Introducción a la Programación con la Placa Microbit

1. Objetivo

Los estudiantes aprenderán los conceptos básicos de programación utilizando la placa Microbit. Desarrollarán habilidades para programar bucles, condicionales, animaciones, y responder a eventos como la presión de botones utilizando tanto MICROPYTHON como bloques en MAKECODE.

2. Duración

2 horas

3. Materiales Necesarios

- Placas Microbit y cables USB para la conexión.
- Computadoras con acceso a Internet.
- Acceso a las plataformas de programación MICROYPYTHON y MAKECODE.
- Proyector o pantalla para mostrar ejemplos de programación.

4. Estructura de la Actividad

4.1. Introducción

4.1.1. Presentación de la Programación con Microbit

- Explicar qué es la Microbit y sus capacidades.
- Demostrar cómo se puede programar la Microbit utilizando MICROYPYTHON y MAKECODE.

4.1.2. Planteamiento de la Actividad

- Introducir la serie de ejercicios que se realizarán, que incluyen la creación de programas simples y complejos para aprender sobre bucles, condicionales, animaciones y eventos de botones.

4.2. Desarrollo de la Actividad

4.2.1. Ejercicio 1: Mi Primer Programa

- **Planteamiento del Problema:** Crear un programa sencillo que muestre un mensaje de texto en la pantalla LED de la Microbit y envíe un mensaje a la consola de la computadora. Este programa debe ejecutarse de manera continua, repitiendo ambos mensajes indefinidamente.

4.2.2. Ejercicio 2: Bucles

- **Planteamiento del Problema:** Utilizar bucles para contar números y crear una animación sencilla en la pantalla LED de la Microbit. El programa debe contar del 0 al 4 usando un bucle while, luego mostrar una animación de un corazón latiendo 5 veces usando un bucle for. Todo el proceso debe repetirse indefinidamente.

4.2.3. Ejercicio 3: Animaciones

- **Planteamiento del Problema:** Crear una animación en la pantalla LED de la Microbit que cambie entre dos imágenes de flechas: una apuntando a la derecha y otra a la izquierda. La animación debe repetirse de forma continua, con una breve pausa entre cada cambio.

4.2.4. Ejercicio 4: Condicionales

- **Planteamiento del Problema:** Crear un programa en la Microbit que lea la temperatura ambiente y muestre diferentes imágenes en la pantalla LED según el valor. Si la temperatura es mayor a 25°C, se mostrará una sombrilla; si es menor a 10°C, un cuadrado; y si está entre 10°C y 25°C, una cara feliz.

4.2.5. Ejercicio 5: Botones

- **Planteamiento del Problema:** Crear un programa en la Microbit que detecte cuándo se presionan los botones A y B, mostrando una letra en la pantalla LED según el botón presionado. Si se presiona el botón A, muestra "A"; si se presiona el botón B, muestra "B"; si no se presiona ningún botón, la pantalla está apagada.

5. Rúbrica de Evaluación

Criterio	No Entrega	Mejorable	Bien	Excelente
Precisión y Exhaustividad	No se entrega trabajo. **(0 puntos)**	Faltan varios conceptos clave o las soluciones no cumplen con los objetivos. **(5 puntos)**	La mayoría de los conceptos están incluidos y las soluciones son funcionales con algunos errores menores. **(8 puntos)**	Todos los conceptos clave están incluidos y las soluciones son precisas, completas y funcionan correctamente. **(10 puntos)**

6. Objetivos de Desarrollo Sostenible (ODS) relacionados

- **ODS 4: Educación de Calidad**: Fomentar una comprensión profunda de la programación a través de la práctica y la resolución de problemas.
- **ODS 9: Industria, Innovación e Infraestructura**: Promover la innovación y la comprensión tecnológica mediante la programación de dispositivos físicos.

7. Principios del Diseño Universal para el Aprendizaje (DUA)

- **Múltiples medios de representación**: Uso de diferentes lenguajes de programación (MICROPYTHON y MAKECODE) para representar soluciones.
- **Múltiples medios de acción y expresión**: Permitir a los estudiantes elegir entre código o bloques para expresar su solución.
- **Múltiples medios de compromiso**: Involucrar a los estudiantes en la creación de programas basados en sus propias interpretaciones y creatividad.

8. Resultados Esperados

- Los estudiantes habrán creado varios programas que demuestran su comprensión de los conceptos básicos de programación.
- Habrán demostrado habilidades en la escritura de código, la resolución de problemas y el pensamiento lógico.

- Habrán participado activamente en la discusión, mostrando su comprensión del tema y su capacidad para conectar ideas de programación con la práctica en hardware.

9. Soluciones

9.1. Solución para Ejercicio 1: Mi Primer Programa

9.1.1. MICROPYTHON

python Copiar código

```python
def on_forever():
    # Mostrar texto en la pantalla LED
    basic.show_string("Hola Mundo!")

    # Mostrar un mensaje en la consola
    serial.write_line("Este texto aparece en la consola")

# Iniciar la ejecución continua de la función on_forever
basic.forever(on_forever)
```

9.1.2. BLOQUES MAKECODE

1. Utiliza "al iniciar" para configurar.
2. Agrega un bloque "para siempre" para crear un bucle infinito.
3. Dentro del bucle, usa el bloque "mostrar cadena" con el texto "Hola Microbit".
4. Añade "serial escribir línea" con el texto "Hola desde la consola".
5. Agrega un "pausar" de 1000 ms.

plaintext Copiar código

```
[para siempre]
    ├─ [mostrar cadena "Hola Mundo!"]
    └─ [serial escribir línea "Este texto aparece en la consola"]
```

9.2. Solución para Ejercicio 2: Bucles

9.2.1. MICROPYTHON

```python
def on_forever():
    # Bucle while que cuenta del 0 al 4
    contador = 0
    while contador < 5:
        basic.show_number(contador)
        basic.pause(1000)
        contador += 1

    # Bucle for que muestra un corazón latiendo 5 veces
    for i in range(5):
        basic.show_icon(IconNames.HEART)
        basic.pause(500)
        basic.show_icon(IconNames.SMALL_HEART)
        basic.pause(500)

basic.forever(on_forever)
```

9.2.2. BLOQUES MAKECODE

1. Utiliza un bloque "para siempre".
2. Usa "repetir mientras" para contar del 0 al 4.
3. Añade un bloque "repetir" 5 veces.
4. Dentro del bucle "repetir", usa "mostrar icono" (corazón) y "pausar" 300 ms.

plaintext Copiar código

```
[para siempre]
    ├─ [mientras contador < 5]
    │    ├─ [mostrar número contador]
    │    ├─ [pausa (1000 ms)]
    │    └─ [cambiar contador por 1]
    └─ [repetir 5 veces]
         ├─ [mostrar ícono]
         ├─ [pausa (500 ms)]
         ├─ [mostrar ícono]
         └─ [pausa (500 ms)]
```

9.3. Solución para Ejercicio 3: Animaciones

9.3.1. MICROPYTHON

```
flecha_derecha = images.create_image("""
    . . # . .
    . . . # .
    # # # # #
    . . . # .
    . . # . .
""")

flecha_izquierda = images.create_image("""
    . . # . .
    . # . . .
    # # # # #
    . # . . .
    . . # . .
""")

def on_forever():
    # Bucle que alterna entre las dos flechas
    flecha_derecha.show_image(0)
    basic.pause(500)
    flecha_izquierda.show_image(0)
    basic.pause(500)

basic.forever(on_forever)
```

9.3.2. BLOQUES MAKECODE

```
[al iniciar]
    ├─ [fijar flecha_derecha a crear imagen]
    │    └─ (imagen de flecha hacia la derecha)
    └─ [fijar flecha_izquierda a crear imagen]
         └─ (imagen de flecha hacia la izquierda)
```

1. Usa el bloque "para siempre".
2. Dentro del bucle, "mostrar icono" (flecha a la derecha).
3. "Pausar" 500 ms.
4. "Mostrar icono" (flecha a la izquierda).
5. "Pausar" 500 ms.

9.4. Solución para Ejercicio 4: Condicionales

9.4.1. MICROPYTHON

```
def on_forever():
    # Obtener la temperatura del micro:bit
    temperatura = input.temperature()

    # Condiciones para mostrar diferentes imágenes según la temperatura
    if temperatura > 25:
        basic.show_icon(IconNames.UMBRELLA)
    elif temperatura < 10:
        basic.show_icon(IconNames.SQUARE)
    else:
        basic.show_icon(IconNames.HAPPY)

basic.forever(on_forever)
```

9.4.2. BLOQUES MAKECODE

```
[para siempre]
    ├─ [fijar temperatura a temperatura (°C)]
    ├─ [si temperatura > 25 entonces]
    │    └─ [mostrar ícono (ícono personalizado)]
    ├─ [si no, si temperatura < 10 entonces]
    │    └─ [mostrar ícono (ícono personalizado)]
    └─ [si no]
         └─ [mostrar ícono (ícono personalizado)]
```

1. Usa el bloque "para siempre".
2. Dentro, utiliza "si... entonces... sino si... sino".
3. Primera condición: temperatura > 25 (mostrar sombrilla).
4. Segunda condición: temperatura < 10 (mostrar cuadrado).
5. Tercera condición: entre 10 y 25 (mostrar cara feliz).
6. "Pausar" 1000 ms.

9.5. Solución para Ejercicio 5: Botones

9.5.1. MICROPYTHON

```
def on_forever():
    # Bucle para detectar presiones de los botones A y B
    if input.button_is_pressed(Button.A):
        basic.show_string("A")
    elif input.button_is_pressed(Button.B):
        basic.show_string("B")
    else:
        basic.clear_screen()

basic.forever(on_forever)
```

9.5.2. BLOQUES MAKECODE

1. Usa "para siempre".
2. "Si... entonces... sino si... sino" para manejar entradas.
3. Primer bloque "si": botón A presionado (mostrar "A").
4. Segundo bloque "sino si": botón B presionado (mostrar "B").
5. "Sino": limpiar pantalla.

```
[para siempre]
    ├─ [si botón A presionado entonces]
    │    └─ [mostrar cadena "A"]
    ├─ [si no, si botón B presionado entonces]
    │    └─ [mostrar cadena "B"]
    └─ [si no]
         └─ [borrar la pantalla]
```

ANEXOS

1. Guía de conceptos clave para la programación en Microbit.
2. Ejemplos de códigos y soluciones utilizando MICROPYTHON y MAKECODE.