

# 1.3 Componentes del Pensamiento Computacional

El pensamiento computacional es un término que sirve para aglutinar una serie de habilidades del pensamiento, imprescindibles para cualquier estudiante del siglo XXI (Kong, Lai y Sun, 2020). El Reino Unido, a través de su currículum en Ciencias de la Computación (DFE.U, 2013), y en el que nos vamos a apoyar principalmente como base para la realización de los ejercicios de este capítulo, detalla que el pensamiento computacional lo forman 6 conceptos (lógica, algoritmos, descomposición, patrones, abstracción, y evaluación sistemática) y 5 aproximaciones (experimentación, creación, depuración, perseverancia y colaboración). Los conceptos explicados serían los siguientes:

1. **Lógica:** el razonamiento lógico nos ayuda a explicar por qué algo sucede. Por esta razón, podemos utilizar el razonamiento lógico para determinar lo que queremos que haga un algoritmo de forma exacta.
2. **Algoritmos:** un algoritmo está escrito para ser entendido por humanos y es una secuencia de instrucciones o conjunto de reglas para solucionar un problema.
3. **Descomposición:** la capacidad que tenemos para poder fraccionar un problema en partes más pequeñas a través de las cuales podamos resolver problemas complejos y gestionar proyectos grandes.
4. **Patrones:** estamos rodeados de patrones. Ser capaces de identificar patrones nos permitirá hacer predicciones, crear reglas y resolver problemas más generales
5. **Abstracción:** la abstracción tiene que ver con simplificar las cosas; identificando qué es importante sin preocuparnos demasiado por lo anecdótico o irrelevante. La abstracción nos permite manejar la complejidad.
6. **Evaluación sistemática:** la evaluación es sistemática y rigurosa; tiene que ver con juzgar la calidad, la efectividad y la eficiencia de las soluciones, sistemas, productos y procesos. La evaluación comprueba que las soluciones aportadas resuelven el problema que nos planteábamos inicialmente no solo de una forma exitosa, sino además la más optimizada. Esto tiene mucho que ver con enseñar a nuestros alumnos que el error no tiene que generar frustración, sino que la programación nos da la maravillosa oportunidad de solucionar un error al estar en nuestras manos, en nuestra comprensión.

Y a continuación las aproximaciones:

1. **Experimentación** ('Tinkering'): significa probar ... haciendo. En los más pequeños es la forma más natural de probar los juegos, de una forma espontánea mediante la

exploración y el descubrimiento. Para los más mayores tiene que ver más con el concepto de ensayo-error-mejora.

2. **Creación** ('Creating'): Programar es un proceso creativo que implica tanto originalidad como la generación de un producto final valioso.
3. **Depuración** ('Debugging'): los errores en un algoritmo, programa o código y el proceso de encontrarlos y arreglarlos se denomina 'debugging'. Algunos 'bugs' son errores lógicos, otros son errores sintácticos.
4. **Perseverancia** ('Persevering'): programar puede ser una tarea ardua y compleja en la que también vamos a necesitar de nuestra predisposición a perseverar en momentos frustrantes.
5. **Colaboración** ('Collaborating'): colaborar significa trabajar con otros para asegurar un mejor resultado. Para la creación de un producto final, en el mundo real no lo crea una sola persona, sino que son varias personas, incluso con diferentes perfiles, las que van creando su propia parte del código para cerrar la solución final. En un caso más sencillo, también se programa por parejas validando la expresión coloquial "cuatro ojos ven más que dos".

Estados Unidos fue otro de los países pioneros en crear un curriculum Computer Science. A través de su Fundación Nacional para la Ciencia de los Estados Unidos ("National Science Foundation") pone en marcha una serie de acciones formativas para la comunidad educativa, como por ejemplo la iniciativa "CS Principles" (Astrachan y Briggs, 2012), cuyo objetivo es fijar y transmitir las bases de las Ciencias de la Computación al alumnado de Bachillerato y primeros años de universidad. Se destacan las siguientes ideas principales relacionadas con procesos cognitivos y operacionales previa a la que fue su primera definición del pensamiento computacional:

1. Formular problemas de un modo que se haga posible utilizar un ordenador y otras máquinas en su resolución
2. Automatizar soluciones a través del pensamiento algorítmico (una serie de pasos discretos y ordenados)
3. Organizar lógicamente y analizar datos
4. Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos
5. Representar datos a través de abstracciones tales como modelos y simulaciones
6. Generalizar y transferir este proceso de solución de problemas a una amplia variedad de situaciones

Otra aproximación distinta es la que realizó el grupo Google for Education (2014), detallando cuales son los 4 fases de procesos cognitivos que trabajan como una rueda circular :

1. Descomposición de un problema o tarea en pasos discretos
2. Reconocimiento de patrones (regularidades)

3. Generalización de dichos patrones y abstracción (descubrir las leyes o principios que causan dichos patrones)
4. Diseño algorítmico (desarrollar instrucciones precisas para resolver el problema y sus análogos)

En el mismo artículo se argumenta que el pensamiento computacional implica una serie de habilidades, incluyendo:

1. Formular problemas de una manera que nos permita usar un ordenador y otras herramientas para ayudar a resolverlos
2. Organizar y analizar de forma lógica un grupo de datos.
3. Representación de datos a través de abstracciones como modelos y simulaciones
4. Soluciones automatizadas a través del pensamiento algorítmico (una serie de pasos ordenados)
5. Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos
6. Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas

Lo realmente novedoso en su propuesta es que declaran que estas habilidades están respaldadas y mejoradas por una serie de disposiciones o actitudes que incluyen:

1. Confianza al desenvolverse bien en la complejidad
2. Persistencia al trabajar con problemas difíciles
3. Ser más tolerante con respecto a la ambigüedad
4. Capacidad de hacer frente a problemas abiertos (sin una solución concreta y evidente)
5. Capacidad de comunicarse y trabajar con otros para llegar a una meta-solución común

Y aún van más allá asegurando que los conceptos de pensamiento computacional no son solo los procesos mentales (por ejemplo, abstracción, diseño de algoritmos, descomposición, reconocimiento de patrones, etc.) sino que también producen resultados tangibles (por ejemplo, automatización, representación de datos, generalización de patrones, etc.) asociados con la resolución de problemas en computación. Estos se definen de la siguiente manera:

1. Abstracción: identificación y extracción de información relevante para definir las ideas principales
2. Diseño de algoritmos: creación de una serie ordenada de instrucciones para resolver problemas similares o para realizar una tarea
3. Automatización: Tener computadoras o máquinas para hacer tareas repetitivas.
4. Análisis de datos: dar sentido a los datos mediante la búsqueda de patrones o el desarrollo de ideas
5. Recopilación de datos: Recopilación de información

6. Representación de datos: representación y organización de datos en gráficos, cuadros, palabras o imágenes apropiados
7. Descomposición: desglosar datos, procesos o problemas en partes más pequeñas y manejables
8. Paralelización: procesamiento simultáneo de tareas más pequeñas de una tarea más grande para alcanzar de manera más eficiente un objetivo común
9. Generalización de patrones: creación de modelos, reglas, principios o teorías de patrones observados para probar los resultados previstos
10. Reconocimiento de patrones: observación de patrones, tendencias y regularidades en los datos
11. Simulación: Desarrollando un modelo para imitar procesos del mundo real

Bers (2017) amplía los elementos relacionados con el pensamiento computacional, concretándolo como un proceso expresivo organizado en siete habilidades de pensamiento computacional: algoritmos, modularidad, estructuras de control, representación, hardware / software, el proceso de diseño, y la depuración.

Para finalizar, partiendo de la propuesta original de Wing (2006), Olabe, Basogain y Basogain (2015) resumen las principales habilidades del pensamiento asociados al pensamiento computacional:

1. Reformular un problema a uno parecido que sepamos resolver por reducción, encuadrarlo, transformar, simular
2. Pensar Recursivamente
3. Procesar en Paralelo
4. Interpretar código como datos y datos como código
5. Generalizar análisis dimensional
6. Reconocer ventajas y desventajas del solapamiento
7. Reconocer coste y potencia de tratamiento indirecto y llamada a proceso
8. Juzgar un programa por simplicidad de diseño
9. Utilizar Abstracción y descomposición en un problema complejo o diseño de sistemas complejos
10. Elegir una correcta representación o modelo para hacer tratable el problema
11. Seguridad en utilizarlo, modificarlo en un problema complejo sin conocer cada detalle
12. Modularizar ante múltiples usuarios
13. Prevención, protección, recuperarse de escenario peor caso
14. Utilizar razonamiento heurístico para encontrar la solución
15. Planificar y aprender en presencia de incertidumbre
16. Buscar, buscar y buscar más
17. Utilizar muchos datos para acelerar la computación
18. Límite tiempo/espacio y memoria/potencia de procesado

Estas tres primeras páginas han servido como base teórica para comprender los procesos de pensamiento computacional con respecto a la resolución de problemas y procesos cognitivos asociados. Dada esta realidad, no es sorprendente que haya aparecido un interés en muchos países por introducir el pensamiento computacional como un conjunto de habilidades de resolución de problemas que los nuevos estudiantes deberían adquirir.

Todos los conceptos y literatura comentadas en este capítulo están extraídos del trabajo de revisión sistemática sobre métodos de evaluación del pensamiento computacional (Ruiz y Bustamente, 2021).

## Bibliografía

Astrachan, O., & Briggs, A. (2012). The CS principles project. *ACM Inroads*, 3(2), 38-42.

Bers, M. U. (2017). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

DFE, U. (2013). National curriculum in England: computing programmes of study. Retrieved July, 16, 2014.

Google for Education. (2014). Exploring Computational Thinking. Recuperado 15 de octubre de 2018, de Google for Education website: <https://edu.google.com/resources/programs/exploring-computational-thinking/>

Kong, S. C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, 151, 103872.

Olabe, X. B., Basogain, M. Á. O., & Basogain, J. C. O. (2015). Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje. *Revista de Educación a Distancia (RED)*, (46)

Ruiz Reinales, C., & Bustamante, J. C. Pensamiento computacional en educación infantil y primaria: una revisión sistemática.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU





---

Revision #3

Created 2022-10-08 07:30:24 CEST by Cristian Ruiz

Updated 2023-01-17 15:49:52 CET by Equipo CATEDU