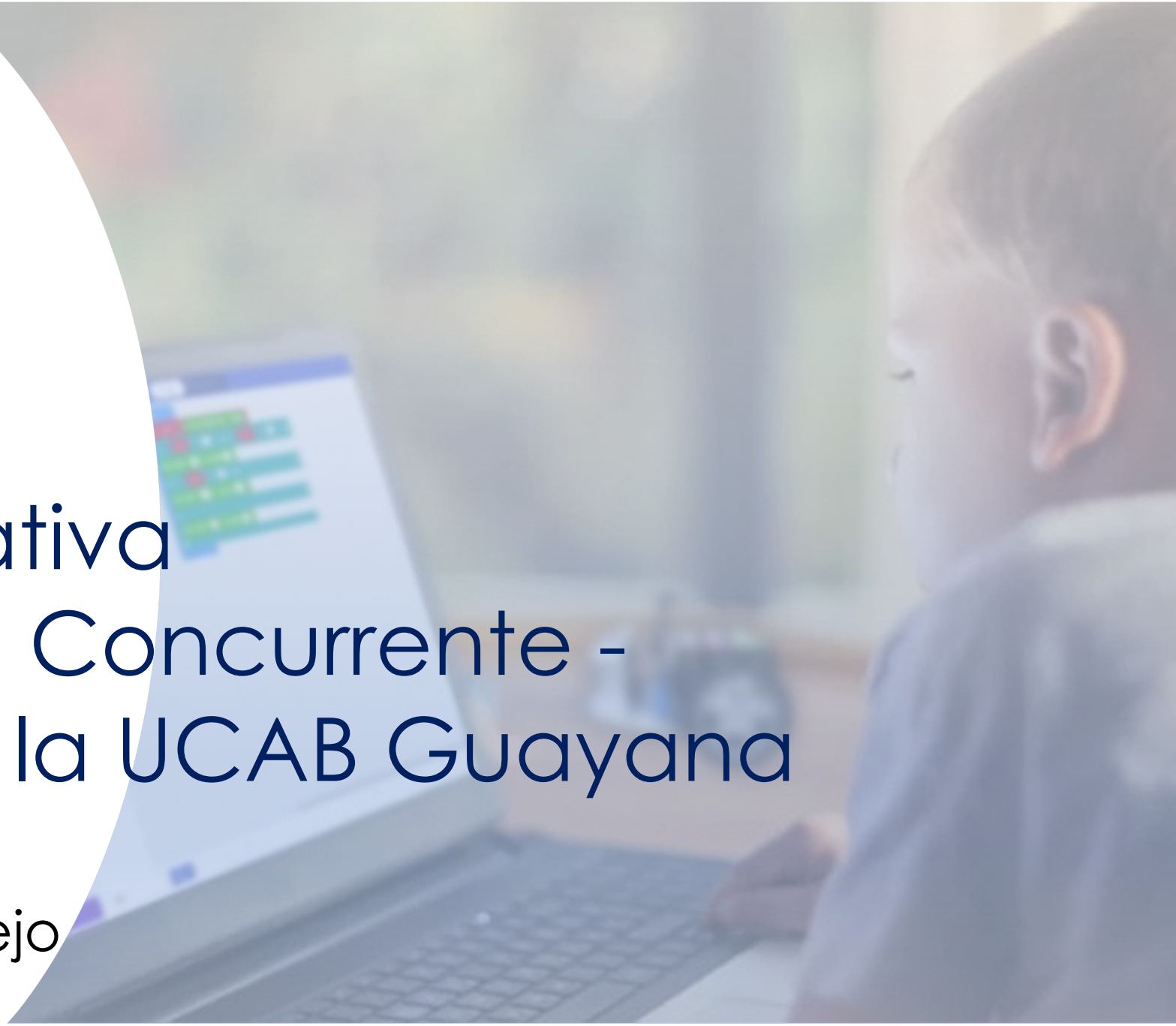
UCAB

Jornada de Educación 2022

Robótica Educativa y Programación Concurrente - Experiencias en la UCAB Guayana



Jesús Lárez - Paulina Espejo

Junio 2022

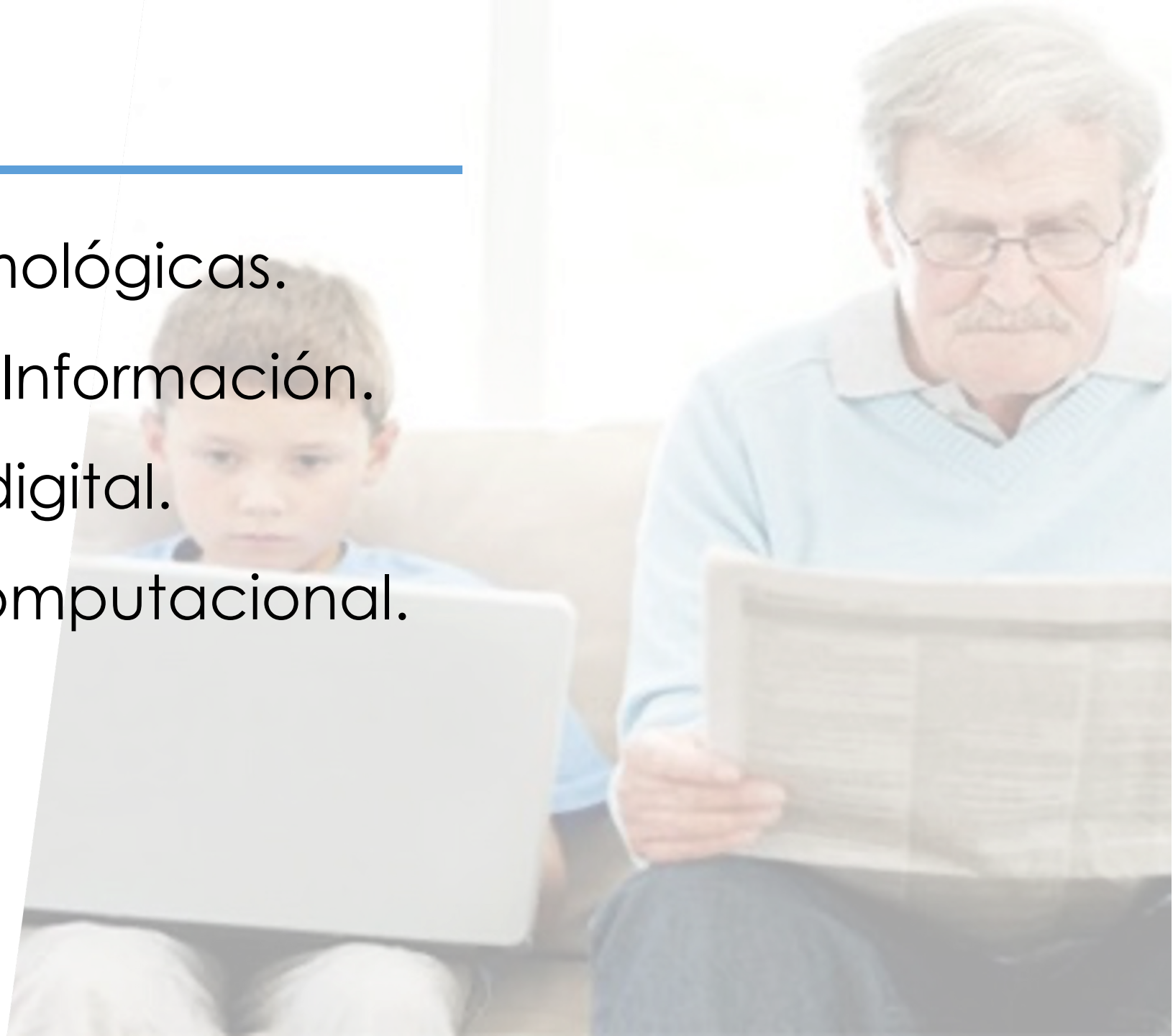
Agenda

- ✓ Motivación.
- ✓ Robótica Educativa.
- ✓ Programación Concurrente.
- ✓ Experiencias con Robots basados en Arduino.
- ✓ Experiencias con Robots basados en Micro-bit.
- ✓ Discusión.
- ✓ Conclusiones y Trabajo Futuro.

Motivación

- Culturas epistemológicas.
- Sociedad de la Información.
- Alfabetización digital.
- Pensamiento computacional.

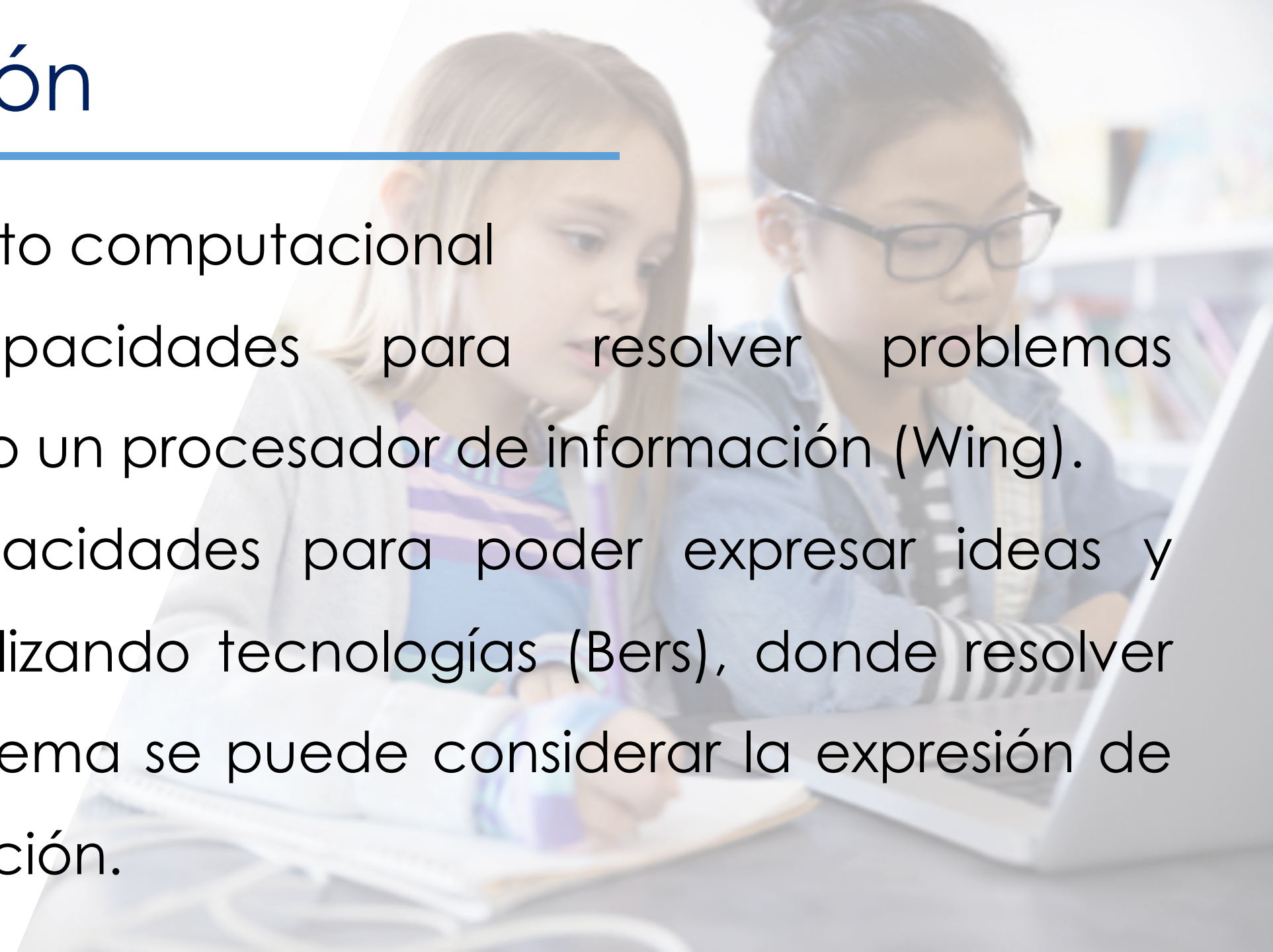
- Brecha digital.



Motivación

Pensamiento computacional

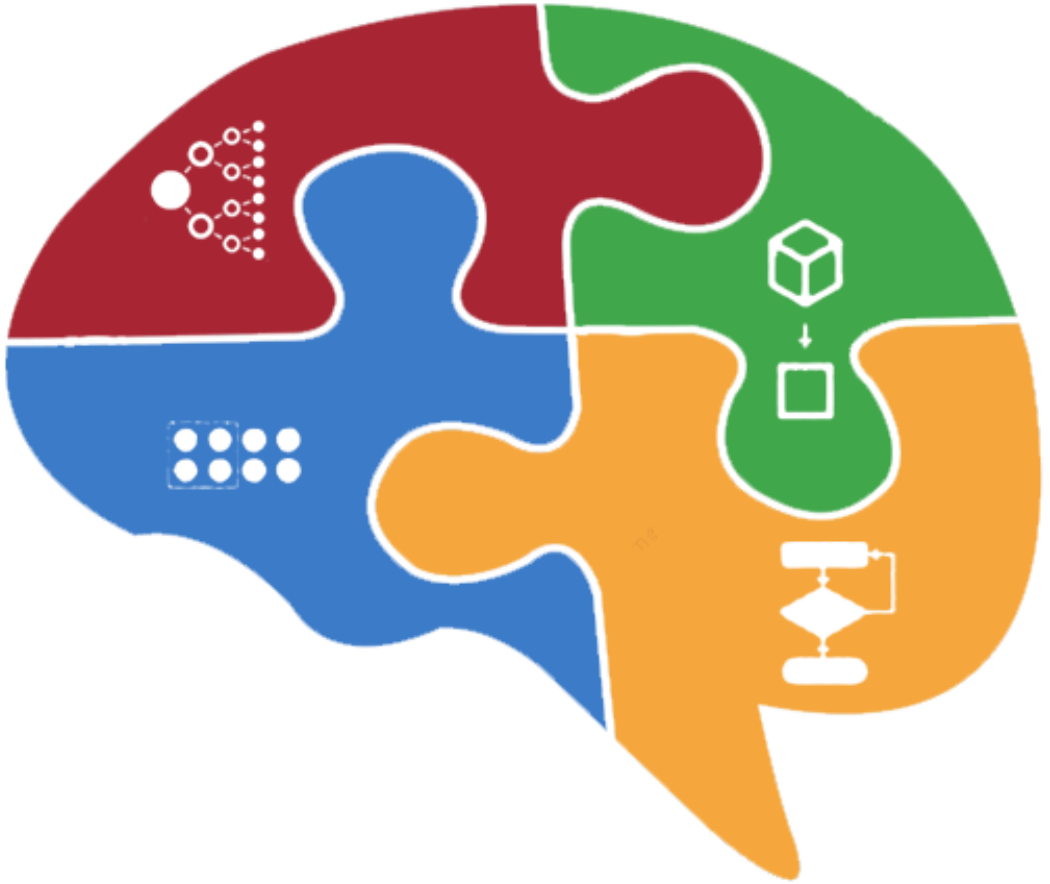
- Las capacidades para resolver problemas utilizando un procesador de información (Wing).
- Las capacidades para poder expresar ideas y crear utilizando tecnologías (Bers), donde resolver un problema se puede considerar la expresión de una solución.



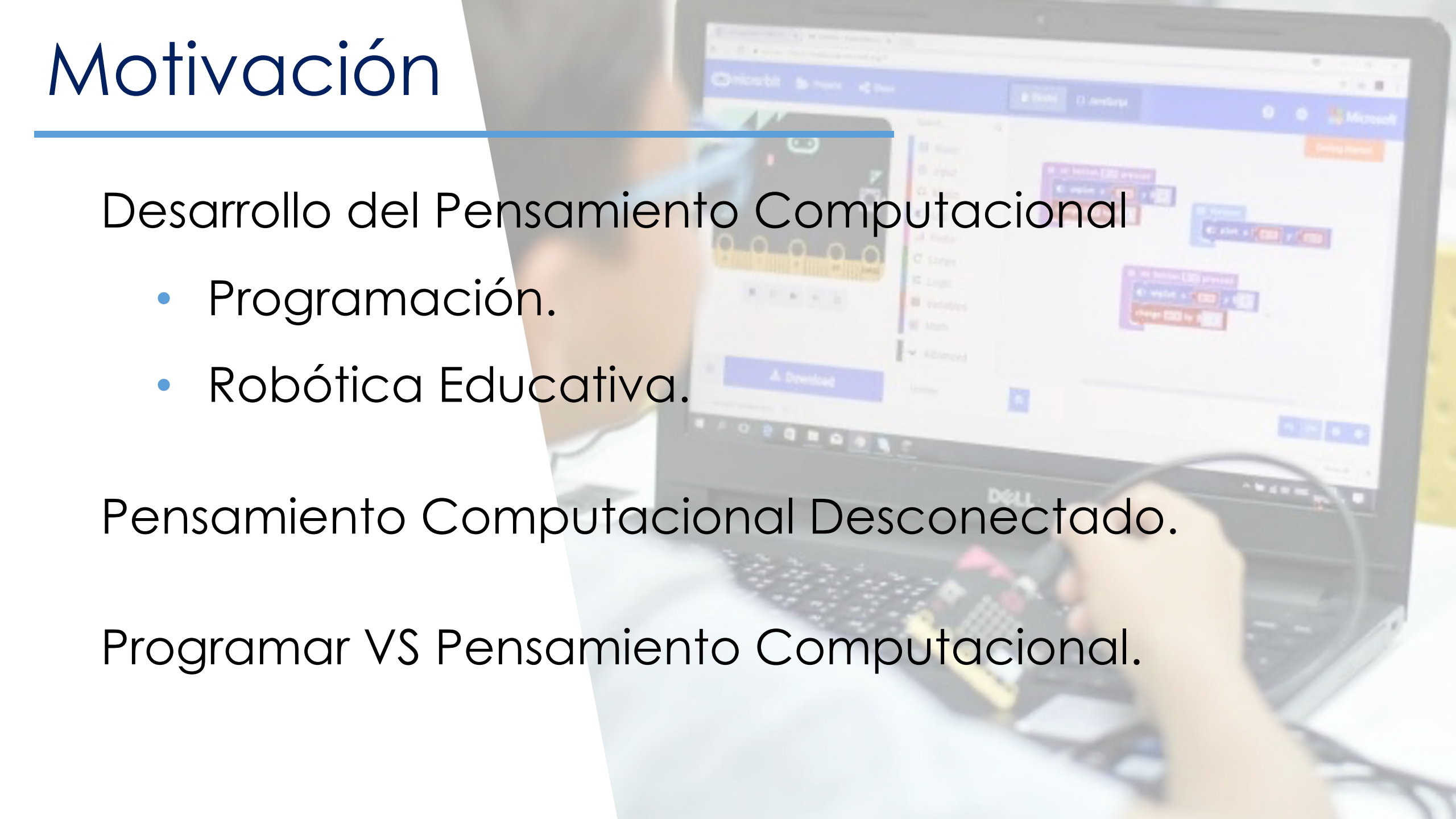
Motivación

Componentes del Pensamiento Computacional

- Abstracción.
- Descomposición.
- Patrones.
- Algoritmos.



Motivación



Desarrollo del Pensamiento Computacional

- Programación.
- Robótica Educativa.

Pensamiento Computacional Desconectado.

Programar VS Pensamiento Computacional.

Robótica Educativa

Un **robot** es "una máquina programable capaz de percibir y actuar en el mundo con cierta autonomía".

La **robótica educativa** es el desarrollo de propuestas educativas que incorporan el uso de robots en las aulas.

Robótica Educativa

Conceptos relacionados:

- Construcción de Robot.
- Programación de Robot.
- Pensamiento Computacional.

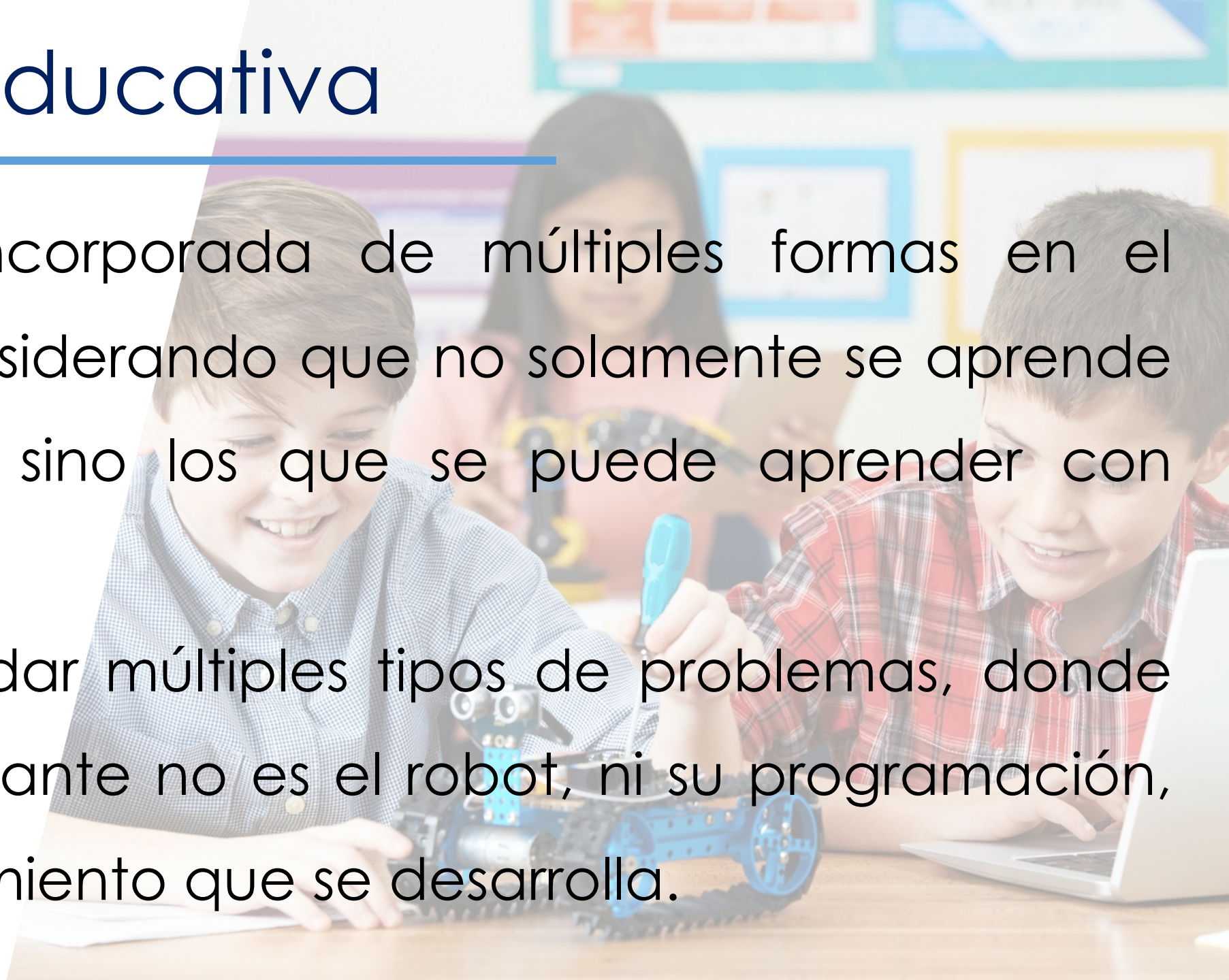
Consideraciones a tener en cuenta:

- Ventajas.
- Desafíos.

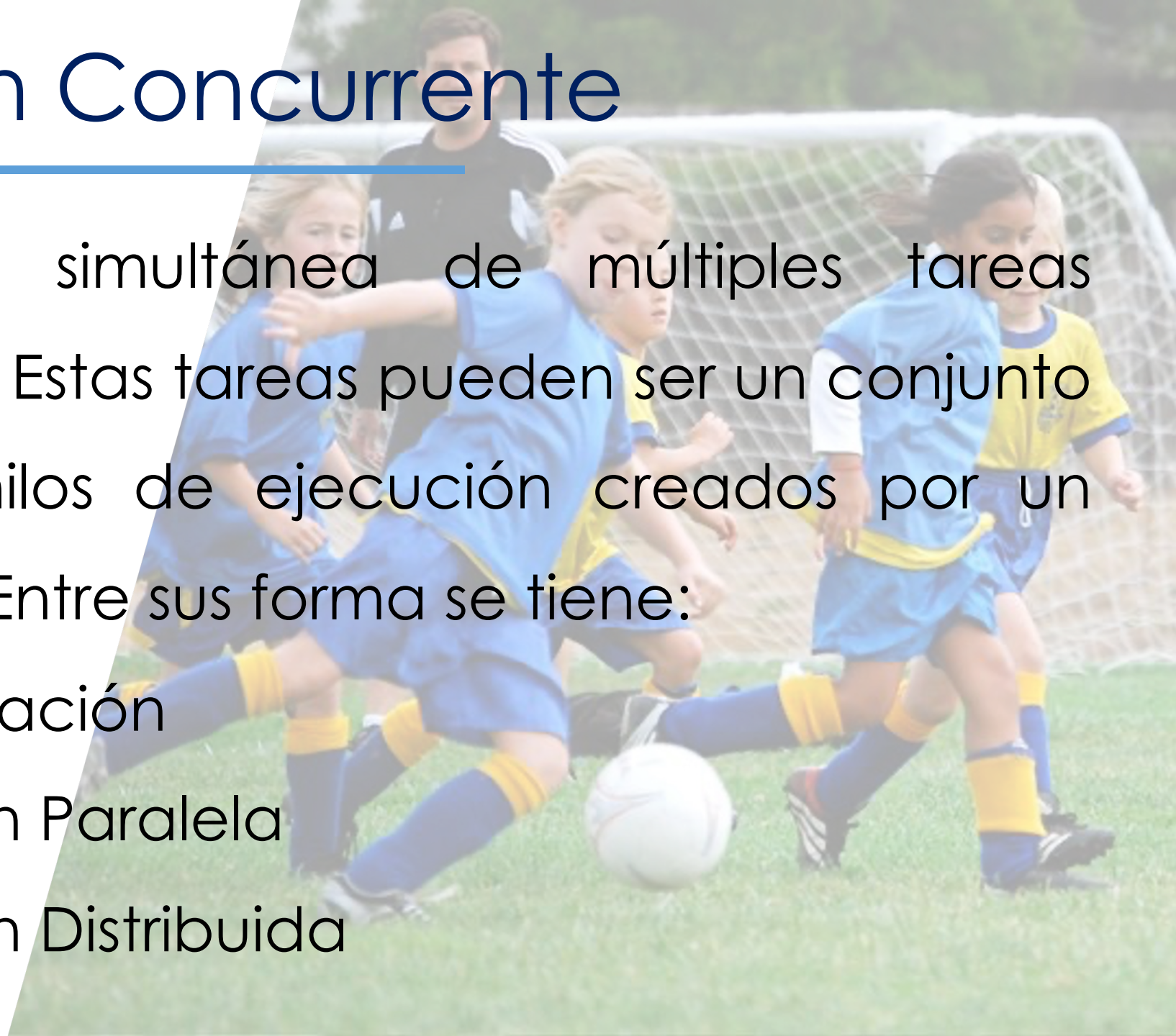
Robótica Educativa

Puede ser incorporada de múltiples formas en el currículo, considerando que no solamente se aprende de robótica, sino los que se puede aprender con robótica.

Permite abordar múltiples tipos de problemas, donde lo más importante no es el robot, ni su programación, sino el pensamiento que se desarrolla.



Programación Concurrente

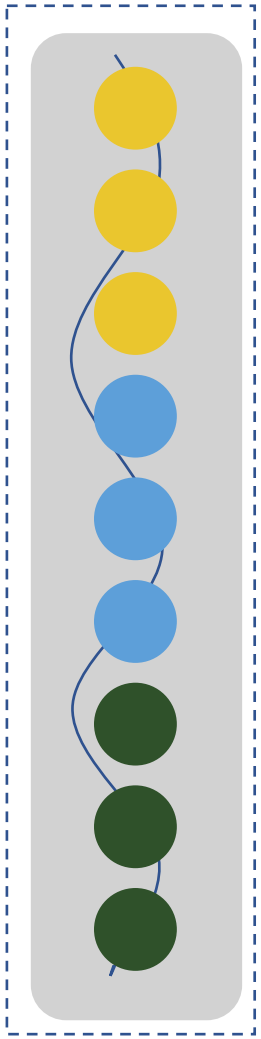


Es la ejecución simultánea de múltiples tareas interactivamente. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa. Entre sus forma se tiene:

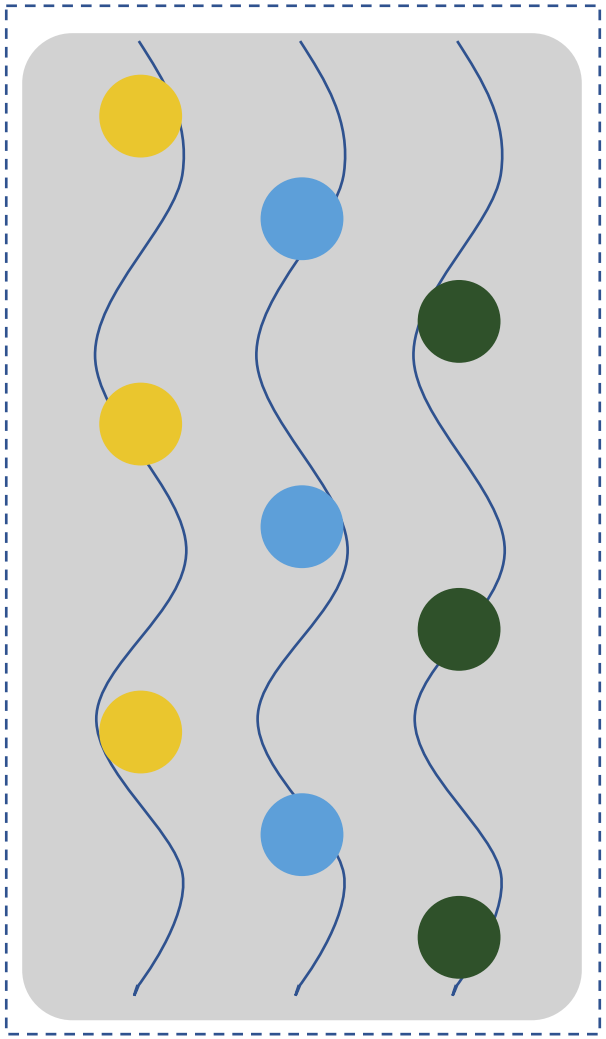
- Multiprogramación
- Programación Paralela
- Programación Distribuida

Programación Concurrente

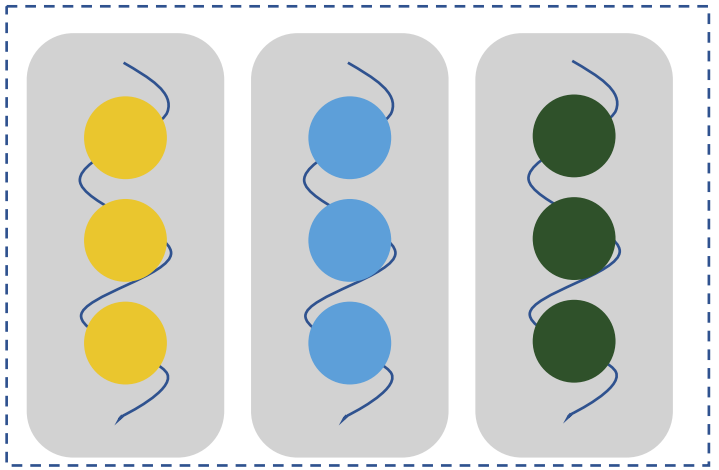
Secuencial



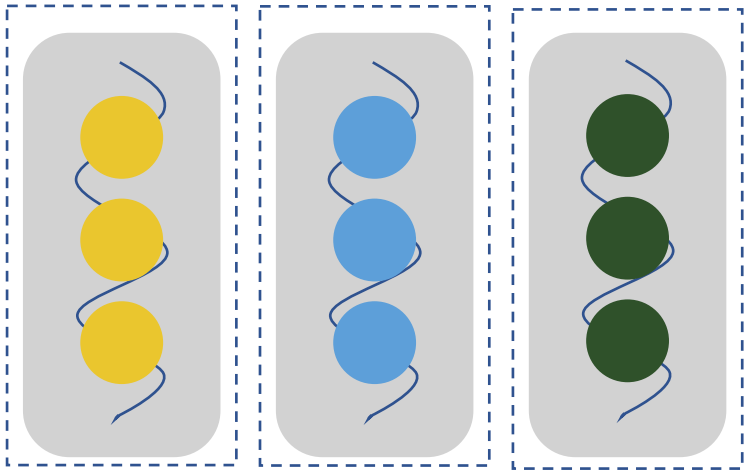
Multiprogramación



Paralelo

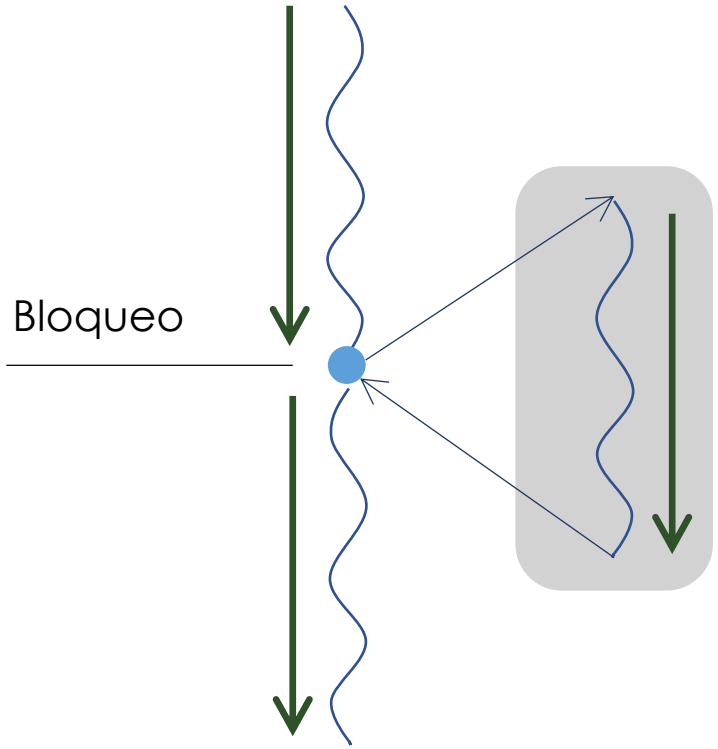


Distribuida

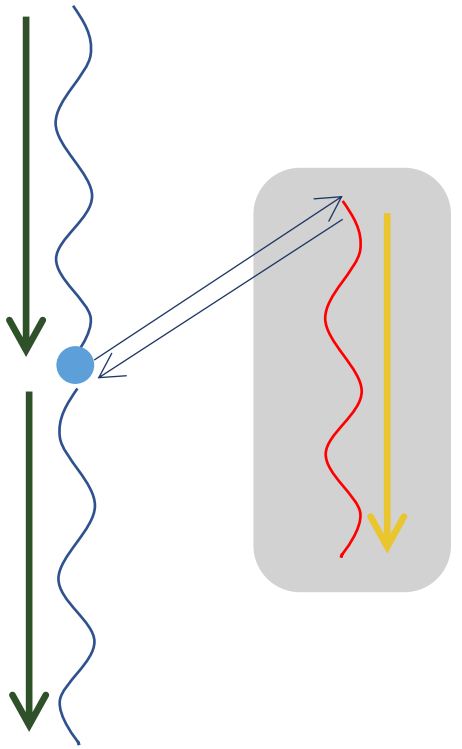


Programación Concurrente

Funciones bloqueantes



Funciones Asíncronas



Programación Concurrente

- Soporte:
  - Lenguaje.
  - Extensiones del lenguaje.
  - Librerías.
- Desafíos:
  - Diseños de soluciones ”paralelas”.
  - Sincronización.

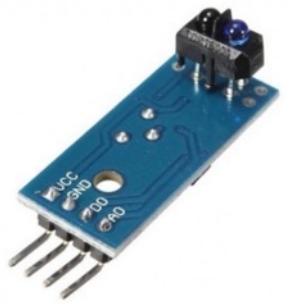
Robots Basados en Arduino

Arduino

Plataforma de desarrollo basada en una tarjeta electrónica que incorpora un microcontrolador re-programable y una serie de pines que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

Robots Basados en Arduino

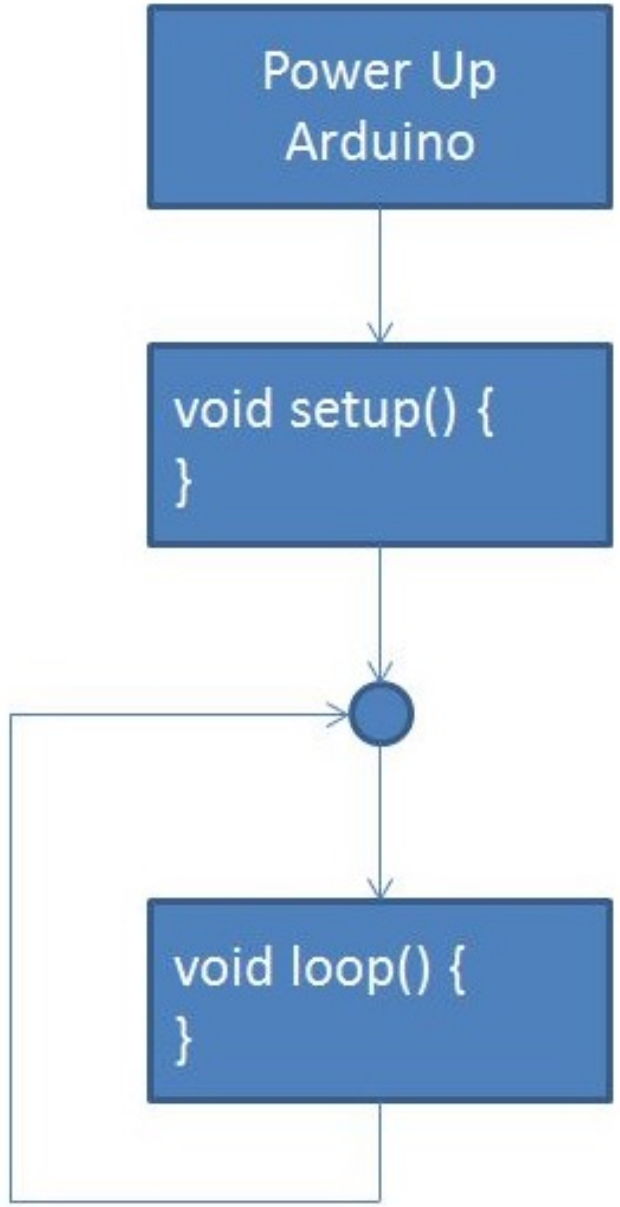
Arduino



Robots Basados en Arduino

Programación Arduino

- Funciones principales:
  - Setup()
  - Loop()
- Interrupciones.
- Serial o secuencial.
- Considerar funciones bloqueantes.



Robot Basados en Micro:Bit

“Micro Bit es una minicomputadora que nació como una colaboración entre la BBC y varias compañías tecnológicas para enseñar a niños - y, sobre todo, niñas - de Reino Unido a programar”

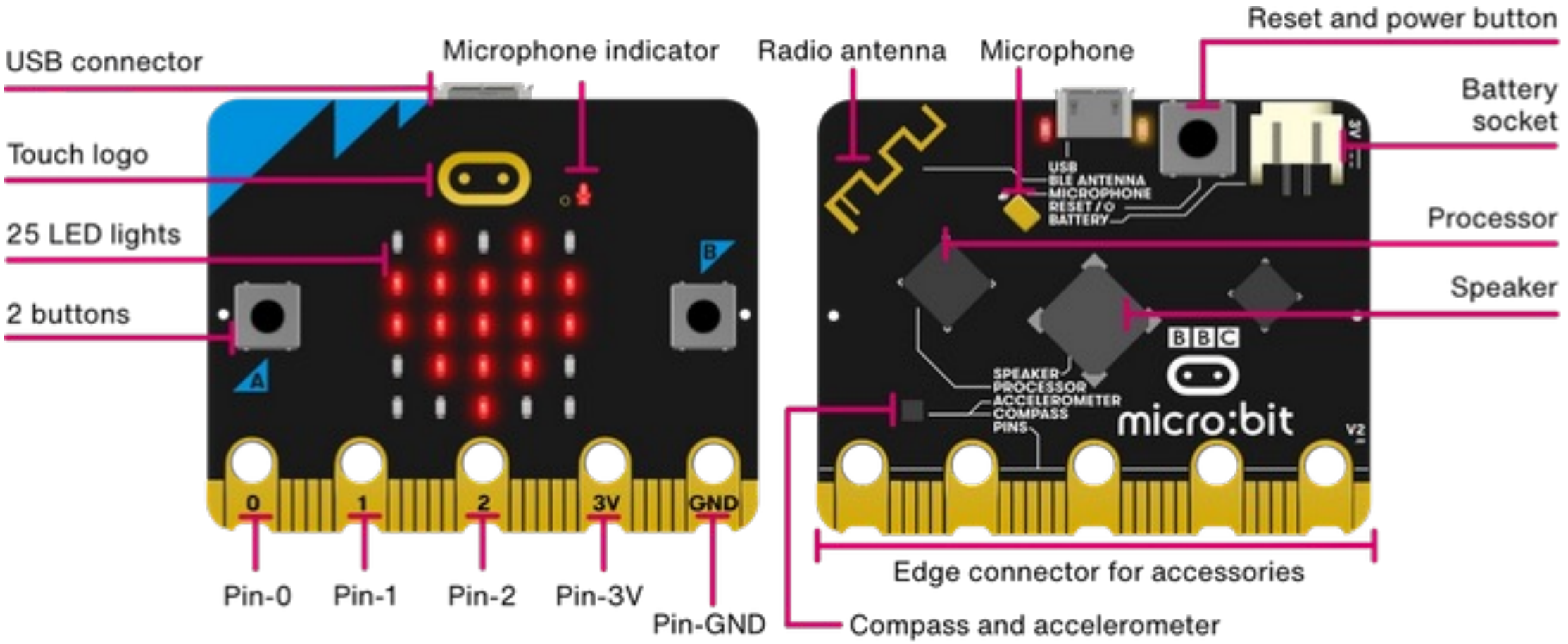
Robot Basados en Micro:Bit

¿Qué es una Micro Bit?

Es una tarjeta de circuitos del tamaño de la palma de una mano con una serie de 25 ledes y un comunicación Bluetooth para conexión inalámbrica. Incluye botones, múltiples sensores y la posibilidad de comunicaciones entre ellas, abriendo un espectro de posibilidades.

Robot Basados en Micro:Bit

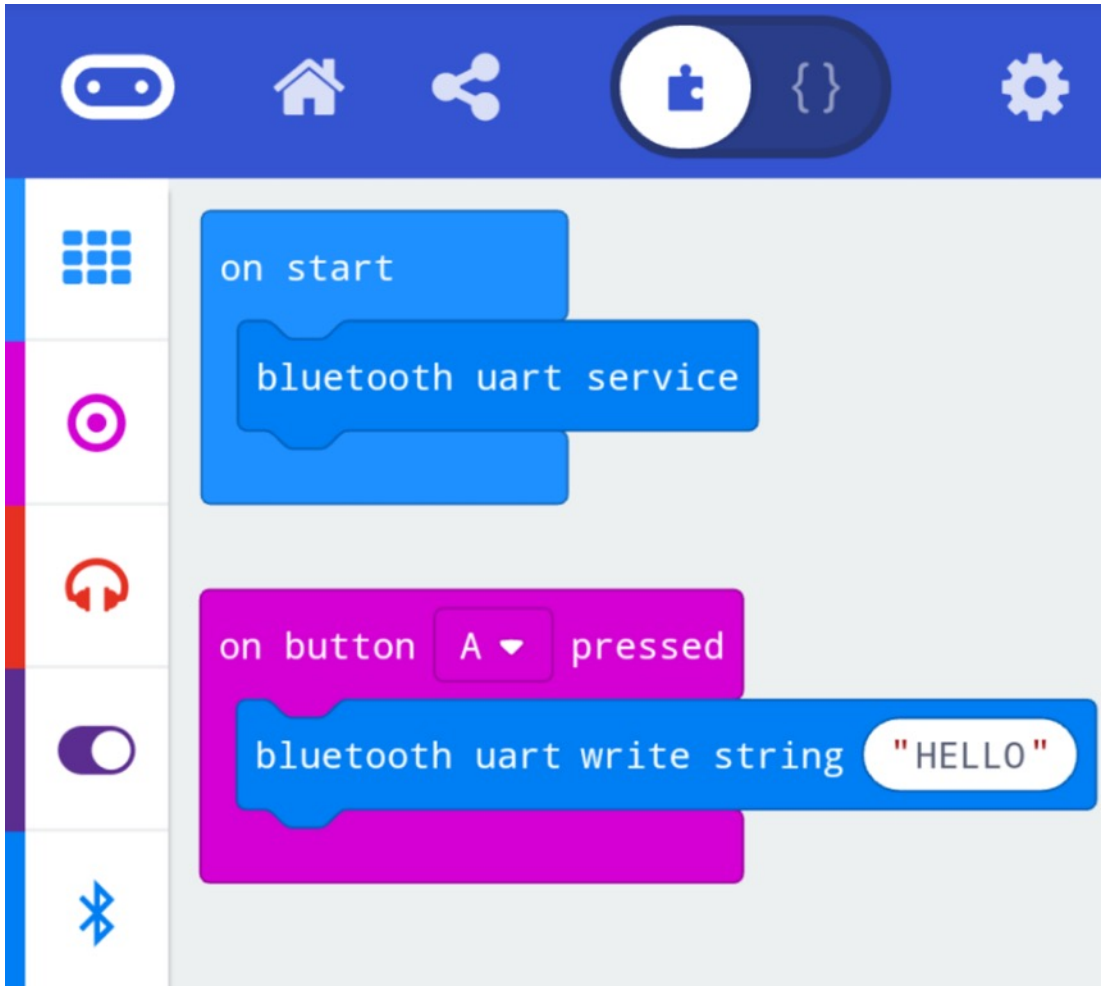
Micro:Bit



Robots Basados en Micro:Bit

Programación

- Event-Drive
- Concurrencia
  - Funciones asíncronas
  - Fibras



Robots Basados en Micro:Bit

Cutebot

- Robot diferencial.
- Sensores:
  - Dos sensores óptico reflectivo.
  - Ultrasónico.

Experiencia

Se plantea el reto siguiente:

El robot hace:

- Recibir una velocidad por radio de un control.
- Seguir a esa velocidad una línea.

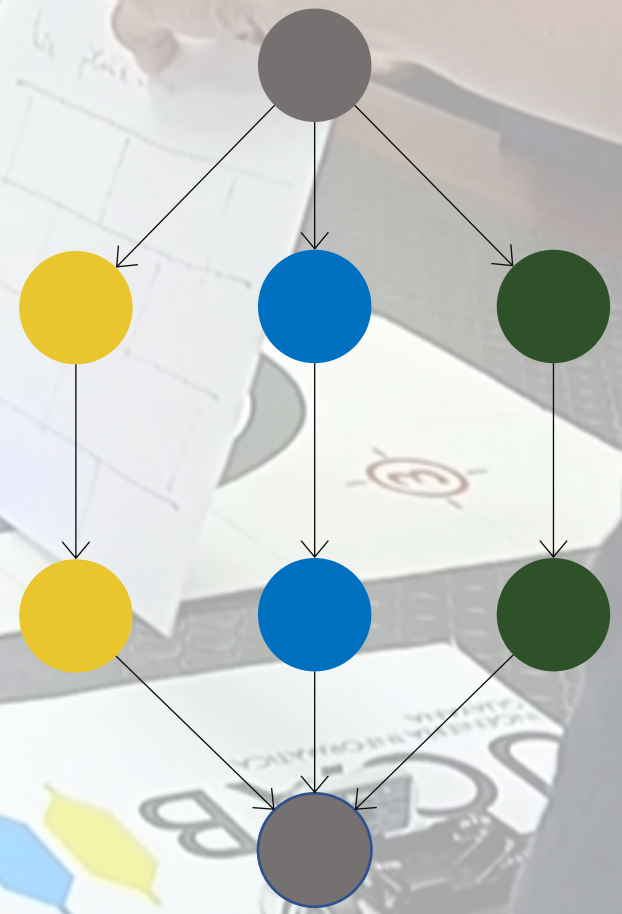
Se pide modificar la programación para:

- Parar si encuentra un obstáculo.
- Continuar si se quita el obstáculo.

Experiencia

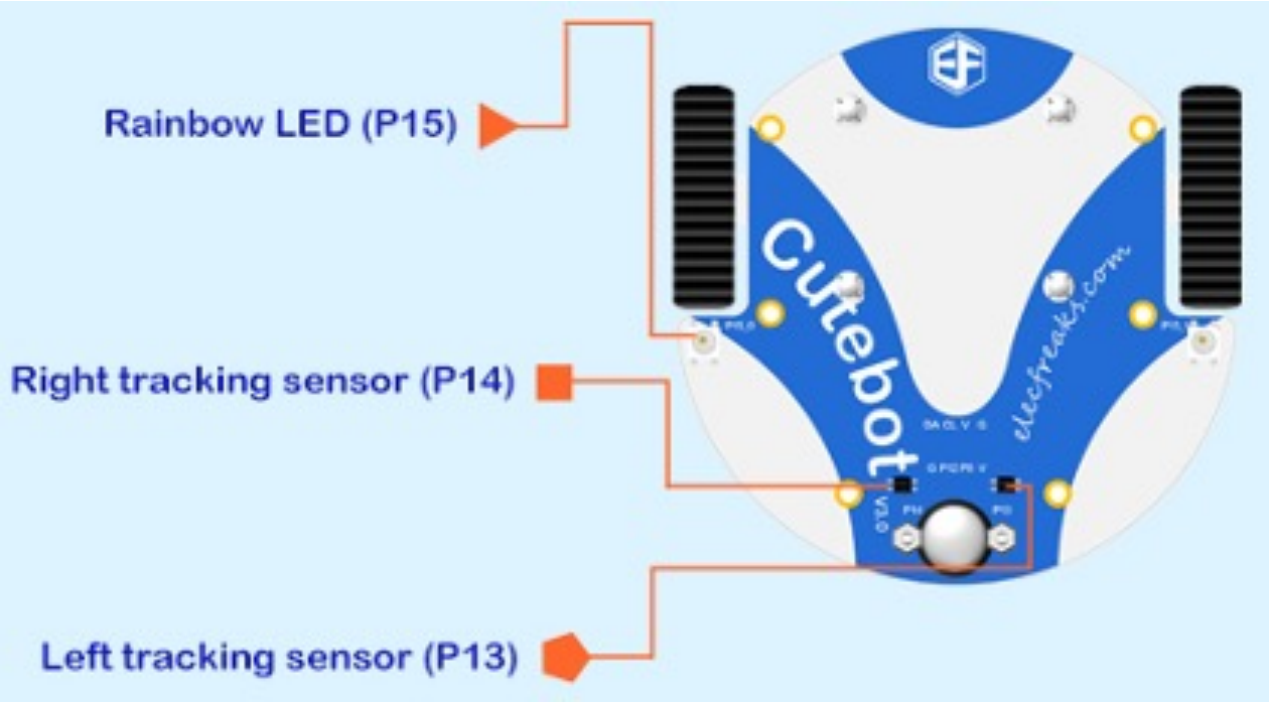
Etapas en la resolución del problema

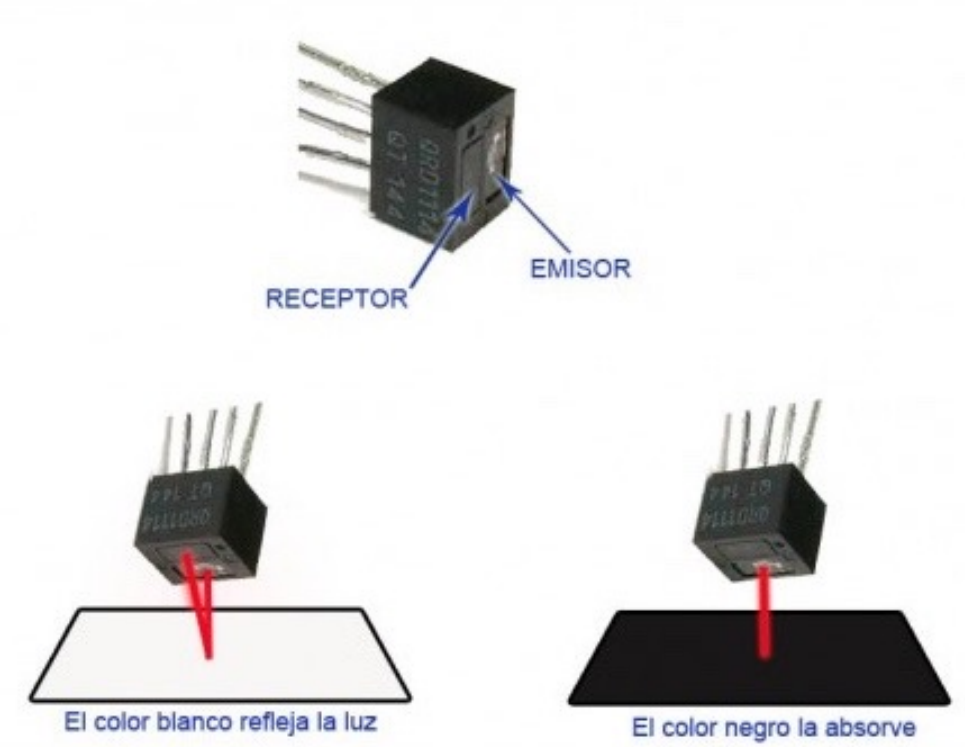
- Comprender el problema
- Generar alternativas solución
- Evaluar alternativas solución
- Seleccionar la mejor solución
- Implementar la solución
- Evaluar la solución



Experiencia

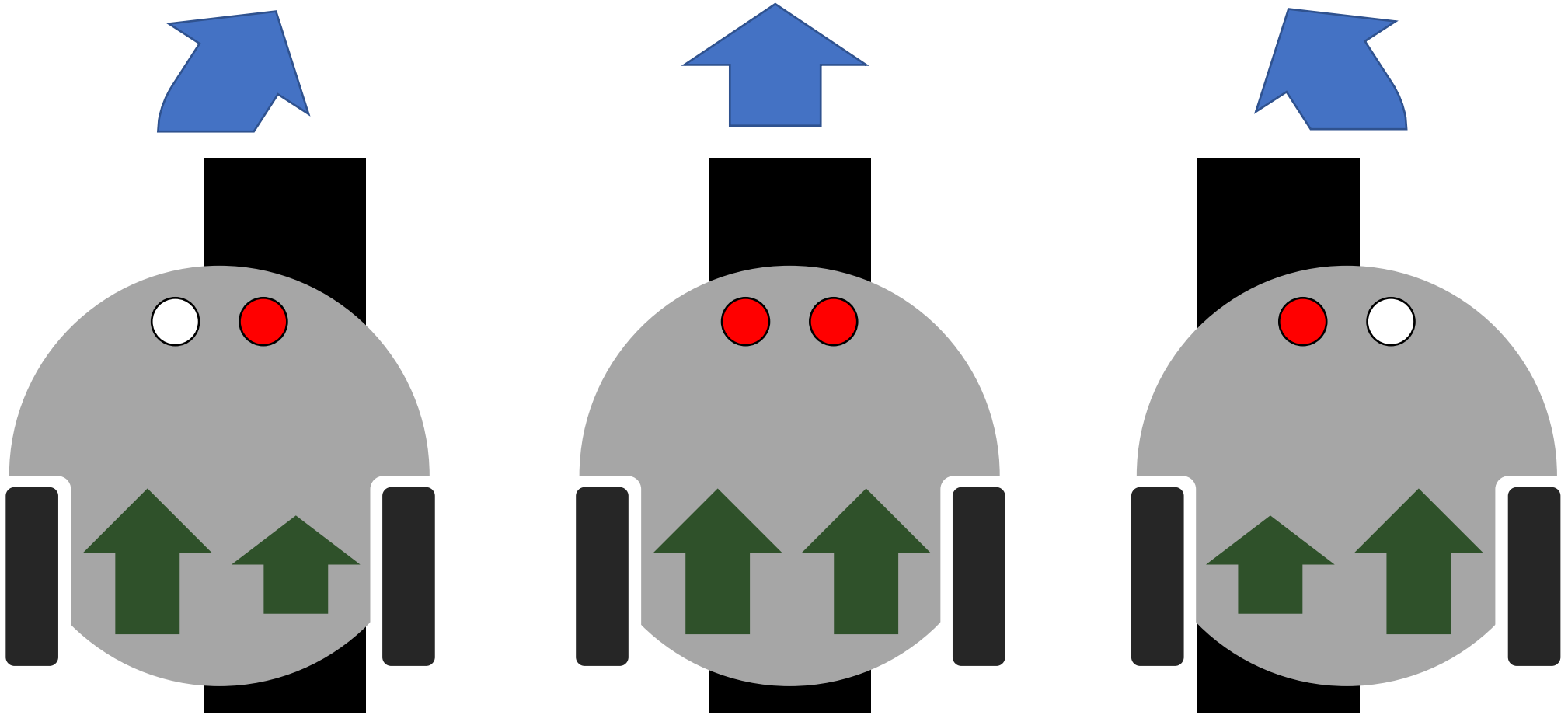
Seguir una línea





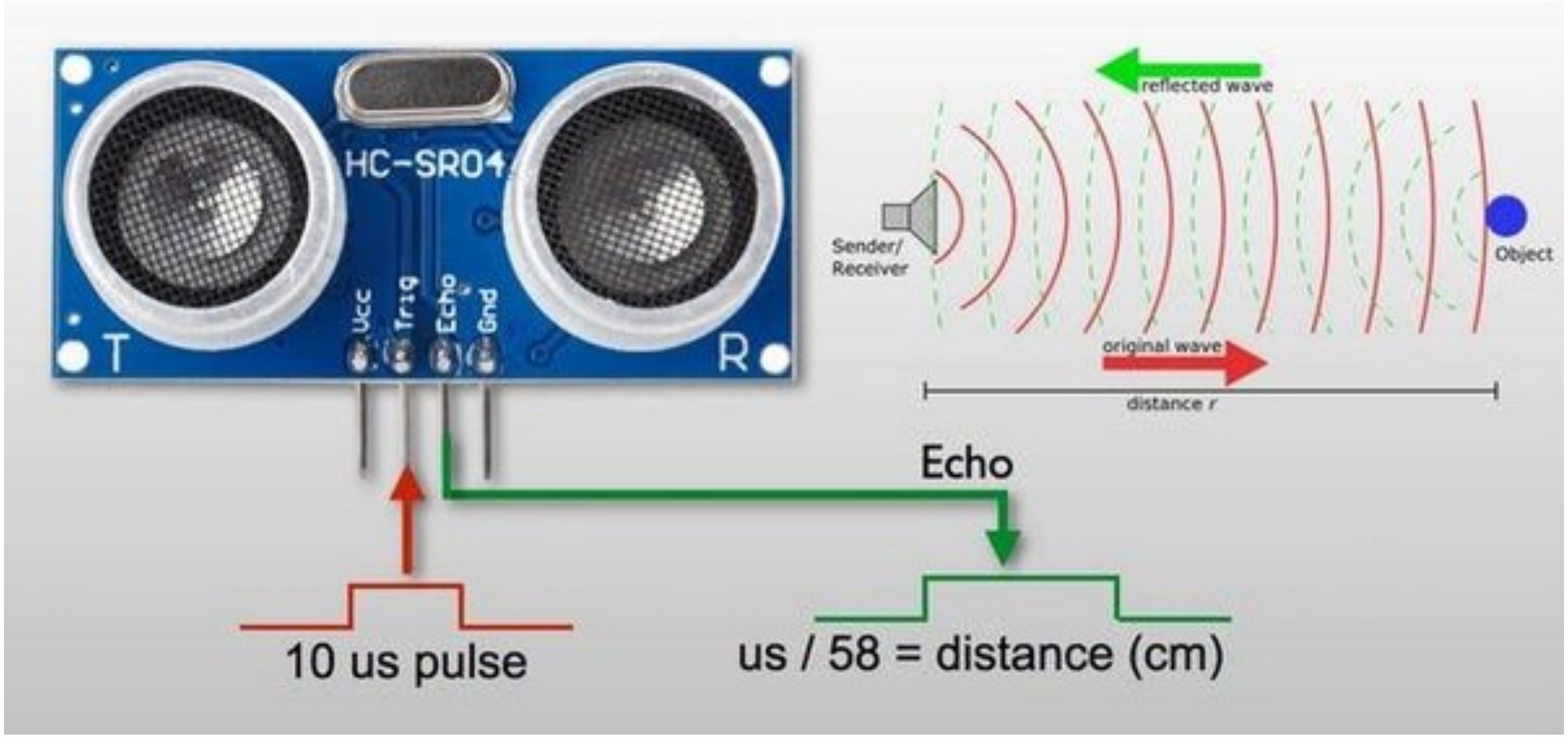
Experiencia

Seguir una línea

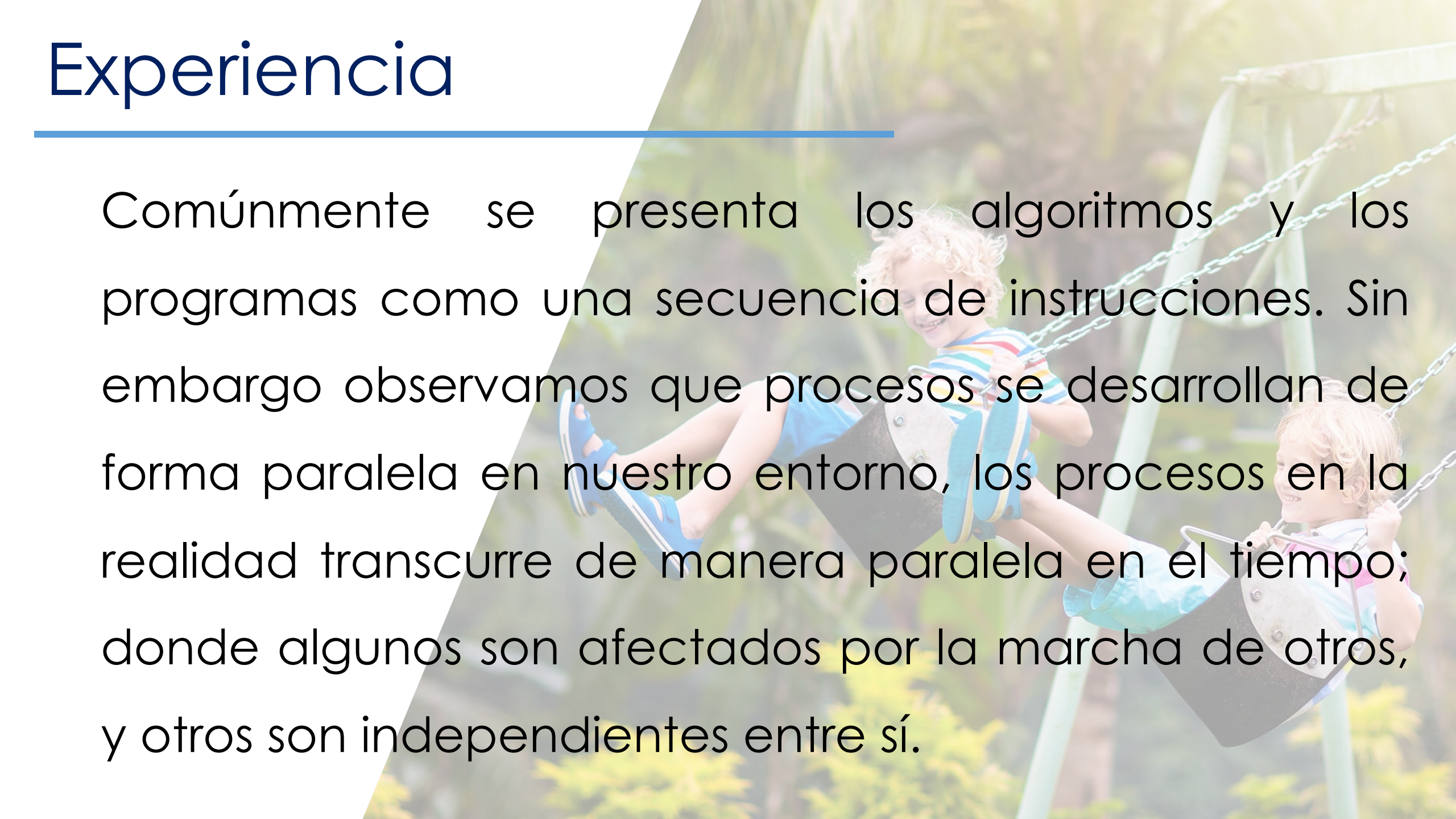


Experiencia

Calcular la distancia a un objeto

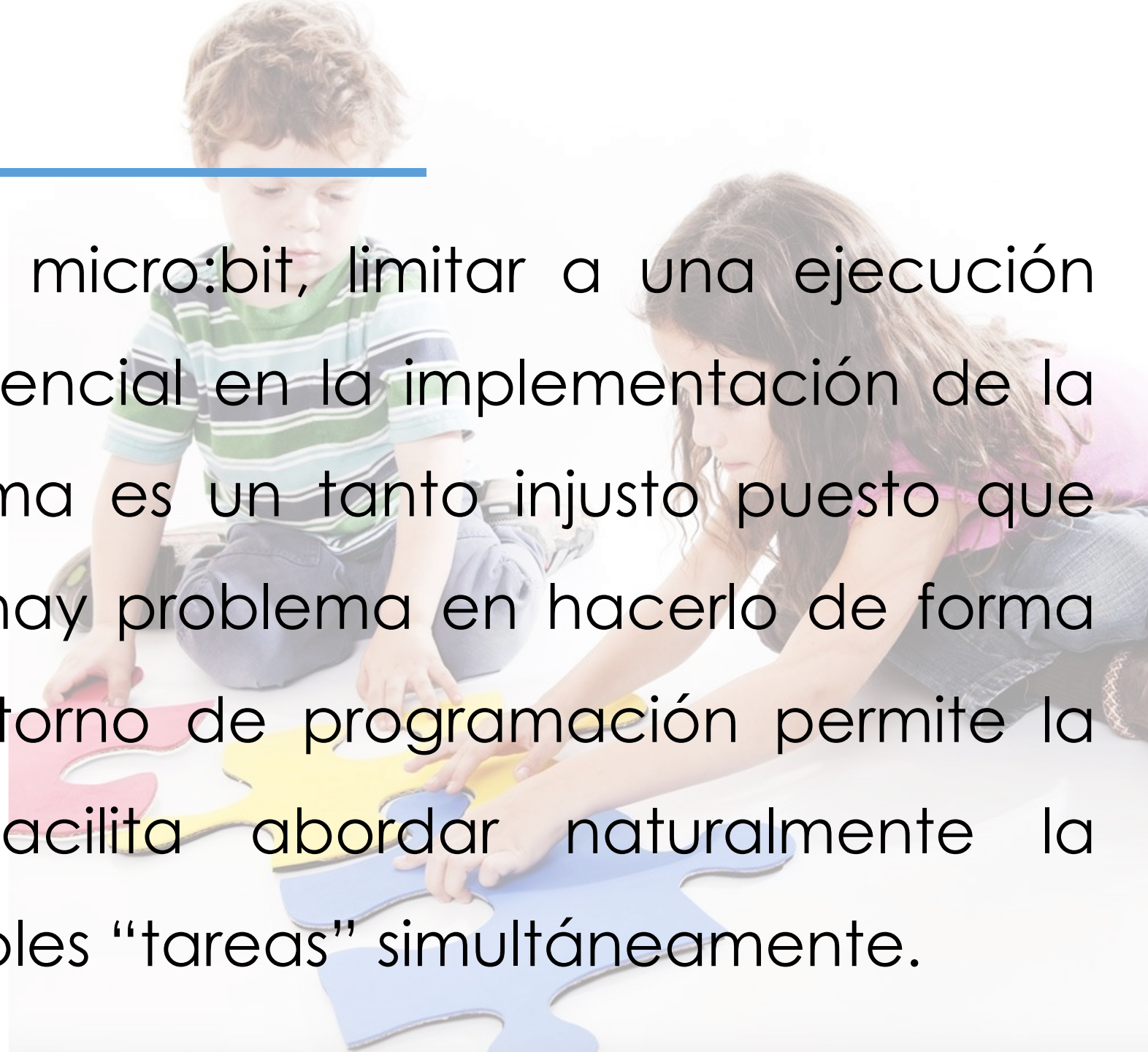


Experiencia



Comúnmente se presenta los algoritmos y los programas como una secuencia de instrucciones. Sin embargo observamos que procesos se desarrollan de forma paralela en nuestro entorno, los procesos en la realidad transcurre de manera paralela en el tiempo; donde algunos son afectados por la marcha de otros, y otros son independientes entre sí.

Experiencia



En el contexto de micro:bit, limitar a una ejecución estrictamente secuencial en la implementación de la solución al problema es un tanto injusto puesto que técnicamente no hay problema en hacerlo de forma concurrente. El entorno de programación permite la concurrencia y facilita abordar naturalmente la ejecución de múltiples "tareas" simultáneamente.

Experiencia

Solución secuencial

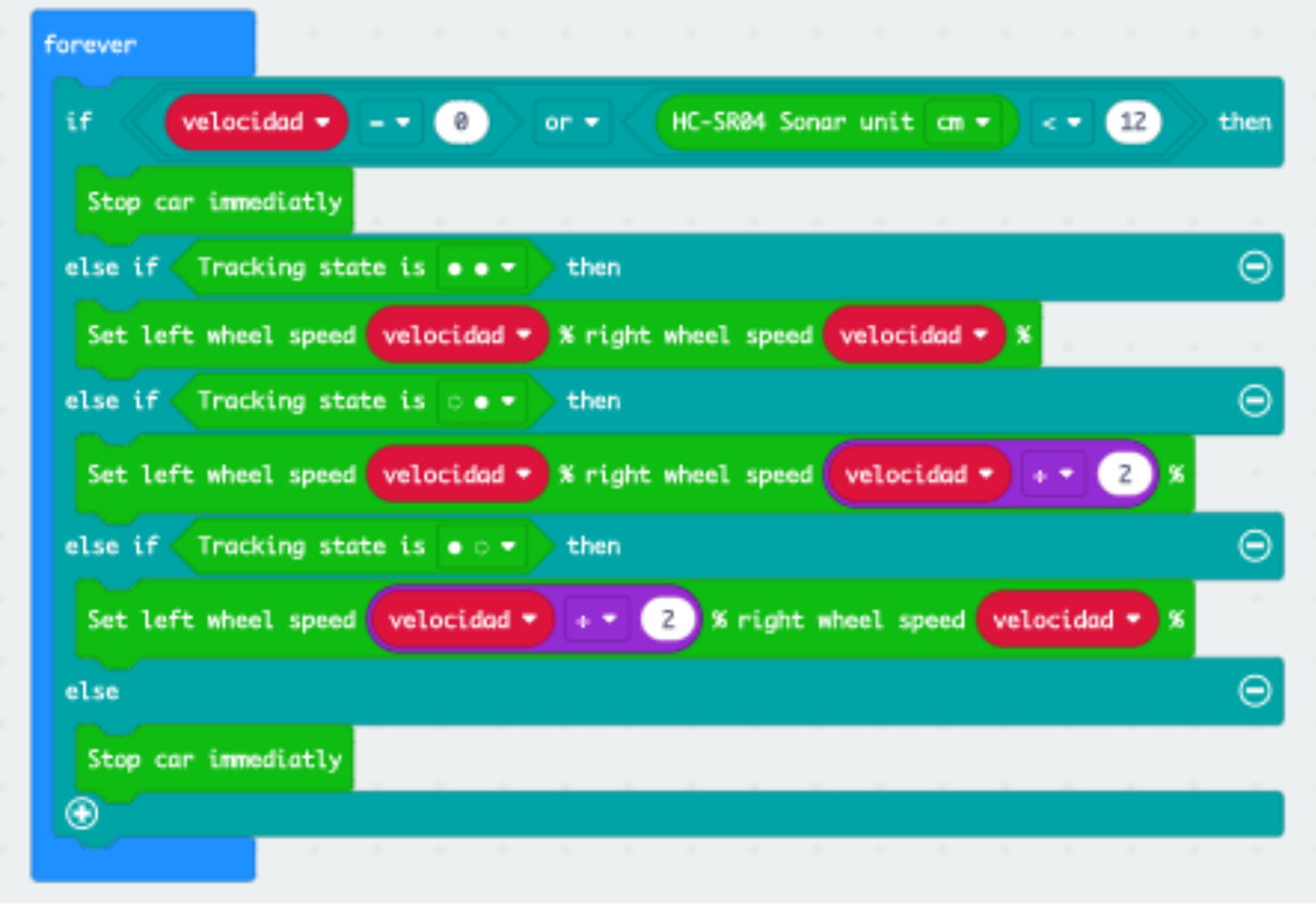
Una tarea lleva a cabo todo el trabajo, verifica que no hay obstáculo a una distancia al frente y sigue la línea.

Solución concurrente

Dos ”tareas”: una sigue la línea y la otra verifica si hay obstáculo y actúa en consecuencia

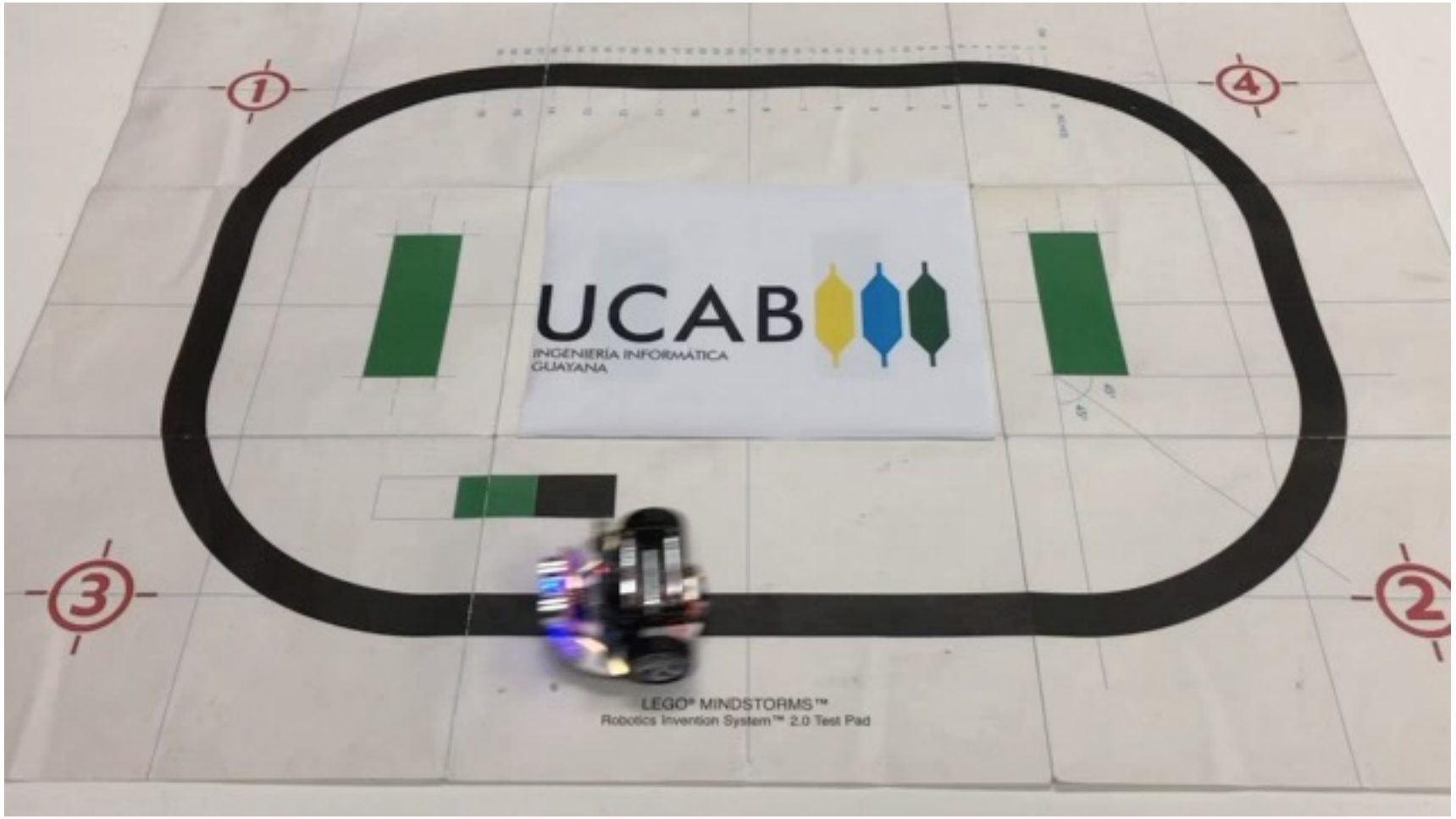
Experiencia

Solución secuencial



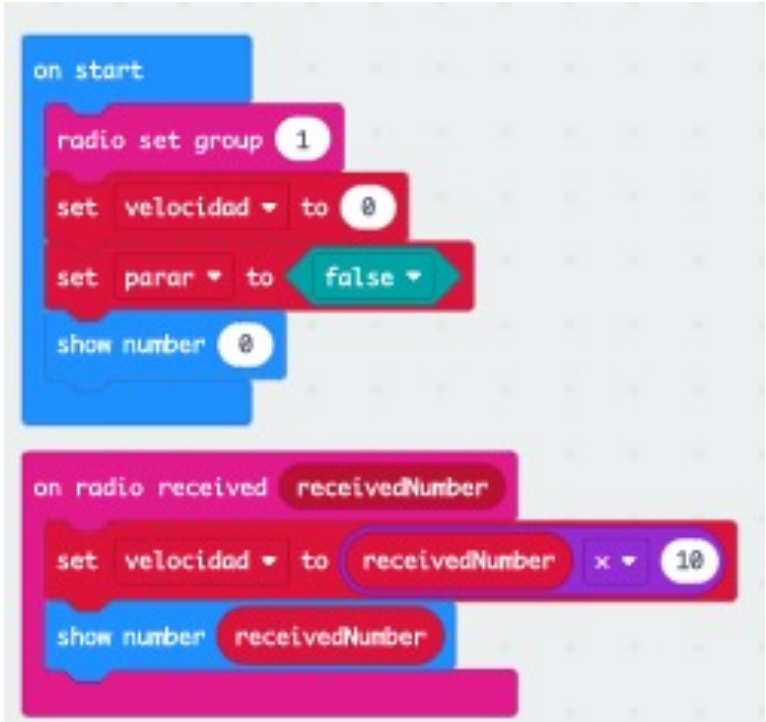
Experiencia

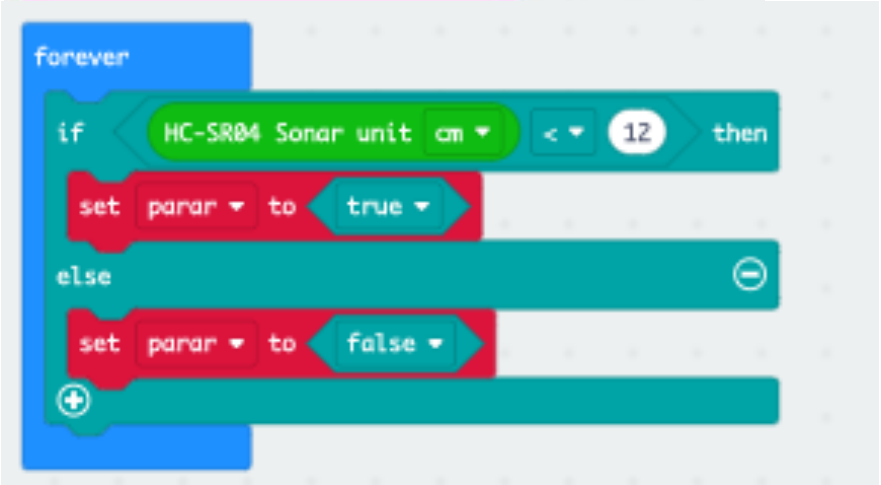
Solución secuencial



Experiencia

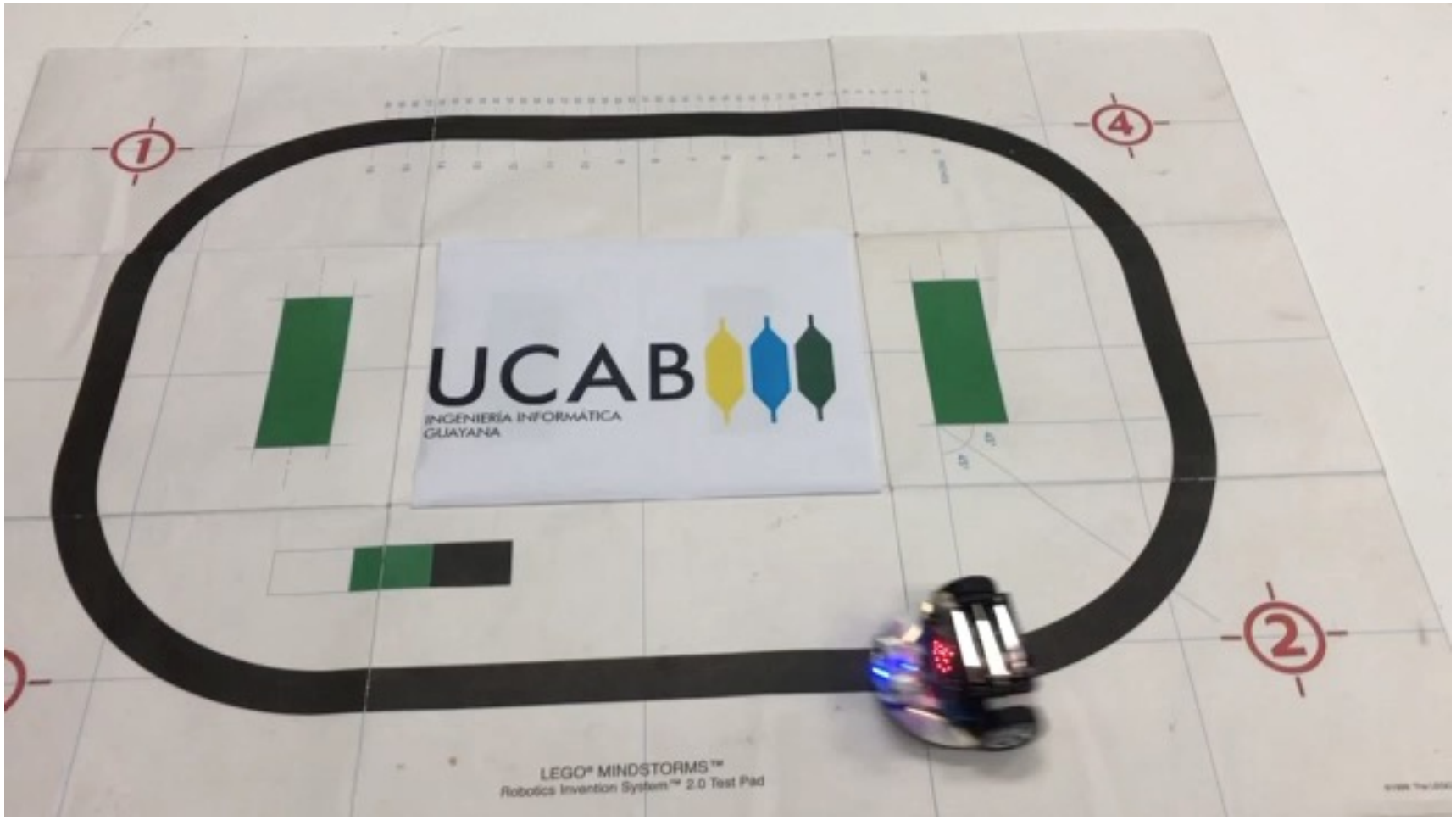
Solución concurrente





Experiencia

Solución concurrente

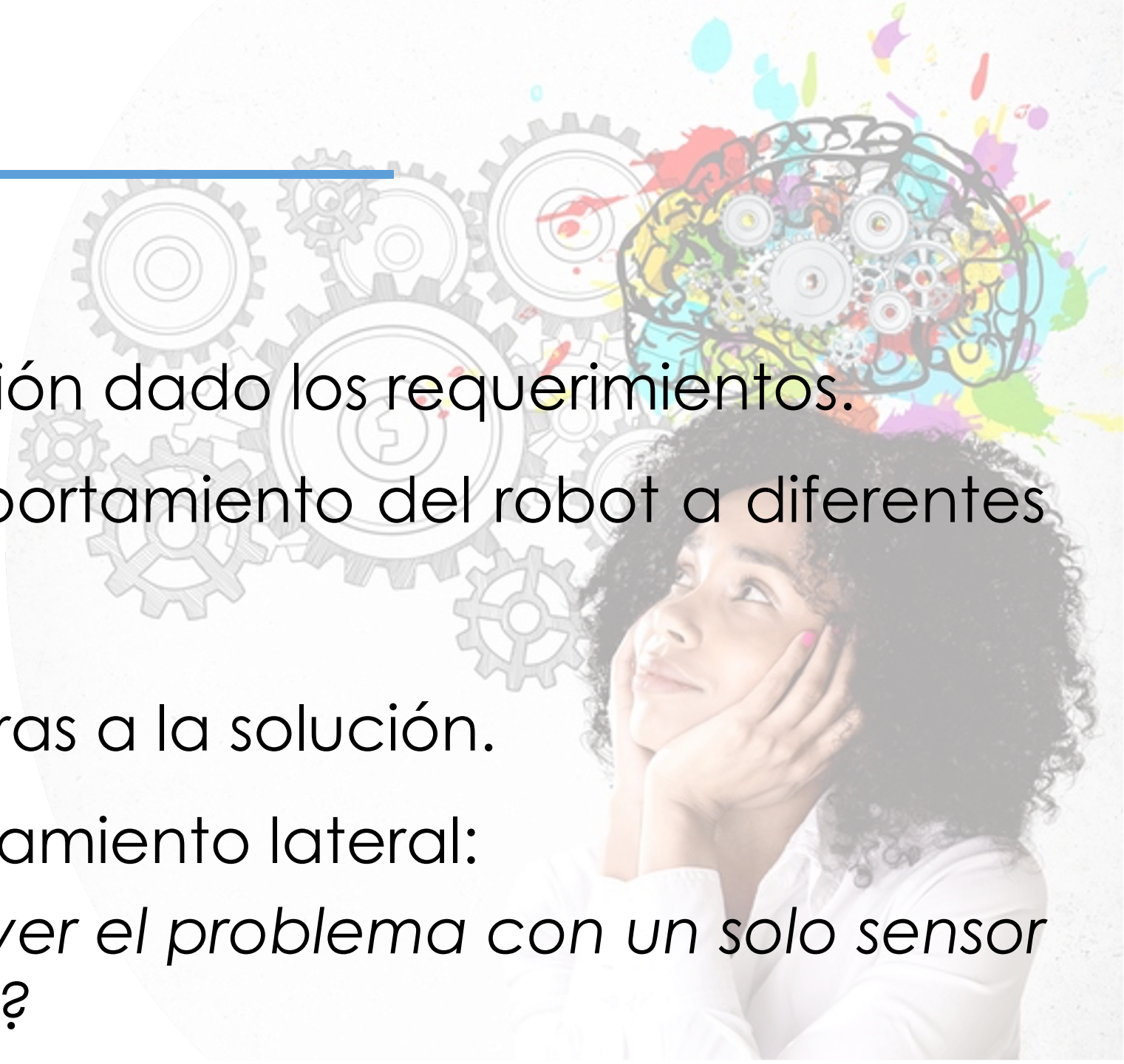


Experiencia

Para finalizar

- Se verifica la solución dado los requerimientos.
- Se evalúa el comportamiento del robot a diferentes velocidades.
- Se piensa en mejoras a la solución.
- Actividad de pensamiento lateral:

  ¿Se puede resolver el problema con un solo sensor óptico reflectivo?



Discución

De la experiencia se tiene

- Aunque hay un soporte de programación concurrente, el uso de fibras, en las guías que viene con los productos esta no es explotada.
- La solución secuencial suele aparecer en aquellos participante que tienen alguna experiencia en programación.

Discución

- La solución concurrente no solo es más natural, sino que presenta un mejor comportamiento, al solapar las operaciones bloqueantes, por ejemplo I/O, sleep(), entre otras.
- Programar múltiples “tareas” concurrente suele implicar algún grado de sincronización, se simplifica en micro:bit dado su ”non-preemptive scheduler”.

Discución

- La programación event-drive en la micro:bit da un mayor nivel de abstracción que se traduce en programas más sencillos, en comparación con la programación secuencial en bajo Arduino.
- En el contexto de micro:bit se oculta la concurrencia en el despacho de eventos.

Discución

- La complejidad de programación en bloque de la micro:bit es mucho más sencilla que en Arduino.
- Al posibilitar observar y manipular el código equivalente asociado a la programación con bloques, permite al participante abordar el aprendizaje de lenguaje como Java y Python.

Conclusiones

- La abstracción ofrecida en la programación en bloques permite abordar la solución sin preocuparse por los detalles de implementación, manteniendo muchas de las capacidades.
- En la programación en bloque, puede existir diferentes niveles de abstracción, llegando incluso a primitivas de robot (vía extensión de Cutebot)

Conclusiones

- Conocer múltiples paradigmas de programación, no solo permite aprovechar mejor los lenguajes y entornos de ejecución, obteniéndose soluciones más sencilla, sino que a futuro facilita el aprendizaje de lenguajes de programación y framework.
- Las soluciones sencillas no solo facilitan la implementación, sino su verificación y mantenimiento.

Trabajo Futuro

- Extender las experiencia a otro tipos de problemas, incluido robótica multiagente, lo que llevaría a utilizar abstracciones asociadas a la programación distribuida.
- Aplicar protocolos que permita generar registros detallados con la finalidad de abordar un análisis en la búsqueda de correlacionar comportamientos.

Muchas Gracias