

## PENSAMIENTO COMPUTACIONAL

Las sociedades más conscientes han visto que se está tratando con una nueva alfabetización: la alfabetización digital. Y que, por tanto, hay que comenzar desde las primeras etapas del desarrollo individual, al igual como sucede con otras habilidades clave, como la lectura, la escritura y las habilidades matemáticas, e incluso estudiando las concomitancias y coincidencias de esta nueva alfabetización con estas competencias claves tradicionales.

Una respuesta, la más frecuente y la más simple suele ser una respuesta mecánica, que consiste en favorecer el aprendizaje de la programación y de sus lenguajes de forma progresiva, esto consistiría en proponer a los niños tareas de programar desde las primeras etapas. De manera que la progresión estuviese en la dificultad de las tareas y en su carácter motivador, desde las más sencillas y más lúdicas a las más complejas y aburridas.

Se vincula aprendizaje con la respuesta a un estímulo, no con las características de aprendizaje y cognitivas del niño, en la tradición más clásica del conductismo. Pero hay una alternativa, ya tenida en cuenta (Papert, 1980), que enlaza con corrientes clásicas del aprendizaje apoyado en la tecnología: el construccionismo. Esta alternativa está sostenida por algunos autores, inspira a profesores y grupos innovadores en la puesta en marcha de actividades y en algunos pocos casos a corporaciones que, frecuentemente y de forma aislada, se plantean la cuestión de otro modo: **las competencias que se muestran como más eficaces en la codificación son la parte más visible de una forma de pensar que es útil, no sólo en ese ámbito de actividades cognitivas, las que se utilizan en el desarrollo y en la creación de programas y de sistemas informáticos.** En definitiva, sostienen que **hay una forma específica de pensar, de organizar ideas y representaciones, que es propicia y que favorece las competencias computacionales.** *Se trata de una forma de pensar que propicia el análisis y la relación de ideas para la organización y la representación lógica de procedimientos.* Esas habilidades se ven favorecidas con ciertas actividades y con ciertos entornos de aprendizaje desde las primeras etapas. **Se trata del desarrollo de un pensamiento específico, de un pensamiento computacional.**

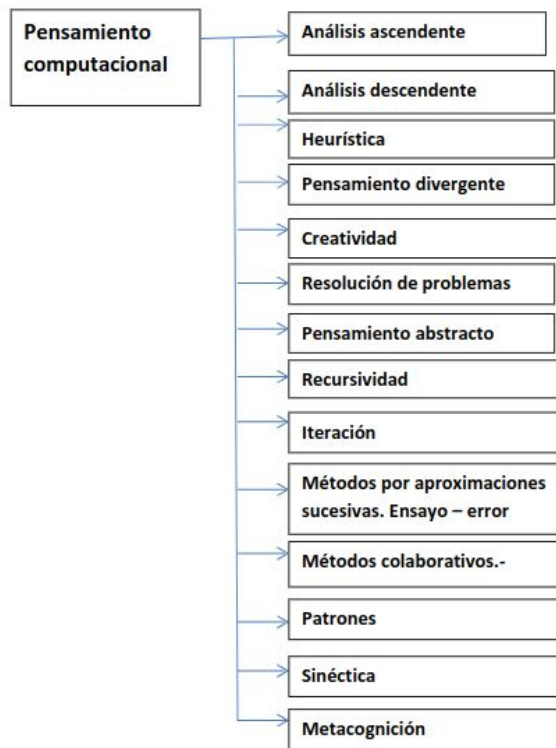
La visión de Papert se podría sintetizar diciendo que *los niños deben programar la computadora en lugar de ser programados por ella* (children should be programming the computer rather than being programmed by it)(Papert, 1980). Ahora, en la fase actual del desarrollo de la tecnología y de las teorías del aprendizaje se podría decir *son los niños los que tienen que educar a los ordenadores no los ordenadores los que tienen que educar a los niños.*

En todo el planteamiento subyace, como idea-fuerza, que, al iguala como sucede con la música, con la danza o con la práctica de deportes, *es clave que se fomente una práctica formativa del pensamiento computacional desde las primeras etapas de desarrollo.*

Wing dice que el pensamiento computacional es una forma de pensar que no es sólo para programadores. Y lo define: **el pensamiento computacional consiste en la resolución de problemas, el diseño de los sistemas, y la comprensión de la conducta humana haciendo uso de los conceptos fundamentales de la informática.** En ese mismo artículo continúa diciendo que *esas son habilidades útiles para todo el mundo, no sólo para los científicos de la computación.* Pero lo interesante es que en ese mismo artículo describe una serie de rasgos que serán muy útiles para establecer un corpus curricular para el aprendizaje basado en el pensamiento computacional.

- En el pensamiento computacional *se conceptualiza, no se programa.* Es preciso pensar como un científico de la computación, se requiere un pensamiento *en múltiples niveles de abstracción;*
- En el pensamiento computacional *son fundamentales las habilidades no memorísticas o no mecánicas.* Para programar los computadores hace falta una mente imaginativa e inteligente. Hace falta la emoción de la creatividad. Esto es *muy parecido al pensamiento divergente,* tal como lo concibieron Polya (1989) y Bono (1986).
- En el pensamiento computacional *se complementa y se combina el pensamiento matemático con la ingeniería.* Ya que, al igual que todas las ciencias, *la computación tiene sus fundamentos formales en las matemáticas. La ingeniería nos proporciona la filosofía base de que construimos sistemas que interactúan con el mundo real.*
- En el pensamiento computacional *lo importante son las ideas, no los artefactos.* Quedan descartados por tanto la fascinación y los espejismos por las novedades tecnológicas. Y mucho menos estos factores como elementos determinantes de la resolución de problemas o de la elección de caminos para resolverlos.

Wing (...) continua con una serie de rasgos pero lo interesante ahora, con ser importante, no es esta perspectiva en sí sino, en el contexto de un análisis y de una elaboración interdisciplinar, ver las implicaciones que tienen estas ideas para una *redefinición de un dominio teórico específico dentro de las teorías del aprendizaje.* Eso por un lado, y por otro encontrar un currículum adecuado a esos dominios conceptuales para las distintas etapas educativas y para la capacitación de maestros y profesores.



Estas componentes no están perfectamente delimitadas ni conceptual ni metodológicamente. No son excluyentes, y según en qué contexto se empleen pueden tener significados distintos. De hecho, ni tan siquiera se puede decir que constituyan elementos de una taxonomía o que correspondan a un mismo nivel operativo o conceptual. Es perfectamente posible que en métodos o procedimientos que se cataloguen por ejemplo como resolución de problemas haya elementos de análisis ascendente, o descendente, y es difícil que un análisis descendente no tenga elementos de recursividad.

### **Fases de creación de un código y componentes de pensamiento computacional**

Las fases del proceso de creación de un código están muy estudiadas desde el punto de vista de la informática. Pero, si queremos estudiarlo desde el punto de vista del pensamiento computacional: qué elementos de este pensamiento, de los que hemos visto en las entradas anteriores, están presentes en cada una de estas fases. Es decir, cuáles son las fases diferenciadas en el proceso de creación de un código:

1. Detección y delimitación del problema y de su naturaleza.
2. Delimitación de métodos y disciplinas en la resolución del problema.
3. Organización de la resolución, feed back e investigación formativa.
4. Diseño de la resolución.
5. Algoritmia/diagrama de flujo.

6. Elaboración del código (programa). Incluye codificación, ejecuciones e implementación, documentación, etiquetas, modularización.
7. Prueba/ Validación .-Incluye implementación y depuraciones de errores,

*La codificación es parte del pensamiento lógico*