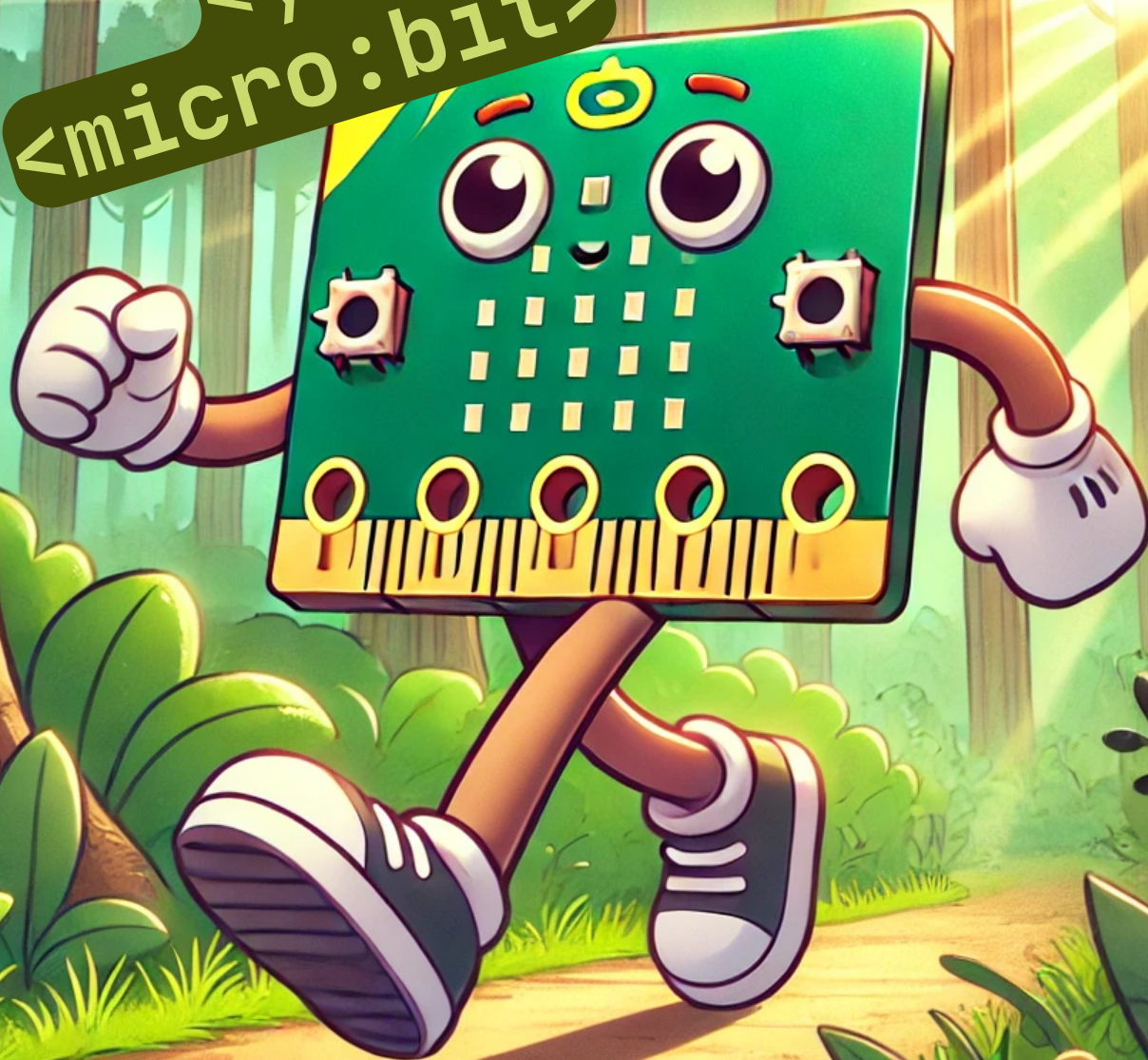


<salud>

<y>

<micro:bit>





SALUD Y MICRO:BIT

¿UNA CAJA DE APRENDIZAJE? ¿QUÉ ES ESO?

La metodología de cajas de aprendizaje es un enfoque educativo que utiliza cajas llenas de retos, recursos y materiales didácticos para trabajar un tema específico. Estas cajas pueden contener libros, actividades, juegos, herramientas interactivas y recursos digitales, diseñados para hacer el aprendizaje más práctico y atractivo.

¿DE QUÉ VA ESTA CAJA DE APRENDIZAJE?

En esta caja de aprendizaje vamos a poner en marcha nuestros conocimientos de **Biología**. Vamos a incidir en la parte de sostenibilidad utilizando las tarjetas **MICRO:BIT** para mejorar el uso de la energía. Se trata de aprender a programar la tarjeta usando [makecode](#) en proyectos relacionados con sostenibilidad, reducción de la contaminación y salud.

DESCRIPCIÓN Y FINALIDAD DE LOS APRENDIZAJES:

Esta actividad persigue diversas finalidades como aplicar a la vida nuestros conocimientos sobre sostenibilidad, sensibilización ciudadana, programación y pensamiento computacional.

SECUENCIA DE ACTIVIDADES:

En esta caja veremos cinco programaciones de la tarjeta que nos ayudarán a hacer un **uso más sostenible de la energía** y a usar el pensamiento computacional en proyectos relacionados con la salud y la sostenibilidad.

Aprenderemos a programar una tarjeta MICRO:BIT

- Para que la tarjeta detecte cuando hay suficiente luz natural y pueda apagarse la luz artificial para ahorrar electricidad y aumentar la vida útil de una bombilla.
- Para que nos diga si una pila está gastada o no y poder llevarla a un contenedor o continuar usándola.
- Para que nos informe de si nuestra planta necesita ser regada o no y ahorrar agua
- Para que nos informe de la temperatura de un lugar y gestionar si debemos abrir ventanas, poner en marcha un sistema de calefacción/refrigeración y ahorrar energía
- Para que nos informe del ruido ambiental y prevenir y reducir la contaminación acústica.

Interesante ¿Verdad?

Si todavía no has usado lenguaje de programación por bloques (tipo Scratch) o bien no sabes lo que es una tarjeta micro:bit, puedes empezar visitando este enlace con contenidos de ayuda

www.bit.ly/microbitayuda



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 necesitéis 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 MICROBIT 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 necesitáis 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.



PISTA 2

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



<https://makecode.microbit.org/S62371-19435-32607-18452>

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



<https://makecode.microbit.org/S70491-74965-55902-12888>

OTRAS SOLUCIONES

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



www.bit.ly/batterytestermicrobit2



www.bit.ly/batterytestermicrobit

TARJETAS RETO 3

RETO 3

Vamos a programar nuestra micro:bit como un medidor de pasos o podómetro. Usando el acelerómetro podremos medir el número de veces que se mueve y hacer una estimación del número de pasos que hemos dado.
¿Dónde vamos a situar la micro:bit? Necesitarás también un portapilas una vez que descargues el código en la tarjeta.

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

- Cómo sabe la micro:bit que hemos dado un paso
- Resolver dónde la ubicamos para que reconozca correctamente el movimiento
- Necesitaremos una variable que vaya guardando los pasos que damos, es decir, el número de veces que se mueve la micro:bit



PISTA 2



CONTADOR DE PASOS (PISTAS)
<https://makecode.microbit.org/S21600-05606-42820-66813>

EXPLICACIÓN

En el siguiente enlace tienes el código completo de cómo hacer que tu MICROBIT se comporte como un contador de pasos, pero recuerda que el reto no era exactamente ese. Tienes que intentar mejorarlo. Puedes ajustar mejor cuándo cuenta un paso, puedes estimar cuánta distancia has recorrido midiendo tu paso medio, ahorrar energía si solo muestra el número al tocar un botón, etc. No olvides rellenar tu diario de aprendizaje



<https://makecode.microbit.org/S67938-13496-07247-74937>

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 necesitéis 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

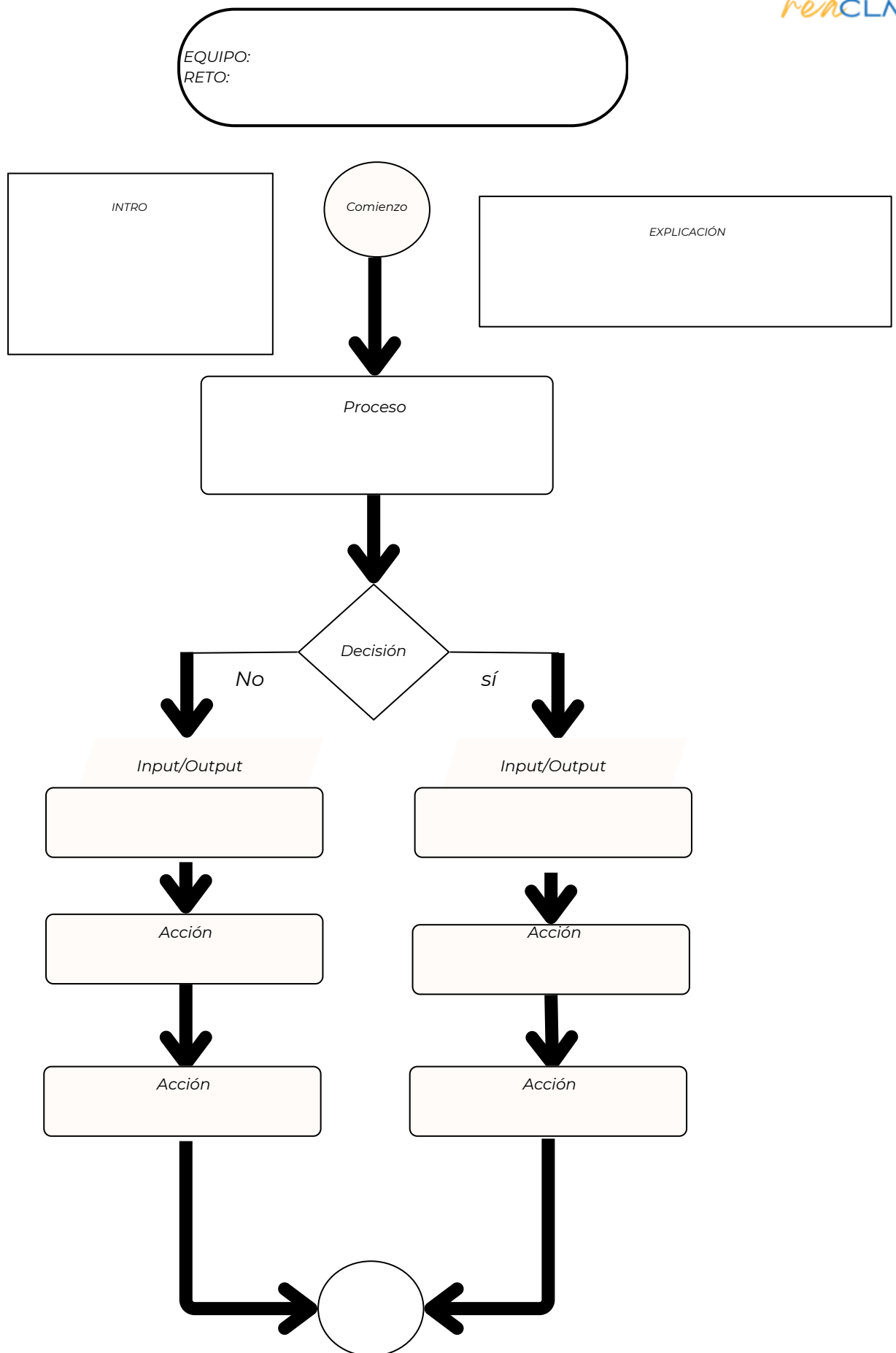
EXPLICACIÓN

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



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?	



Rúbrica de Evaluación para Retos de Cajas de Aprendizaje

Criterio	1 - Insuficiente	2 - Suficiente	3 - Notable	4 - Excelente
Entrega en tiempo y forma	No entrega la tarea o lo hace de manera incompleta y fuera de plazo.	Entrega la tarea con retraso, aunque está completa.	Entrega la tarea completa en plazo, pero con errores menores de formato o claridad.	Entrega la tarea completa, en plazo y siguiendo todos los requisitos de formato y claridad.
Documento de relaciones lógicas	No elabora el documento, o este es incomprensible y carece de lógica.	Elabora el documento, pero presenta errores importantes en las relaciones o no es claro.	Elabora el documento correctamente, con relaciones lógicas claras y solo pequeños errores.	Elabora un documento claro, completo y con relaciones lógicas bien fundamentadas.
Programación y uso de la Micro:bit	No programa la Micro:bit o el código no funciona y no está relacionado con el reto.	El código es funcional pero incompleto o tiene errores significativos.	El código funciona correctamente, está relacionado con el reto y es claro en su implementación.	El código es completamente funcional, eficiente y resuelve el reto de manera innovadora.
Precisión científica	La información presentada en relación al reto de sostenibilidad es incorrecta o muy incompleta.	Incluye información básica sobre sostenibilidad, pero con errores relevantes.	Incluye información científica correcta y relacionada con sostenibilidad, aunque no profundiza lo suficiente.	La información es precisa, completa y está científicamente fundamentada en relación a la sostenibilidad.
Creatividad en el diseño del proyecto	El proyecto carece de originalidad y no utiliza adecuadamente las herramientas de programación y hardware.	Muestra algo de creatividad, pero es limitada en diseño o uso de herramientas.	El proyecto es creativo, bien diseñado y utiliza correctamente las herramientas de programación y hardware.	El proyecto es altamente creativo, visualmente atractivo y utiliza la programación y hardware de manera innovadora.



