Aprender pensando y aprender haciendo

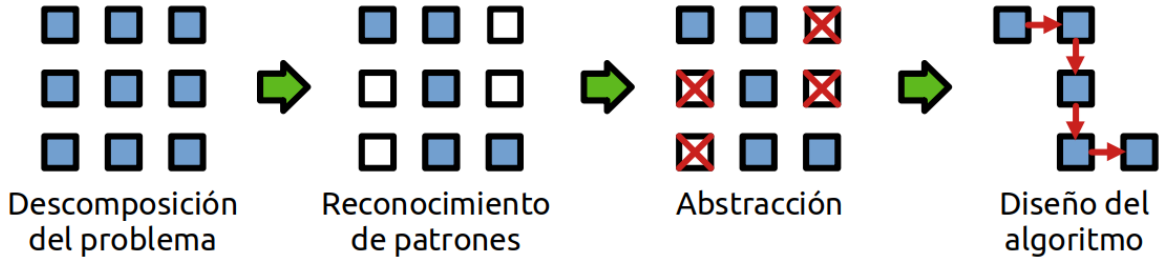
La introducción del **pensamiento computacional** en los currículos es tan reciente que no existe una definición precisa ni del concepto ni de su alcance. Esta cuestión comentada por Serrano (2022) es resuelta por el mismo autor, quien proporciona una definición operativa generada desde una perspectica histórica:

> “ El pensamiento computacional está compuesto por un conjunto de procesos de pensamiento involucrados en la formulación de problemas para que sus soluciones puedan representarse como pasos computacionales y algoritmos.

Según esta definición el pensamiento computacional tiene como propósito **resolver problemas**. Un problema parte de una situación nueva en la que el proceso de resolución es desconocido. Es por ello que la simple memorización de procedimientos para automatizar una tarea, no pone en juego los procesos mentales propios del pensamiento computacional.

Los procesos mentales básicos involucrados en el pensamiento computacional son:

- **Descomposición**. Se analiza y se fragmenta el problema en subproblemas más simples.
- **Identificación de patrones.** Los patrones son situaciones ya experimentadas cuya solución es previamente conocida o bien situaciones que se presentan repetidamente en el problema.
- **Abstracción**. Se pasan por alto los detalles que carecen de importancia.
- **Diseño del algoritmo**. Se secuencian los pasos necesarios para llegar a la solución.

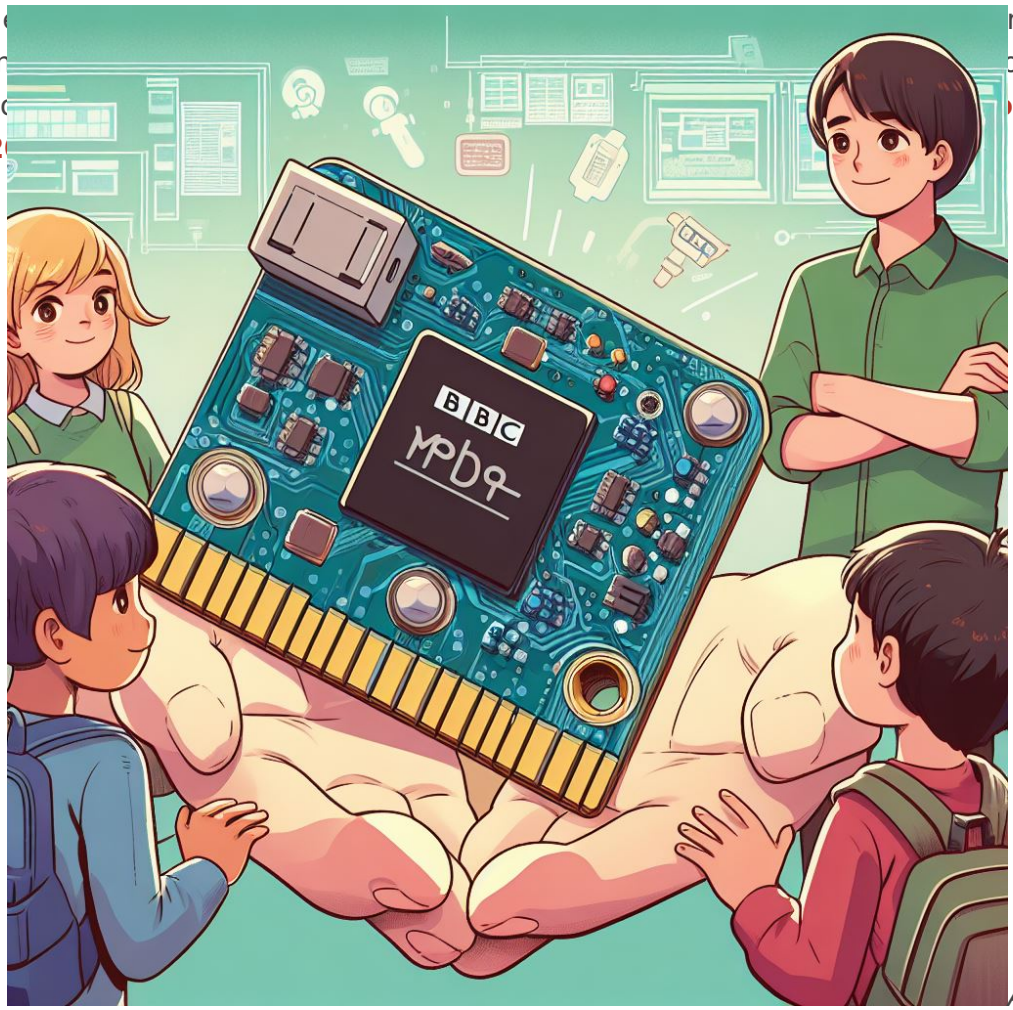


Fases del pensamiento computacional. Elaboración propia

Si la computación permite **aprender pensando**, la robótica educativa contribuye a **aprender haciendo**. Por este motivo, la robótica es denominada **computación física**. La robótica materializa el algoritmo, que puede de esta forma interactuar con el mundo real.

Más complicado resulta determinar **para qué introducir el pensamiento computacional en el aula**. Serrano (2022) cita dos corrientes principales, la que limita su desarrollo al ámbito del desarrollo de la competencia digital del alumnado y la que sostiene que el pensamiento computacional puede ser trabajado en todas las disciplinas con el objetivo de contribuir a la mejora global de los aprendizajes.

La **fundación micro:bit** sigue el primer enfoque, basando su visión en el riesgo de perpetuación y de profundización de las desigualdades sociales que producen las tecnologías digitales. Con su **Teoría del Cambio** (Micro:bit Educational Foundation, 2023c) pretende que todas las personas d r y trabajar en el mundo m on socios tecnológicos, c **nes de estudiantes en 2**



Alegoría de BBC micro:bit generada con IA de Bing.

El segundo enfoque tiene en cuenta el perfecto encaje del pensamiento computacional en las **metodologías activas** que se derivan de la **corriente pedagógica construccionista**, que postula que el sujeto construye su conocimiento a través de la acción orientada a la creación. Dos metodologías muy adecuadas para introducir el pensamiento computacional son el aprendizaje basado en problemas y el aprendizaje basado en juegos.

Por otro lado, las actividades de robótica y de pensamiento computacional contribuyen a desarrollar toda una serie de **habilidades secundarias** o periféricas que no conviene desdeñar. Estas habilidades comunes a todas las **metodologías activas** son, entre otras, la creatividad, el espíritu crítico, la autonomía personal y las capacidades comunicativas.

Dado el gran número de personas y países involucrados en los programas de micro:bit, diversas organizaciones han realizado **estudios de impacto** desde los primeros años. Por ejemplo, un estudio de la BCC (2017) encontró que tras un año con micro:bit, la mayoría del alumnado de 11 años de edad consideraba que programar era fácil y que cualquiera podía hacerlo. Además se produjo un incremento significativo de estudiantes que consideraban que las TIC podían ser una opción profesional para su futuro. Este incremento fue del 70% entre las chicas. En cuanto al profesorado, alrededor de un 80% valoró muy positivamente la introducción de micro:bit en el aula.

Revision #10
Created 2023-06-29 12:02:37 CEST by mario monteagudo alda
Updated 2023-10-05 10:47:17 CEST by mario monteagudo alda

Copyright 2026