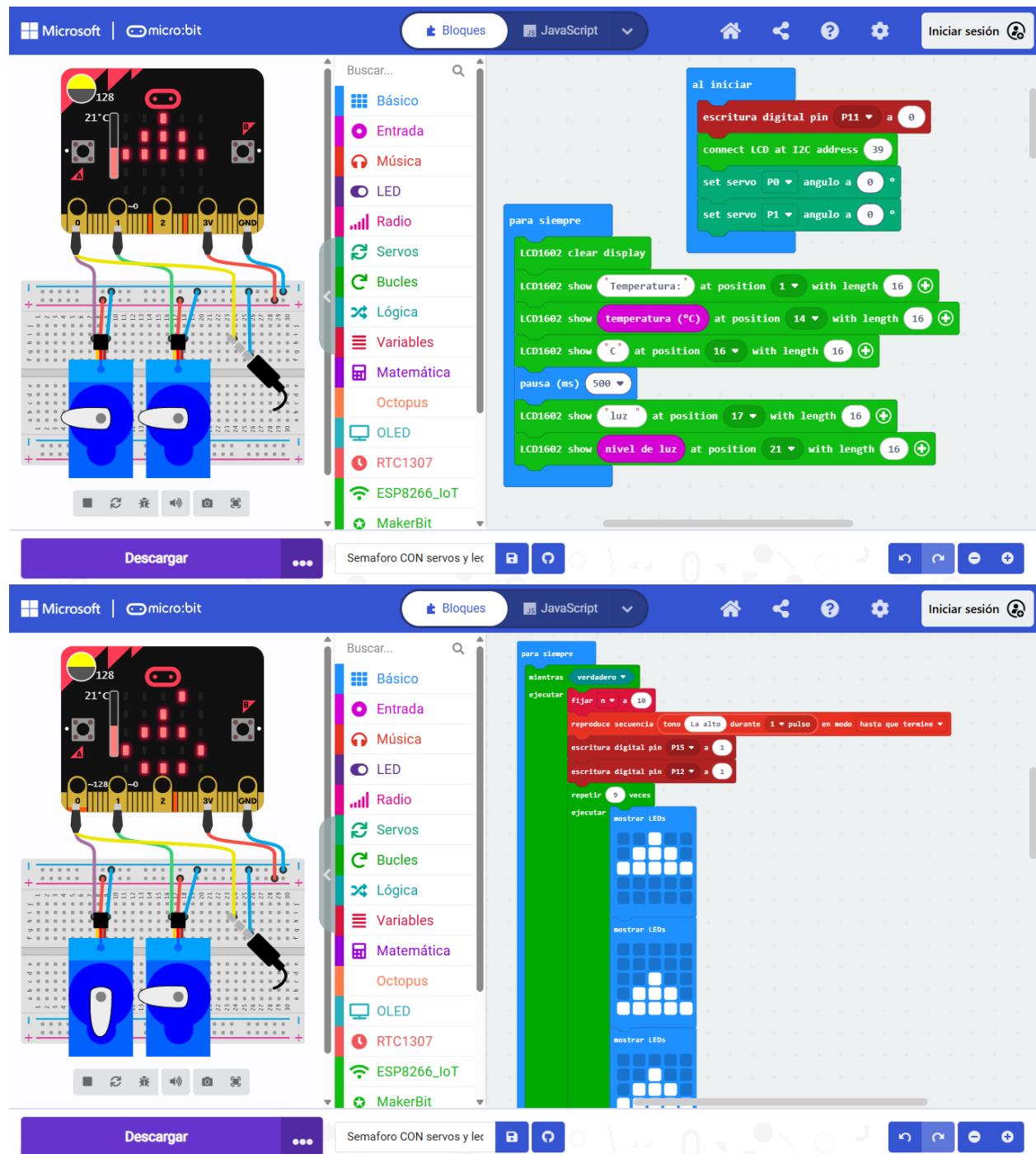
\* Construcción de "Dado Electrónico" programado en el software MakeCode con la tarjeta Micro:Bit, para jugar en tableros de parqués, ludo y otros juegos de azar, aplicación de la metodología SCAMPER (Sustituir, Combinar, Adaptar, Modificar, Poner otro uso, eliminar y Reorganizar). Experiencia expuesta a la Fundación Telefónica Movistar Colombia en el evento: 10° Encuentro Nacional de Docentes ProFuturo.

Enlace: <https://www.youtube.com/watch?v=RUAi3XrkRBs>

\* Elaboración de una aplicación móvil, por estudiantes para los jóvenes de la comunidad del sector Simón Bolívar y Toledo Plata de la ciudad de Cúcuta; en la que se informa a los jóvenes sobre los riesgos a los que pueden exponerse al utilizar las redes sociales. Además, elaborar un sistema "Interruptor" de encendido y apagado programado para timbre de Colegio Camilos Daza, Sede San Andrés II. Experiencia expuesta en el Foro Educativo Municipal para la S.E.M. Cúcuta.

Enlace: <https://www.youtube.com/watch?v=rp1VRSVIALc>

## Algunas captures de proyecto “Semáforo inteligente con LCD y Servos” (Encuentro 12).



The image shows two Scratch projects for a micro:bit board, each featuring a 2x2 servo matrix and an LCD screen.

**Top Project:**

- Circuit Diagram:** Shows a 2x2 grid of servos connected to pins 0-3. Pin 0 is connected to ground. Pin 1 is connected to digital pin P14. Pin 2 is connected to digital pin P11. Pin 3 is connected to digital pin P13. An LCD screen is connected to pins 12 and 13.
- Scratch Script:** A green script consisting of:
  - mostrar LEDs
  - borrar la pantalla
  - escritura digital pin P14 a 0
  - escritura digital pin P12 a 0
  - escritura digital pin P14 a 1
  - mostrar LEDs
  - pausa (ms) 2000
  - repetir [5 veces]
  - ejecutar mostrar LEDs

**Bottom Project:**

- Circuit Diagram:** Same as the top project, with the addition of a speaker connected to digital pin P10.
- Scratch Script:** A green script consisting of:
  - escritura digital pin P14 a 1
  - escritura digital pin P11 a 1
  - pausa (ms) 200
  - escritura digital pin P14 a 0
  - escritura digital pin P11 a 0
  - escritura digital pin P10 a 1
  - set servo P0 a 90
  - pausa (ms) 200
  - set servo P1 a 0
  - escritura digital pin P13 a 1
  - repetir [10 veces]
  - ejecutar cambiar n por -1
  - mostrar número n
  - tomo de timbre (Hz) Do bajo
  - pausa (ms) 500
  - tomo de timbre (Hz) 0 Hz
  - borrar la pantalla
  - escritura digital pin P13 a 0

