

TARJETAS RETO 1

RETO 1

Imaginad que necesitamos que una luz artificial se encienda y se apague de modo automático dependiendo de la luz ambiental natural. Seguro que se os ocurren un montón de contextos en que eso podría ser muy útil. Solo tendremos la luz encendida cuando sea necesario. Vamos a programar nuestra tarjeta micro:bit para que detecte los cambios de luz ambiental. Más adelante podremos programar que se encienda automáticamente la luz de una habitación solo cuando haga falta y la apague cuando no sea necesario. Así reduciremos el consumo de electricidad.

CONJETURA

Antes de empezar a programar nuestra tarjeta en Makecode vamos a estudiar cómo es el razonamiento que debe seguir la tarjeta para hacer lo que queremos que haga.

Rellenad la siguiente ficha con las partes de la tarea y las relaciones lógicas entre esas partes. Se trata de hacer una conjetura o una especie de hipótesis de lo que creemos que debe ocurrir. Podéis rellenar solo las casillas que necesitáis y tachar las que no o incluir nuevas relaciones usando flechas.

PISTA 1

Si hay demasiada luz, entonces la tarjeta deberá...

Qué consideramos demasiada luz
Qué tipo de instrucción debemos programar para que la tarjeta haga una u otra cosa.

En este vídeo se explica brevemente cómo funciona un sensor de luz y cómo MICRO:BIT puede comportarse como si fuera uno.



www.bit.ly/cajamicrobitR1P1

PISTA 2

Puedes usar estos bloques de makecode para hacer la programación, pero la relación entre ellos la tenéis que encontrar en equipo.



www.bit.ly/cajamicrobitR1P2

EXPLICACIÓN

En el siguiente enlace tienes el código completo de una de las posibles soluciones para hacer que tu MICRO:BIT se comporte como un sensor de luz, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. No olvides rellenar tu diario de aprendizaje.



www.bit.ly/cajamicrobitR1solucion

TARJETAS RETO 2

RETO 2

¿Cuántas veces hemos tenido un montón de pilas en casa y no estábamos seguros de cuáles estaban gastadas y cuáles no?

Vamos a programar nuestra tarjeta micro:bit para que nos diga si la pila está gastada o no y así poder deshacernos de ella de modo sostenible en un contenedor específico o bien seguir usándola.

En realidad con nuestra micro:bit solo podremos comprobar pilas de menos de 3 voltios, es decir, AA y AAA que son las más usadas en casa. Necesitaremos dos cables de tipo cocodrilo.






CONJETURA

Antes de empezar a programar nuestra tarjeta en Makecode vamos a estudiar cómo es el razonamiento que debe seguir la tarjeta para hacer lo que queremos que haga.

Rellenad la siguiente ficha con las partes de la tarea y las relaciones lógicas entre esas partes. Se trata de hacer una conjetura o una especie de hipótesis de lo que creemos que debe ocurrir. Podéis rellenar solo las casillas que necesitéis y tachar las que no o incluir nuevas relaciones usando flechas.

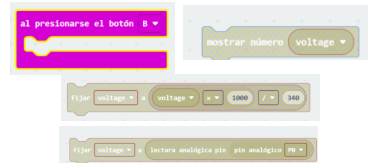
PISTA 1

- Si la medida es ___, entonces la tarjeta deberá... (mostrar un icono, una medida, hacer un sonido, etc.)
- Qué consideramos como pila gastada, es decir, qué medida es la que indica que está gastada
- Qué tipo de instrucción debemos programar para que la tarjeta haga una u otra cosa
- Qué tipo de mensaje debe dar el programa para que se entienda sin dificultad el estado de la pila.

voltage		analog value
1V		341
1,2 V		409
1,3 V		444
1,4 V		478
1.5 V		512

PISTA 2

Puedes usar estos bloques de makecode para hacer la programación, pero la relación entre ellos la tenéis que encontrar en equipo. No es el único modo



www.bit.ly/cajamicrobitR2P2

EXPLICACIÓN

En el siguiente enlace tienes el código completo de cómo hacer que tu MICROBIT se comporte como un sensor de luz, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. No olvides rellenar tu diario de aprendizaje



www.bit.ly/cajamicrobitR2solucion

OTRAS SOLUCIONES

En estos vídeos hay soluciones alternativas por si la que os propongo os parece algo complicada



www.bit.ly/batteryttestermicrobit2



www.bit.ly/batteryttestermicrobit

TARJETAS RETO 3

RETO 3

¿A que sería divertido y útil que nuestra planta nos dijera cuándo necesita agua y cuándo no? Vamos a programar nuestra tarjeta micro:bit para que mida las necesidades de agua de nuestra planta usando el sensor de humedad. Así podremos ahorrar un montón de agua y la planta siempre estará perfecta.

Necesitaremos, además de nuestra micro:bit, dos cables tipo "cocodrilo", dos clavos metálicos... y una planta.

CONJETURA

Antes de empezar a programar nuestra tarjeta en Makecode vamos a estudiar cómo es el razonamiento que debe seguir la tarjeta para hacer lo que queremos que haga.

Rellenad la siguiente ficha con las partes de la tarea y las relaciones lógicas entre esas partes. ¿Cómo puede "saber" la tarjeta micro:bit que la tierra está más seca o más húmeda, si básicamente no hace otra cosa que conducir la electricidad?

PISTA 1

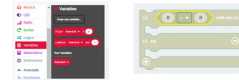
- Cómo podríamos medir la humedad de un terreno
- Por qué deducimos que la tierra está más seca o más húmeda
- Piensa en cables, clavos, etc.
- Qué consideramos mucha o poca humedad...
- Si hay poca humedad, entonces la tarjeta deberá...



www.bit.ly/cajamicrobitR3P1

PISTA 2

Puedes usar estos bloques de makecode para hacer la programación, pero la relación entre ellos la tenéis que encontrar en equipo.



www.bit.ly/cajamicrobitR3P2

EXPLICACIÓN

En el siguiente enlace tienes el código completo de cómo hacer que tu MICROBIT se comporte como un sensor de luz, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. No olvides rellenar tu diario de aprendizaje



www.bit.ly/cajamicrobitR3solucion

TARJETAS RETO 4

RETO 4

Sería muy útil que nuestro sistema de calefacción/refrigeración para una habitación, por ejemplo, funcionara de modo autónomo encendiéndose y apagándose cuando fuese necesario. Todavía no podemos hacer eso, pero sí podemos programar la micro:bit para que nos informe de cuando la temperatura de un lugar sube o baja y poder actuar en consecuencia. ¿Para qué más cosas podría servir esta información?

CONJETURA

Antes de empezar a programar nuestra tarjeta en Makecode vamos a estudiar cómo es el razonamiento que debe seguir la tarjeta para hacer lo que queremos que haga.

Rellenad la ficha con las partes de la tarea y las relaciones lógicas entre esas partes. Se trata de hacer una conjetura o una especie de hipótesis de lo que creemos que debe ocurrir. Podéis rellenar solo las casillas que necesitáis y tachar las que no o incluir nuevas relaciones usando flechas.

PISTA 1

- La tarjeta debe medir temperatura según la escala...
- Si la temperatura es demasiado alta, entonces la tarjeta deberá...
- Si la temperatura es demasiado baja, entonces la tarjeta deberá...
- Qué consideramos alta o baja temperatura

Lo que se considere una temperatura más o menos deseable puede debatirse antes de programar ya que va depender del contexto. En otras palabras, la temperatura deseable de una oficina, una aula o un hospital probablemente son diferentes.

PISTA 2

- Puedes usar estos bloques para hacer la programación, pero la relación entre ellos la tenéis que encontrar en equipo.



www.bit.ly/cajamicrobitR4P1

EXPLICACIÓN

En el siguiente enlace tienes el código completo de cómo hacer que tu MICROBIT se comporte como un termómetro, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. No olvides rellenar tu diario de aprendizaje



www.bit.ly/cajamicrobitR4solucion

TARJETAS RETO 5

RETO 5

Sería muy práctico tener un medidor de exceso de ruido ambiental. Podría usarse en una clase, en un lugar donde el fuera importante como una biblioteca o un hospital. Seguro que se nos ocurren un montón de aplicaciones útiles de un sensor de exceso de ruido. Vamos a programar nuestra MICRO:BIT para que detecte el exceso de ruido y nos avise de algún modo o ejecute alguna acción relacionada.

CONJETURA

Antes de empezar a programar nuestra tarjeta en Makecode vamos a estudiar cómo es el razonamiento que debe seguir la tarjeta para hacer lo que queremos que haga.

Rellenad la siguiente ficha con las partes de la tarea y las relaciones lógicas entre esas partes. Se trata de hacer una conjetura o una especie de hipótesis de lo que creemos que debe ocurrir. Podéis rellenar solo las casillas que necesitéis y tachar las que no o incluir nuevas relaciones usando flechas.

PISTA 1

- Si hay demasiado ruido, entonces la tarjeta deberá...
- Si el nivel de ruido es saludable, entonces la tarjeta deberá...
- Qué consideramos demasiado ruido o una cantidad de ruido aceptable

Lo que se considere un nivel de ruido más o menos deseable puede debatirse antes de programar ya que va depender del contexto. En otras palabras, el ruido ambiental deseable de una biblioteca, una aula o un hospital probablemente son diferentes.

PISTA 2

Puedes usar estos bloques de makecode para hacer la programación, pero la relación entre ellos la tenéis que encontrar en equipo.



www.bit.ly/cajamicrobitR4P2

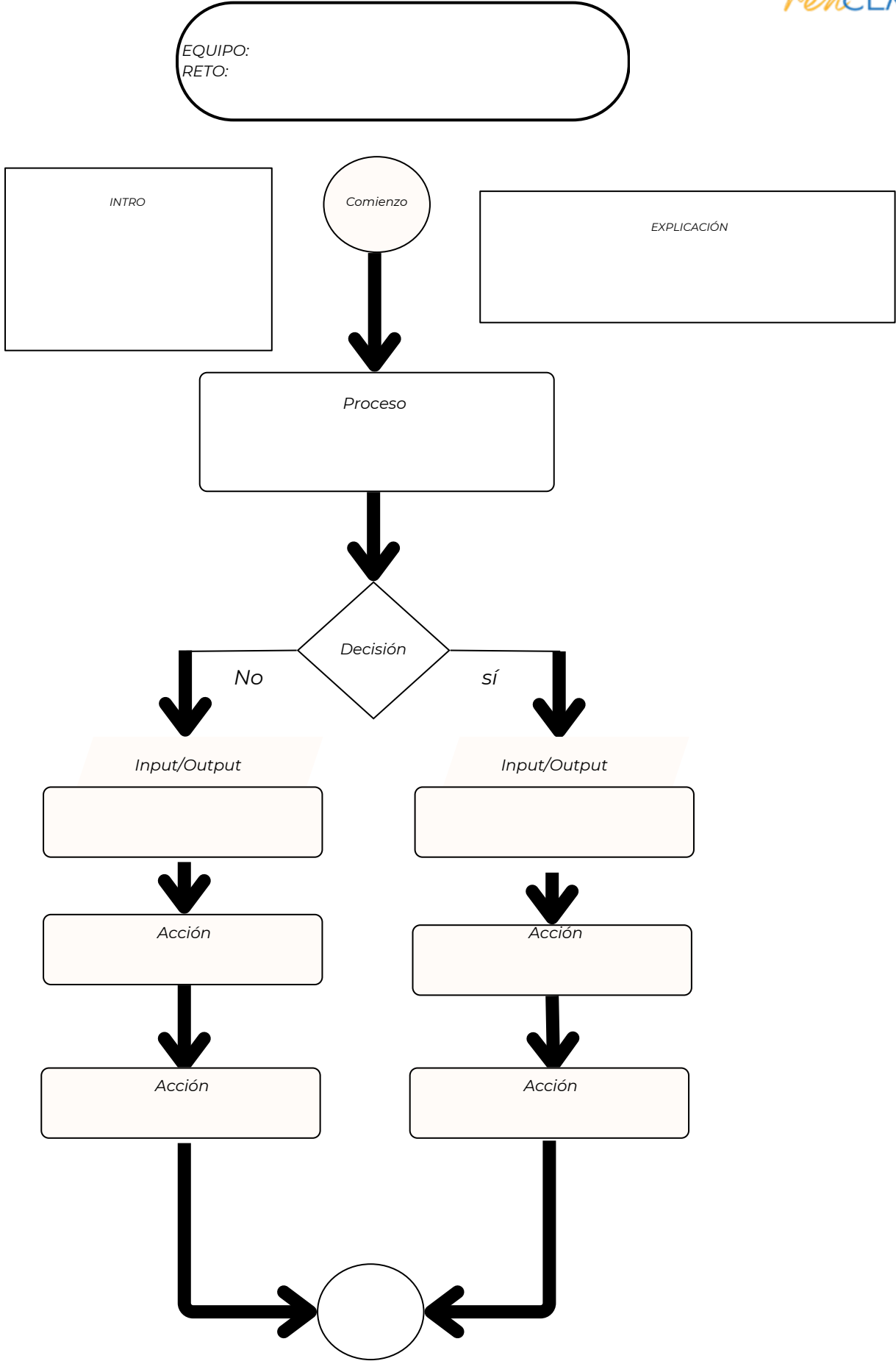
EXPLICACIÓN

En el siguiente enlace tienes el código completo de cómo hacer que tu MICROBIT se comporte como un sensor de luz, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. No olvides rellenar tu diario de aprendizaje



www.bit.ly/cajamicrobitR5solucion

RELACIONES LÓGICAS



EQUIPO:
RETO:

INTRO

Comienzo

EXPLICACIÓN

Proceso

Decisión

No

Sí

Input/Output

Input/Output

Acción

Acción

Acción

Acción



Diario de Aprendizaje

Reto:	
¿Qué ideas o saberes han sido los más importantes de este reto?	
Algo que no sabía...	
Algo que todavía no he entendido...	
¿Qué me ha costado más?	
¿Qué me ha resultado más fácil?	
¿En qué he necesitado más ayuda?	
¿Qué tal ha resultado el trabajo en equipo?	
¿He alcanzado los objetivos del proyecto? (¿Sí? ¿No? ¿A medias?)	
¿Este reto tiene algo que ver con mi vida o con el mundo actual?	

