

## **Pensamiento computacional: Una nueva alfabetización digital**

### **Computational Thinking: A New Digital Literacy**

Miguel Zapata-Ros,  
Universidad de Murcia. España  
mzapata@um.es

#### **Resumen.-**

La sociedad y la economía demandan profesionales cualificados en las industrias tecnológicas. Se da la paradoja de países con un alto índice de paro en las que actualmente se quedan sin cubrir puestos de trabajo de ingenieros y técnicos de industrias y servicios digitales. Esto ha sensibilizado a gestores e instituciones a abordar el problema desde el punto de vista de la formación. Se trata de una nueva alfabetización, la alfabetización digital, y que como tal hay que comenzar desde las primeras etapas del desarrollo individual, al igual como sucede con otras habilidades clave: la lectura, la escritura y las habilidades matemáticas.

El planteamiento, el más frecuente ha consistido en favorecer el aprendizaje de la programación de forma progresiva. Proponiendo a los niños tareas de programar, desde las más sencillas y más lúdicas a las más complejas.

Pero se puede plantear la cuestión de otro modo: Las competencias de codificar son la parte más visible de una forma de pensar que es válida no sólo en ese ámbito de la actividad mental, la que sostiene el desarrollo y la creación de programas y de sistemas. Hay una forma específica de pensar, de organizar ideas y representaciones, que es terreno abonado y que favorece las competencias computacionales. Se trata de una forma de pensar propicia para el análisis y la relación de ideas, para la organización y la representación lógica. 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: el pensamiento computacional.

En este trabajo hacemos una revisión de las formas de pensamiento que se han manifestado y han sido estudiadas como útiles a esta forma de pensar y de resolver problemas en este ámbito cognitivo y a sentar unas bases que en un futuro permitan desarrollar pormenorizadamente los contenidos en un curriculum útil a las distintas modalidades y niveles de educación, así como para la formación de maestros y profesores que los impartan.

#### **Palabras clave.-**

Pensamiento computacional, alfabetización digital, curriculum de Educación Infantil, Primaria y Secundaria, formación del profesorado, competencias digitales

#### **Abstract**

Society and Economy demand skilled professionals in technological industries. There is a current paradox in countries with a high rate of unemployment consisting in unfilled positions for engineers and technicians for the industry and digital services. This has sensitized managers and institutions to address the problem from the point of view of training. There is a new literacy, the digital literacy, and as such we must begin teaching it from the earliest stages of individual development, as well as with other key skills: reading, writing and Math skills.

The most frequent approach to teaching digital literacy has been to gradually encourage the learning of programming. It usually consists in teaching children programming tasks, from the simplest and most entertaining to the most complex.

But one can raise the question in a different way: The powers of encoding are the most visible part of a way of thinking that is valid not only in the field of mental activity, which supports the development and creation of programs and systems. There is a specific way of thinking, to organize ideas and representations, which is fertile ground and favors computational skills. It is a mindset that leads to the analysis and the relationship of ideas to the organization and logic representation. Those skills favor certain activities and certain learning environments from the early stages. It is the development of a specific way of thinking: computational thinking.

In this paper we review forms of thinking which have emerged and have been studied as useful to this way of thinking and problem solving in this cognitive domain. We state they can provide a basis for future development of contents in a curriculum which can be applied to different types and levels of education, as well as to train teachers and professors applying computational thinking.

**Keywords.-**

Computational Thinking, Digital Literacy, Kindergarten Primary and Secondary Curricula, Teacher Training, Digital Skills

### **La vigencia de un corpus curricular sobre “pensamiento computacional”**

En la actualidad las instituciones y agencias competentes, los expertos y los autores de informes de tendencia se han visto sorprendidos por un hecho: la sociedad y los sistemas de producción, de servicios y de consumo demandan profesionales cualificados en las industrias de la información. Particularmente en el mundo desarrollado se da la paradoja de países y regiones con un alto índice de paro en las que actualmente se quedan sin cubrir puestos de trabajo de ingenieros de software, desarrolladores de aplicaciones, documentalistas digitales, por falta de egresados de las escuelas técnicas, por falta de demanda de estos estudios por parte de potenciales alumnos y sobre todo por la falta de personal capacitado. Ante esta situación los sistemas educativos de los países más sensibles han abordado el problema desde la perspectiva de una reorganización del curriculum en la mayor parte de los casos donde se ha producido esa reacción. Sin embargo la cuestión de fondo supone la aparición de unas nuevas destrezas básicas. Las sociedades más conscientes han visto que se trata de una nueva alfabetización, una nueva alfabetización digital, y que por tanto hay que comenzar desde las primeras etapas del desarrollo individual, al igual como sucede con otras habilidades clave: 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 a fuer de ser una respuesta mecánica, ha consistido 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. Nos referimos al 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 de forma aislada, nos planteamos 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.**

Hemos dicho que un precedente remoto de estas ideas está en el construccionismo, en las ideas de autores como Seymour Paper.

Paulo Blikstein (2013) de la Universidad de Stanford, dice que si un historiador tuviera que trazar una línea que uniese la obra de Jean Piaget sobre la psicología del desarrollo a las tendencias actuales en la tecnología educativa, la línea simplemente se llamaría “Papert”. Seymour Papert ha estado en el centro de tres revoluciones: el desarrollo del pensamiento en la infancia, la inteligencia artificial y las tecnologías informáticas para la educación. Quizá el que no haya tenido el impacto debido se deba a que se anticipó.

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 a través de Blikstein, 2013)

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*”

Este trabajo y en general las actividades y reflexiones que pueda propiciar, están justificados por el papel que, en el nuevo contexto, tiene el desarrollo de habilidades que, desde las primeras etapas, faciliten un aprendizaje orientado hacia la programación. Es decir, como vamos a ver, se trata de una nueva alfabetización, de una alfabetización que permita a las personas en su vida real afrontar retos propios de la nueva sociedad y que vaya más allá, permita a los individuos organizar su entorno, sus estrategias de desenvolvimiento, de resolución de problemas cotidianos, además de organizar su mundo de relaciones, en un contexto de comunicación más racional y eficiente. Todo ello con el resultado de poder organizar estrategias para conseguir objetivos personales. En definitiva se trata de conseguir una mayor calidad de vida y un mayor nivel de felicidad.

En todo el planteamiento subyace, como idea-fuerza, **que, al igual a 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.** Y para ello, al igual que se pone en contacto a los niños con un entorno musical o de práctica de danza o deportiva,... se haga con un entorno de objetos y de acciones que promuevan a través de la observación y de la manipulación, aprendizajes adecuados para favorecer este pensamiento.

Aunque pensamos por la experiencia y la práctica que ocurre, no tenemos en muchos casos evidencias de que esos entornos y esas manipulaciones desarrollen las destrezas computacionales o habilidades asociadas a lo que hemos llamado pensamiento computacional. Habría pues que fomentar investigaciones para tenerlas.

Tradicionalmente se ha hablado de aprendizajes o de destrezas concretas: Seriación, discriminación de objetos por propiedades, en las primeras etapas, y en las del pensamiento abstracto o para la resolución de problemas se ha hablado de la modularización, de análisis descendente, de análisis ascendente, de recursividad,... Para lo primero hay multitud de recursos, juegos y actividades que los educadores infantiles conocen bien.

Para ilustrarlo podemos recurrir a un caso concreto que el autor ha tenido oportunidad de conocer de primera mano, que está documentada en un [vídeo](https://www.youtube.com/watch?v=FvOveN8RATw) (<https://www.youtube.com/watch?v=FvOveN8RATw>).

El niño juega con los juguetes de Ikea, con sus padres y con el abuelo. Son juguetes de seriación por colores, formas tamaños y figuras. En su transcurso el bebé de forma rudimentaria organiza y secuencia objetos por colores, tamaños, forma, etc. y supuestamente percibe el plano y el espacio y sus propiedades. En el juego manifiesta gusto por hacerlo, cuando con exclamaciones y aplausos se le comunica que lo hace bien.

La grabación procede de una ocasión que sorprendió a su padre, por eso la conserva. En ella se guía con total precisión por la secuencia de colores alternativos para usar la cuchara. La experiencia fue espontánea, no es un caso de laboratorio sino familiar. El niño no está “adiestrado”: En la grabación se conserva la voz de la madre para que se observe su sorpresa.

¿Qué es lo que había pasado en su cabeza para que conservase tan estrictamente la secuencia de colores alternativos al usar las cucharas? ¿Hay alguna relación entre los juegos que había utilizado y el gusto o la preferencia por una secuencia basada en el color? No tenemos ninguna evidencia que nos lleve a esa conclusión. Pero el que utilizase el color para organizar la secuencia, de la misma forma que lo había hecho en los juegos, nos induce a pensar que alguna relación había. Y que lo que posiblemente hubiera sucedido es que había transferido la idea (o la pre-idea, la asociación) desde el contexto inicial, los juegos, a las cucharas. Es decir **había manejado la secuencia asociada al color con independencia de la situación.**

Hay por tanto multitud de áreas del aprendizaje que conviene explorar e investigar en esta nueva frontera. Y en la planificación de los currícula tendrá que plantearse esta dicotomía: Enseñar a programar con dificultad progresiva (si se quiere incluso de forma lúdica o con juegos) o favorecer este nuevo tipo de pensamiento. Obviamente no hace falta decir que nuestra propuesta es la segunda, que además incluye a la primera.

Pero volvamos al artículo, su naturaleza y su tema.

Tropezamos con varios problemas de comienzo: delimitar el contenido y encontrar los términos y conceptos adecuados para definirlo.

Pensamos inicialmente en utilizar la expresión codificación y precodificación. La segunda extraída de la literatura anglosajona, *coding* o *code*. En este sentido es utilizada en los textos que publicitan este año como el año del código, o de la codificación, o de la programación (*Year of code*). Así se dice:

A través de la codificación (*code*) la gente puede descubrir el poder de la informática, cambiando su forma de pensar acerca de su entorno y obtener el máximo provecho del mundo que le rodea.

O bien tal como se dice en el informe *Computing our future Computer programming and coding – Priorities, school curricula and initiatives across Europe*:

La codificación (*coding*) es cada vez más una competencia clave que tendrá que ser adquirida por todos los jóvenes estudiantes y cada vez más por los trabajadores en una amplia gama de actividades industriales y profesiones. La codificación es parte del razonamiento lógico y representa una de las habilidades clave que forma parte de lo que ahora se llaman “habilidades del siglo 21”.

Como vemos es un dominio conceptual muy próximo a lo que hemos visto y veremos que es el pensamiento computacional, al menos se expresa con ese sentido que le hemos atribuido.

Por otro lado de igual forma que se habla de prelectura, pre-escritura o precálculo para nombrar competencias que allanan el camino a las destrezas clave y a las competencias instrumentales que anuncian, cabe hablar de precodificación para designar las competencias que son previas y necesarias en las fases anteriores del desarrollo para la codificación. Como hemos dicho nos referimos por ejemplo a construcciones mentales que permiten alojar características de objetos de igual forma a como lo hacen las variables con los valores: Son en este caso el color, la forma, el tamaño,... O también operaciones con estos rasgos como son la seriación. Evidentemente hay muchas más habilidades y más complejas en su análisis y en el diseño de actividades y entornos para que este aprendizaje se produzca. Este ámbito de la instrucción es lo que podría denominarse precodificación.

Podría llamarse igualmente preprogramación, pero creemos que es más propio llamarle precodificación, pues codificación describe con más precisión y más ajuste conceptual, la transferencia de acciones e informaciones para que puedan ser interpretados por los ordenadores y otros dispositivos de proceso, transporte y almacenamiento de la información.

Podría ser así, y sin duda sería correcto. Sin embargo vimos que esto no correspondía con el propósito más amplio que es no solo prepararse para la programación y para la codificación sino para dotarse de claves de comprensión y de representación de los objetos de conocimiento en general. Pensamos pues como más adecuada en la expresión “pensamiento computacional” (*computational thinking*), que después desarrollaremos

Por otro lado hubiéramos podido elegir igualmente el término alfabetización digital, reconociendo que impropia, en español, tiene resonancias próximas al término “alfabetización informática”, al menos en su uso. Y éste hace hincapié en la informática de usuario al considerarse esta alfabetización como el conocimiento y la destreza para manejarse en entornos de usuario. Así es frecuente entre la gente poco ilustrada confundir al buen informático con el que maneja bien, es hábil, con los programas de usuario, las APPs, o al que se maneja con fluidez y rapidez en los ambientes de menús, ventanas y opciones, o simplemente al que tiene habilidad en los pulgares para manejar un **smartphone**, o con el índice para moverse por un tablet.

No obstante podemos aceptar la definición de alfabetización digital (*computer literacy*) como el conocimiento y la capacidad de utilizar las computadoras y la tecnología relacionada con ellas de manera eficiente, con una serie de habilidades que cubren los

niveles de uso elemental de la programación y la resolución de problemas avanzada. (Washington, US Congress of Technology Assessment, OTA CIT-235 April 1984, page 234). Pero como hemos dicho ese mismo documento acepta que la alfabetización digital también se puede utilizar para describir el nivel de acomodo que un individuo tiene con el uso de programas de ordenador y otras aplicaciones que están asociados con las computadoras. La alfabetización digital por último se puede referir a la comprensión de cómo funcionan los ordenadores y a la facilidad de operar con ellos.

En lo que sigue hablaremos más de “pensamiento computacional” (*computational thinking*) y de iniciativas necesarias como son desarrollar un curriculum y propuestas de formación para maestros y profesores.

### **¿Cuál es el problema del pensamiento computacional?**

---

Al principio del artículo decíamos

La respuesta más frecuente y más simple, a fuer de ser una respuesta mecánica, ha consistido 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.

Este es el tipo de planteamiento que está detrás de inducir a los niños a hacer muchas líneas de programa y a hacerlas muy rápidamente, sin pensar previamente mucho en el problema a resolver, sin diagramas, sin diseño,... Es la idea que hay detrás de los concursos de programación, de enseñar a programar a través de juegos, etc.

Es un planteamiento competitivo que deja a fuera a muchos niños y que posiblemente haga poco deseable para muchos ser programador, o al menos les de una imagen de *frikis*. O unos tipos raros con un perfil difícil de tener. En definitiva puede llegar a ser un planteamiento por muchas razones excluyente.

Pero volvamos al principio, antes de empezar a escribir código de forma compulsiva, lo importante es saber cómo se representan la realidad, el mundo de objetivos y expectativas, lo que piensan en definitiva los que tienen éxito en hacer programas potentes. Lo importante no es el software que escriben sino lo que piensan. Y sobre todo la forma en que piensan.

Conocer este mundo de ideas y de representaciones, como operan constituye el principio básico del “pensamiento computacional”, y cualquier otro conocimiento como memorizar a la perfección las reglas de toda la sintaxis y las primitivas de cualquier lenguaje de programación no le sirve de nada a los alumnos si no pueden pensar en buenas maneras de aplicarlas.

Desgraciadamente la forma más frecuente de enseñar a programar y la que se está empezando a utilizar en nuestro país, como por ejemplo se ha hecho en la Comunidad

de Madrid<sup>1</sup> (Valverde et al, 2015), está en la clave señalada: Conducir a los alumnos, en este caso, de secundaria por el camino más áspero, el de la programación *per se*. Si no se proporcionan otro tipo de ayudas o de claves se excluyen a los que no tienen el don de para programar directamente, ante solo la comprensión de procedimientos puramente de programación, creando con ello el estereotipo de que la programación es solo cosa de los programadores.

Hay un efecto derivado. Si se aprende a programar asociado a un lenguaje es posible que no se produzca la transferencia y en futuras ocasiones no se pueda repetir el proceso. Esto hace que la inserción profesional no se produzca con toda eficacia que podría ser si se hiciera vinculado a operaciones cognitivas superiores. Las asociaciones profesionales se quejan de que las empresas contraten a informáticos de forma efímera. Sin reparar que es posible que suceda porque han aprendido de forma vinculada a lenguajes y programas efímeros, y que cuando cambie el programa o la versión no tendrán flexibilidad mental para adaptarse a nuevos entornos, no solo de programación, sino de problemas. Esto no sucede, y las empresas lo saben, si contratan a titulados más familiarizados con elementos de pensamiento heurísticos o de otro tipo entre los glosados en este trabajo, como son matemáticos o físicos que sí tienen esa competencia de resolver problemas en entornos cambiantes.

Sin embargo sí existen referencias (Raja, 2014) de investigaciones que ponen de manifiesto que cuando se empieza por enseñar el pensamiento computacional en vez de por la elaboración de códigos, desvinculando la iniciación en el aprendizaje a ser diestros con el ordenador, tal como se entiende habitualmente, se evita el principio de discriminación que hace que algún tipo de niños y de niñas se inhiban. Supone pues un principio de democratización del aprendizaje. Esto supone además que los que en un futuro pueden ser bibliotecarios, médicos o artistas pueden ser también buenos programadores. Y por ende podría ampliar la base de conocimiento que se vuelca al mundo de la computación, lo que constituye el motor y el combustible de la Sociedad del Conocimiento.

Veamos que algunas habilidades propias del pensamiento computacional no tienen por qué estar vinculadas a los ordenadores. Hay en un ejemplo de interés en este sentido (en una época en que no había ordenadores), se trata de un proceso de análisis de datos hecho por un lego, un médico especializado en epidemias, para resolver un problema crucial. Hoy probablemente esto no podría haber sucedido así. Los programadores no tienen conocimientos de epidemiología, y los epidemiólogos no tienen por lo general pensamiento computacional (Koch and Denike, 2009, a través de Raja, 2014).

En 1854, un médico de Londres llamado John Snow ayudó a sofocar un brote de cólera que había matado a 616 vecinos. Dejando de lado la teoría predominante de la época, la Teoría miasmática de la enfermedad, que aseguraba que las enfermedades se producían por emanaciones surgidas del terreno, indagó las costumbres y los rasgos de la forma de vivir de los enfermos. En un mapa levanto columnas con la ubicación de los muertos, la frecuencia (¿no nos recuerda un *infograma*?):

---

<sup>1</sup> DECRETO 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria. Recuperado a partir de [http://www.bocm.es/boletin/CM\\_Orden\\_BOCM/2015/05/20/BOCM-20150520-1.PDF](http://www.bocm.es/boletin/CM_Orden_BOCM/2015/05/20/BOCM-20150520-1.PDF)



Fig. 1.- Mapa obtenido de “On the Mode of Communication of Cholera,” 1854. En Rev. Henry Whitehead mapped almost 700 cholera deaths, sewer lines, and both the incorrect location of the old plague burial site (oval) and its correct size and location a block from the Broad Street pump in this 1855 map (Koch and Denike, 2009) y en JOHN SNOW’S MAP 1 (Broad Street Pump Outbreak, 1854).

Observó que las columnas crecían alrededor de una bomba de agua en Broad Street en el Soho (ver el mapa) que además estaba cerca de un pozo negro con fugas. La hipótesis que formuló es que la causa de la enfermedad estaba en el agua. Implícitamente estaba aplicando principios que ahora son clásicos de pensamiento computacional entraron: Cruzar dos conjuntos de datos para obtener un conocimiento nuevo (cruzar la ubicación de las muertes con las ubicaciones de las bombas de agua), contrastar el resultado por iteraciones sucesivas y el reconocimiento de patrones. Cuando se clausuró la bomba el brote cesó.

### **Definición de pensamiento computacional.**

La idea más extendida sobre lo que es la alfabetización digital (*Digital Literacy*) es que consiste en una trasposición.

A lo largo de la historia se han sucedido distintas alfabetizaciones y todas han tenido una significación común: han supuesto una adaptación a los nuevos medios de comunicación, representación y proceso de la información entre humanos. Así, según esta idea, la Alfabetización Digital es la adaptación y la capacitación para esas funciones de comunicación, representación y proceso a las coordenadas de la revolución tecnológica y de la sociedad de la información, consideradas en sentido estrictamente tecnológico, como revolución de medios de comunicación y de difusión de ideas.

Ésta es la idea del autor que pasó como el creador del concepto: Paul Gilster. De esta forma el concepto de alfabetización digital, tal como se utiliza ahora de forma general, fue presentado en el libro del mismo nombre (Gilster, 1997). Gilster no proporcionó una listas de habilidades, competencias o actitudes en la definición de lo que es una cultura digital (a diferencia de como nosotros tratamos de hacer). Más bien lo explicó, de manera muy general: como la capacidad de entender y utilizar la información de una gran variedad de fuentes digitales. Por tanto, se trata de la actualización per se de la idea



tradicional de la alfabetización. Se trata de la capacidad de leer, escribir y realizar cualquier transacción con la información, pero ahora utilizando las tecnologías y los formatos de datos actuales, al igual que la alfabetización clásica utilizaba la tecnología de la información y los formatos de cada época (libros, papiros, pergaminos, tablillas,...). Pero sobre todo, en ambos casos, se considera como **un conjunto de habilidades esenciales para la vida**. La crítica es que ésta es una expresión genérica del concepto, sin ir ilustrada o acompañada "listas de competencias".

Gilster no fue el primero a utilizar la expresión "alfabetización digital"; ya se había aplicado a lo largo de la década de 1990 por una serie de autores, que la utilizaron para significar esencialmente la capacidad de leer y comprender elementos de información en los formatos de hipertexto o multimedia (Bawden, 2001).

Un planteamiento típico en este sentido fue el de Lanham (1995), que consideraba la alfabetización digital como una especie de "alfabetización multimedia". Su argumento era que desde una fuente digital se podrían generar muchas formas de texto, de informaciones, imágenes, sonidos, etc. Esto justificaba la necesidad de una nueva forma de alfabetización, con el fin de interpretar, de dar sentido a estas nuevas formas de presentación. La crítica es que con el tiempo este aspecto dejó de ser importante, y era muy restrictivo, frente al concepto más amplio de la alfabetización digital, y demasiado focalizado en la tecnología de una época. Frente a la conceptualización mucho más amplia de Gilster. Distintas concepciones de este tipo son revisadas por Eshet (2002), que llega a la conclusión, de que la alfabetización digital debe considerarse más como la capacidad de utilizar las fuentes digitales de forma eficaz. Se trata pues de un tipo especial de mentalidad o pensamiento. Esta conceptualización está bastante más próxima a lo que planteamos en este trabajo, solo que se refiere a la forma de procesar la información no a organizar la resolución de problemas. El pensamiento computacional es más una resolución de problemas.

De todas formas Gilster en su libro de 1997 ya rompe con la idea que dio lugar al mito de los "nativos digitales". Afirma explícitamente que "la alfabetización digital tiene que ver con el dominio de las ideas, no con las pulsaciones en el teclado" -así distingue, en su concepción lo limitado de las "habilidades técnicas" desde la perspectiva de la alfabetización digital. Señala que "no sólo hay que adquirir la habilidad de hallar las cosas, sobre todo se debe adquirir la capacidad de utilizar esas cosas en la vida del individuo"(pp. 1-2).

Sobre estas ideas David Bawden (2008 Capítulo 1), a partir de lo dicho por Pablo Gilster, afirma que la alfabetización digital implica una forma de distinguir una variedad creciente de conceptos y de hechos, para delimitar los que son relevantes en orden a conseguir el dominio de las ideas. E insiste en lo necesario para ello de una evaluación cuidadosa de la información, en el análisis inteligente y en la síntesis. Para ello proporciona listas de habilidades específicas y de técnicas que se consideran necesarias para estos objetivos, y que en conjunto constituyen lo que califica como una cultura digital.

Sobre estas competencias Bawden (2008) remite a las expuestas en otro trabajo anterior (Bawden, 2001):

- Adquirir un "conjunto de conocimiento", y con ellos construir un "bagaje de información fiable" de diversas fuentes
- Habilidades de recuperación, utilizando además un "pensamiento crítico" para hacer juicios informados sobre la información recuperada, y para asegurar la validez e integridad de las fuentes de Internet
- Leer y comprender de forma dinámica y cambiante material no secuencial
- Ser consciente del valor de las herramientas tradicionales en contextos y en relación con los medios de comunicación en red
- Ser consciente del valor de las "redes populares" como fuentes de asesoramiento y ayuda
- Utilizar filtros y otras herramientas, lógicas y cognitivas, para gestionar la información disponible, valorando su relevancia
- Sentirse cómodo y familiarizado con la publicación y comunicación de la información en los nuevos medios, así como con el acceso a ella.

Llegados a este punto hay una segunda línea de delimitación conceptual, estudiada por Eshet-Alkalai (2004). A partir de su reflexión advierte sobre la incompatibilidad entre los planteamientos de aquellos que conciben la alfabetización digital como "principalmente constituida por habilidades técnicas, y los que la ven centrada en aspectos cognitivos y socio-emocionales del trabajo en entornos digitales".

Otro criterio que se ha tenido en cuenta en la aproximación al concepto de Alfabetización Digital fue el de clasificar (Lankshear y Knobel, 2006) según se tratase de un enfoque conceptual o de un enfoque "operacional", es decir a partir de operaciones estandarizadas.

Esta última tendencia, la de definiciones basadas en operaciones estandarizadas, confieren un carácter "funcional" a la Alfabetización Digital, lo que le dota de una índole de cultura digital, centrando el estudio en la naturaleza de las tareas, presentaciones, demostraciones de habilidades, etc. que se realizan, y avanzando en la construcción de estándares para definir qué es o no Alfabetización Digital.

Por último hay una variante comercial de la Alfabetización Digital, que consiste en una certificación de competencias. Es la acreditación conocida como Internet and Computing Core Certificación (IC<sup>3</sup>) ([www.certiport.com](http://www.certiport.com)). Su página web afirma que la "certificación IC<sup>3</sup> ayuda a aprender y demostrar a Internet y la alfabetización digital a través de un estándar de evaluación del aprendizaje válido para la industria en todo el mundo". Se basa en un sistema de formación y de certificación a través de un examen que abarca contenidos sobre Fundamentos de Informática, en aplicaciones básicas y clave para la vida, y en lo que llaman la "vida conectada".

Lo que se propone en este trabajo, con la construcción de idea del pensamiento computacional a partir de elementos o de formas específicas de pensamiento para resolver problemas, tiene que ver con la Alfabetización Digital en que éste está constituido por competencias clave que sirven para aprender y comprender ideas, procesos y fenómenos no sólo en el ámbito de la programación de ordenadores o incluso del mundo de la computación, de Internet o de la nueva sociedad del conocimiento, sino que es sobre todo útil para emprender operaciones cognitivas y elaboración complejas que de otra forma sería más complejo, o imposible, realizar. O bien, sin estos elementos de conocimiento, sería más difícil resolver ciertos problemas de cualquier ámbito no solo de la vida científica o tecnológica sino de la vida común. Como dijimos se considera como **un conjunto de habilidades esenciales para la vida en la mayoría de los casos y como un talante especial para afrontar problemas científicos y tecnológicos**. Señalamos en relación con esta última acepción algunos ejemplos que veremos después, o hemos visto, en este trabajo, como son el de la determinación de los génesis de los contagios, en el caso que cuestiona teorías generales como la teoría miasmática del origen de las enfermedades, o la secuencia de llenado de los asientos en el embarque de aeronaves.

Tras lo visto sobre Alfabetización Digital y conceptos asociados podemos abordar directamente la idea de pensamiento computacional.

En ese sentido es muy interesante lo que dice, en un primer acercamiento al concepto, la informática Tasneem Raja (2014) en el post *We Can Code It!*, de la revista-blog *Mother Jones*:

“El enfoque computacional se basa en ver el mundo como una serie de puzzles, a los que se puede romper en trozos más pequeños y resolver poco a poco a través de la lógica y el razonamiento deductivo”.

Esta es una forma intuitiva en la que una autora, que proviene del mundo computacional, aborda una serie de métodos ampliamente conocidos en el mundo de la psicología del aprendizaje. Implícitamente está hablando de análisis descendente y de elaboración: *Puzzles* —problemas— que se pueden dividir en *puzzles*—problemas— más pequeños, para ir resolviéndolos. También, en el mismo párrafo, vemos una alusión implícita a la recursividad, falta la cláusula de parada y la vuelta atrás, pero evidentemente después de armar los puzzles pequeños cada uno hay que ensamblarlo en el *puzzle* general. Y también habrá que decir en qué nivel habrá que parar y dar marcha atrás.

Pero hay otros procedimientos para abordar tareas complejas que igualmente se pueden considerar como propias de este pensamiento, como son el análisis ascendente, y todo lo que constituye la heurística, el pensamiento divergente o lateral, la creatividad, la resolución de problemas, el pensamiento abstracto, la recursividad, la iteración, los métodos por aproximaciones sucesivas, el ensayo-error, los métodos colaborativos, el entender cosas juntos, etc. que veremos en lo que sigue.

La definición de pensamiento computacional que se considera la más apropiada es la que dió Jeannette Wing (March 2006), vicepresidente corporativo de Microsoft Research y profesora de Computer Science Department Carnegie Mellon University, que fue quien popularizó el término en su artículo “*Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use*”, cuyo título es en sí mismo una definición.

Wing (March 2006) 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 nos van a ser muy útiles para establecer un corpus curricular para el aprendizaje basado en el pensamiento computacional. Así por ejemplo se dice:

- *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.*- Memoria significa mecánico, aburrido, rutinario. 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 (March 2006) 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.

### **Un dominio teórico específico del pensamiento computacional en las teorías del aprendizaje y un currículum.**

---

En este artículo, hasta aquí, hemos justificado la necesidad de contar con un corpus curricular, en el sentido que da Eggleston (1980) a la expresión, sobre pensamiento computacional y de una relación de habilidades asociadas.

Pero sobre todo es importante plantearlo 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. Y desde luego definir descriptivamente en un primer acercamiento un currículum

adecuado a esos dominios conceptuales para las distintas etapas educativas y para la capacitación de maestros y profesores.

Esto es lo que en una primera aproximación tratamos de hacer con las limitaciones de un tratamiento general, pero que sin duda será tratado y ampliado en posteriores documentos con más extensión y documentación. En un primer repaso pues y en la búsqueda correspondiente hemos intentamos conectar con todo lo dicho las conceptualizaciones y modalidades del pensamiento según las teorías del aprendizaje. Así pues hemos encontrado las siguientes componentes del pensamiento computacional.

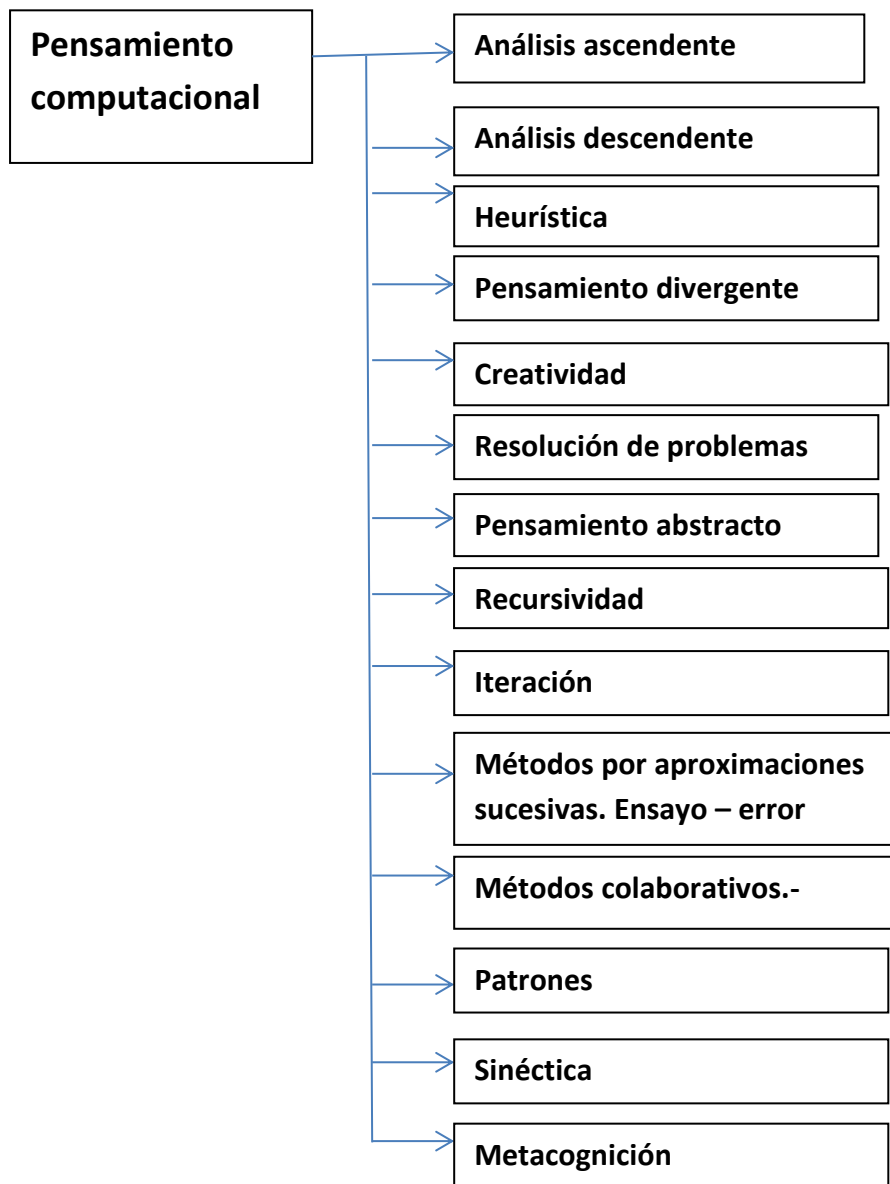


Fig. 2

Conviene decir que 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.

#### **Análisis descendente.-**

---

La obtención de un método general de resolución, de un algoritmo, implica un proceso de análisis descendente que puede llevar al diseño de submétodos de resolución, o bien de módulos de resolución de problemas distintos y auxiliares, o bien a definir acciones concretas, modelos o funciones matemáticas auxiliares, etc.

#### **Análisis ascendente.-**

---

A la hora de plantear un problema complejo una de las formas de abordarlo es resolver primero los problemas más concretos, para pasar después a resolver los más abstractos. Es decir, vamos de lo más concreto a lo más abstracto. Esta forma de plantearlo recibe el nombre de análisis ascendente.

Hay que ser muy cuidadoso a la hora de elegir este método, está repleto de dificultades, la mayor parte de las cuales consisten en el tiempo que lleva y en que necesita de otro tipo de pensamientos, por ejemplo el pensamiento divergente para orientarse en el camino para llegar al resultado general, abstracto que se desea. Muchas veces hay que situarse primero en un nivel abstracto y de allí seleccionar los problemas concretos que nos ilustran para la resolución del problema general.

#### **Heurística<sup>2</sup>.-**

---

Habitualmente se define Heurística como un saber no científico, pero que se aplica en entornos científicos y que se refiere técnicas basadas en la experiencia para la resolución de problemas, al aprendizaje y al descubrimiento de propiedades o de reglas. Los métodos heurísticos no tienen el valor de la prueba sobre los resultados obtenidos con ellos, tienen más bien el valor de la conjetura o de la “regla de oro”, ni tienen tampoco la garantía de que la solución que se obtiene es única ni es la óptima. Este saber se obtiene frecuentemente mediante la observación, el análisis y el registro, como un conocimiento derivado del estudio de los hábitos de trabajo de los científicos (en ese sentido es un arte, una técnica o un conjunto de procedimientos prácticos o informales) para resolver problemas. Cada uno de los procedimientos que constituyen ese saber es un heurístico. Así podemos decir que un heurístico es cada una de las reglas metodológicas, no necesariamente formuladas como enunciados formales, en las que se propone cómo proceder y cómo evitar dificultades para resolver problemas y conjeturar hipótesis. Tiene cada vez más sentido la heurística por dos razones: Para orientar la búsqueda de soluciones en este tipo de procesos y también cuando la búsqueda exhaustiva es poco práctica porque tiene ramas poco probables o porque conducen a soluciones inviables. Entonces los métodos heurísticos son utilizados para acelerar el proceso de encontrar una solución satisfactoria a través de atajos cognitivos, para aliviar la carga de tomar una decisión. Los ejemplos de este método incluyen el uso de una regla de oro, una conjetura, un juicio intuitivo, el

---

<sup>2</sup> La descripción que presentamos está obtenida, salvo giros y adaptaciones al formato de paper, del post [Enseñanza Universitaria en línea: MOOC, aprendizaje divergente y creatividad \(II\)](#) (Zapata-Ros, 2014a))

Principio de Parsimonia (Navaja de Occan) o ciertos estereotipos (no sociales) que se forman en el devenir de los trabajos de los investigadores o creativos.

También se considera de forma consensuada que la heurística es un rasgo propio de los humanos. No es un producto original sino derivado otros procesos como son la creatividad y de lo que se conoce como pensamiento lateral o pensamiento divergente.

La heurística como disciplina tiene múltiples vertientes. La reclaman los matemáticos, los especialistas en lógica, los psicólogos, los pedagogos e incluso los filósofos. De hecho puede incluirse en todos esos dominios pues responde a algunos de sus objetivos. Sin embargo quien tiene más experiencia y la ha desarrollado más en la resolución de problemas ha sido Polya y en la enseñanza en la que hay que resolver problemas, de la resolución de problemas. No confundir con el aprendizaje o la enseñanza basada en problemas. Hablamos de la enseñanza de las matemáticas en diversos niveles.

Según Pólya (1945) “La heurística moderna trata de comprender el método que conduce a la solución de problemas, en particular las operaciones mentales típicamente útiles en este proceso. Son diversas sus fuentes de información y no se debe descuidar ninguna. Un estudio serio de la heurística debe tener en cuenta el trasfondo tanto lógico, como psicológico; no deben descuidarse las aportaciones al tema hechas por autores tales como Pappus, Descartes, Leibniz y Bolzano, pero debe apegarse más a la experiencia objetiva. Una experiencia que resulta, a la vez de la solución de problemas y de la observación de los métodos del prójimo, constituye la base sobre la cual se construye la heurística.”

Así la heurística tiene su base en la sistematización de la experiencia de resolver problemas a partir de cómo lo hacen los expertos. Para ello da una serie de pautas como son: Cómo analizar el problema, concebir un plan, ejecutar el plan y utilizar técnicas recursivas descomponiendo problemas en problemas similares más sencillos.

### **Pensamiento lateral y pensamiento divergente.-**

---

El pensamiento lateral (*lateral tinquen*) es, en expresión introducida por Edward de Bono (1968, 1970 y 1986):

*El pensamiento lógico, selectivo por naturaleza, ha de complementarse con las cualidades creativas del pensamiento lateral. Esta evolución se aprecia ya en el seno de algunas escuelas, aunque la actitud general hacia la creatividad es que constituye algo bueno en sí pero que no puede cultivarse de manera sistemática y que no existen procedimientos específicos prácticos a ese fin. Para salvar este lapso en la enseñanza se ha compuesto este libro, que tiene como tema el pensamiento lateral, o conjunto de procesos destinados al uso de información de modo que genere ideas creativas mediante una reestructuración perspicaz de los conceptos ya existentes en la mente. El pensamiento lateral puede cultivarse con el estudio y desarrollarse mediante ejercicios prácticos de manera que pueda aplicarse de forma sistemática a la solución de problemas de la vida diaria y profesional. Es posible adquirir habilidad en su uso al igual que se adquiere habilidad en la matemática y en otros campos del saber.*

En cualquier caso, el “pensamiento lateral” se ha difundido como paradigma dentro del área de la psicología individual y de la psicología social. Es la forma de pensamiento que está en la génesis de las ideas que no concuerdan con el patrón de pensamiento habitual. La ventaja de este tipo de pensamiento con respecto a cualquier otro radica en

evitar, al evaluar un problema, la inercia que se produce en esos casos producida por ideas comunes o comúnmente aceptadas, que limita las soluciones al problema. El pensamiento lateral ayuda pues a romper con ese esquema rígido de pensar y de formularse las ideas en el aprendizaje, y por consecuencia posibilita obtener ideas creativas e innovadoras. El principio contrario es igualmente cierto, estar en un contexto de ignorancia y de prejuicios o de mediocridad inhibe el pensamiento lateral, divergente, y la creatividad.

Polya y Bono estudian los recursos del pensamiento divergente. Estos recursos empleados en educación, insertos en estrategias y métodos educativos, producen unos aprendizajes distintos, constituyen el aprendizaje divergente. Es un aprendizaje que está en el origen y en la práctica de los estudios de las artes y de los oficios, es común en los talleres de los artistas, de los artesanos y de los científicos e investigadores. En general allí donde se produce creación. De esta forma se puede considerar aprendizaje divergente como aquel que utiliza los recursos del pensamiento divergente.

### **Creatividad<sup>3</sup>.**

---

El pensamiento divergente y el pensamiento convergente son tratados en relación con la creatividad por Mihály Csíkszentmihályi (1998) en su libro *Creativity: Flow and the psychology of discovery and invention*, traducido y publicado por Paidós como *Creatividad: el flujo y la psicología del descubrimiento y la invención* (págs. 83 y 84). El libro no es solo un estudio sobre una amplia variedad de comportamientos, hábitos, e ideas de individuos que han realizado aportaciones sustanciales sobre las cuales hay consenso de su carácter creativo, sino que establece un marco epistemológico y teórico de lo que es la creatividad como facultad humana y como fenómeno (un requisito de la creatividad es su validación social).

Csíkszentmihályi, como hemos visto que lo hace Pólya, coincide en que la creatividad no es consecuencia exclusivamente del pensamiento divergente sino de una combinación de ambos pensamientos, el convergente y el divergente, y desde luego sin el primero no podría producirse aunque el *insight* lo produzca el segundo. Señala que los creativos, “quienes producen una novedad aceptable en un campo, parecen capaces de usar bien dos formas opuestas de pensamiento: el convergente y el divergente.” Éste sería uno de los principales rasgos de la creatividad. El pensamiento convergente es el pensamiento que sirve para estructurar los conocimientos de una forma lógica y para aplicar sus leyes. Por decirlo de forma simplificada es el que se mide por los test de CI, y es condición indispensable para establecer modelos donde se resuelven los problemas bien definidos, que tienen soluciones validables, mediante un procedimiento sin ambigüedades. Pero hay otro pensamiento, es el que guía la acción investigadora hacia las soluciones, y sobre todo el que conduce a unas soluciones no convencionales, e implica fluidez y capacidad para generar una gran cantidad de visiones e ideas sobre el problema que se trabaja, para cambiar de unas a otras, y para establecer asociaciones inusuales. Es el pensamiento divergente, como hemos visto. Estas variables — capacidad de orientar la indagación, fluidez, facilidad para generar ideas, para cambiar de marco y para establecer asociaciones inusuales— son las que se tienen en cuenta y se miden en los test de creatividad, y las habilidades que se trabajan en la mayoría de los talleres de creatividad.

---

<sup>3</sup> Esta descripción está prácticamente transcrita adaptada del post de RED-Hypotheses *Enseñanza Universitaria en línea: MOOC, aprendizaje divergente y creatividad (III)* (Zapata-Ros, 2014b)



Pero hay otros factores que también tienen que ver en la forma como se organiza la atención docente. Nos referimos por ejemplo a la valoración de las opciones que se eligen o de las soluciones para los problemas. Es importante que, en un sistema orientado a captar más la creatividad, un alumno cuyo pensamiento sea fluido, flexible y generador de soluciones originales, tiene más probabilidad de ofrecer creaciones. Por tanto, tiene sentido cultivar el pensamiento divergente en el aula y en talleres y laboratorios. Pero es igualmente importante que exista alguien, un profesor adecuado, capaz de escoger y orientar hacia la práctica las ideas más apropiadas de entre las que se generan.

Conviene aclarar en este contexto que el objetivo principal de un programa de este tipo no es la generación de novedades, sino crear un clima donde las innovaciones significativas se produzcan, o al menos no sean inhibidas por el propio sistema.

Csikszentmihalyi (1996) lo explica con claridad:

*Sin embargo, sigue existiendo la sospecha persistente de que en los niveles más elevados de logro creativo la generación de novedad no es la cuestión principal. Un Galileo o un Darwin no tuvieron tantas ideas nuevas, pero aquellas a las que se aferraron fueron tan fundamentales, que cambiaron la cultura entera. Así mismo, los individuos de nuestro estudio a menudo afirmaban que sólo habían tenido dos o tres buenas ideas en toda su trayectoria profesional, pero que cada idea fue tan fecunda que los mantuvo ocupados durante toda una vida de pruebas, hallazgos, elaboraciones y aplicaciones.*

Sin embargo para que estas ideas se produzcan es necesario: Un contexto adecuado donde haya un enlace con el aprendizaje convergente, un contexto que pueda hacer posible que fructifiquen. Es como una huerta, hace falta allanar un campo, labrarlo, abonarlo, regarlo, sembrar, luchar contra las heladas,...para al final recoger la cosecha. Solo que en este caso la cosecha es tan escasa como importante.

### **Resolución de problemas.-**

---

En realidad el pensamiento computacional es una variante del dominio metodológico que se conoce como “resolución de problemas”. Es una restricción de la resolución de problemas a aquellos problemas cuya resolución se puede implementar con ordenadores. En este caso es muy importante distinguir que los aprendices no son sólo los usuarios de la herramienta, sino que sobre todo se convierten en los constructores y en los autores de las herramientas.

Para eso los alumnos utilizan procedimientos, conjuntos de objetos de conocimiento y conceptos que constituyen dominios que tratamos de forma separada en este escrito. Como son la abstracción, la recursividad y la iteración- Los utilizan para procesar y analizar los datos de cara a crear métodos de resolución de problemas, y crear artefactos reales y virtuales para resolverlos. El pensamiento computacional de esta forma se puede considerar también como una metodología de resolución de problemas que se puede automatizar.

La otra vinculación del pensamiento computacional con la resolución de problemas lo constituye la visión que se puede desarrollar en los alumnos y que se manifiesta en el aula para encontrar soluciones a problemas a través del ordenador. Para esta visión también son importantes elementos de pensamiento que veremos con entidad propia como son el desarrollo de herramientas para resolver problemas por métodos de ensayos

progresivos y error y por las posibilidades que tienen los ordenadores para trabajar en “una atmósfera de entender las cosas juntos”.

### **Pensamiento abstracto.-**

---

Es la capacidad para operar con modelos ideales abstractos de la realidad, abstrayendo las propiedades de los objetos que son relevantes para un estudio. Una vez obtenido el modelo abstracto de la realidad se estudian sus propiedades, se extraen conclusiones o reglas que permiten predecir los comportamientos de los objetos. El pensamiento abstracto por excelencia es el pensamiento matemático, la geometría, etc.

El pensamiento abstracto tiene mucho que ver con la edad del niño, no solo porque según las teorías de Piaget y las de la Psicología genética, consideran que la abstracción es producto del desarrollo, de la maduración cognitiva del niño, sino porque los mecanismos de abstracción son muy distintos según la edad del niño, existiendo desde las primeras etapas. Para un niño de dos años, “el día después del día de mañana” es un concepto muy abstracto. Para un estudiante de la universidad, el día después de mañana es un concepto relativamente concreto, sobre todo si lo comparamos con las ideas realmente abstractas o muy abstractas como son el Teorema de Bayes o el principio de indeterminación de Heisenberg.

Por supuesto, hay muchos niveles de abstracción entre estos dos extremos. Una componente importantísima en el diseño curricular es tener en cuenta el proceso de desarrollo intelectual que supone este proceso: Transitar gradualmente de pensamiento muy concreto al pensamiento abstracto, en función del desarrollo individual y esto tenerlo en cuenta en la presentación de los contenidos y destrezas a desarrollar. Esta cautela tiene que ver mucho con otra cuestión muy frecuente: Considerar lo abstracto como difícil y lo concreto como lo fácil, cuando muchas veces lo que sucede es que se presenta una habilidad o un concepto para ser aprendido en un momento poco adecuado, no por la edad exclusivamente sino sobre todo por las condiciones en que se produce el aprendizaje.

Frecuentemente la capacidad de pensar en abstracto, para diferenciarlo de pensar en concreto, se confunde con la capacidad de transferir lo aprendido a partir de un contexto a otro. Por ejemplo, un estudiante tiene un dominio razonable del pensamiento abstracto si entiende la organización de un texto en clave de ensayo independientemente de que lo aprenda en la asignatura de lengua española y luego sin un aprendizaje adicional pueda aplicar lo aprendido para escribir un ensayo en la asignatura de sociales. También se puede confundir con el pensamiento generativo. El concepto de objeto de conocimiento generativo (Zapata-Ros, 2009) (básico para la comprensión del concepto de *Generative Learning Object* (Leeder et al, 2004)) hace referencia a un conjunto de objetos de conocimiento que lo tienen todo en común excepto a lo más unos valores de adaptación o de contextualización, asignables a unos parámetros definidos o decididos por el aprendiz.

### **Recursividad**

---

A veces un problema por su tamaño, o porque depende de un número natural (o de un cardinal) no puede ser resuelto por sí mismo pero puede ser remitido a otro problema de las mismas características o naturaleza pero más pequeño o dependiendo de un cardinal

menor, que sí puede ser resuelto, o nos puede dar la pista de una regla de remitir problemas a problemas menores (regla de recurrencia). Y en ambos casos nos permite resolver el problema. A estos métodos, que son así considerados unos métodos de resolución de problemas, se les llama recursividad o recurrencia.

Con el término recursividad también se quiere en otras ocasiones abordar una forma de conceptualizar, de definir, objetos de conocimiento o ideas: De esta forma se dice que están definidos por recurrencia.

En esta forma de abordar el conocimiento se ha visto por un lado una forma más útil, o más económica cognitivamente, de abordar la resolución mediante procesos automatizados, o una forma más eficaz y elegante de abordar conceptos y definiciones que permiten integrarlas más eficientemente desde el punto de la lógica en un sistema teórico.

Presenta un dificultad que hace necesaria una predisposición mental que no siempre es frecuente ni fácil de alcanzar según qué individuos.

En matemáticas es muy frecuente:

El concepto de potencia de exponente natural se puede definir como un producto de factores iguales:

$$a^n = a.a.a\dots a$$

O bien mediante una formulación recursiva: una potencia se define en función de la potencia anterior

$$a^n = a.a^{n-1}$$

excepto si el exponente vale 0, que entonces es  $a^0=1$ , a esto se le llama cláusula de parada.

Con la función factorial sucede igual, se puede definir como un producto de factores consecutivos decrecientes:

$$n! = n.(n-1).(n-2)\dots 3.2.1$$

O con una fórmula recursiva:

$$n! = n.(n-1)! \text{ y } 1! = 1$$

La experiencia me dice que los alumnos cuando se les presenta adecuadamente la primera formulación la suelen comprender y aprender con relativa facilidad. No así la segunda. Necesita un tipo especial de forma de pensar. Necesita utilizar el pensamiento recursivo.

Sin embargo la segunda forma tenía indudables ventajas no solo para programar sino incluso de economía de pensamiento y de ejecución. Mis alumnos de secundaria comprendían perfectamente esto cuando les ponía el siguiente ejemplo:

Una azafata quiere comprobar que todos los pasajeros tienen que llevar ajustado y bien sujeto el cinturón de seguridad. En principio podría hacerlo de dos formas: recorriendo el pasillo y verificando que todos lo llevaran correctamente o bien asegurando solo dos cosas:

Que cada pasajero comprobase que lo lleva igual que el anterior y que el primero lo llevase bien.

La recursividad es algo que va más allá de las matemáticas o de la computación, es propiamente una forma de pensar: Pensar sobre el pensamiento, también tiene un ámbito de conocimiento o de modelado en la psicología: la metacognición. De hecho en un conocido trabajo “El pensamiento recursivo” Michael C. Corballis (2007) dice que “La facultad de pensar sobre el pensar constituye el atributo crítico que nos distingue de todas las demás especies”.

Los fractales, como caos y sistemas dinámicos, son también estructuras recursivas. Los podemos encontrar incluso en la fundamentación del conectivismo (Siemens, September 2014, a través de Zapata-Ros, 2014).

Pero también constituye la base de la geometría (Zapata-Ros, 1996a, 1996b) y del arte fractal (Zapata-Ros, 2013)

La expresión fractal viene del latín *fractus*, que significa fracturado, roto, irregular. La expresión, así como el concepto, se atribuyen al matemático Benoit B. Mandelbrot, del Centro de Investigación Thomas J. Watson, que la empresa IBM tiene en Yorktown Heights, Nueva York, y aparecen como tal a finales de la década de los setenta y principios de los ochenta (Mandelbrot, 1977 y 1982). Aunque anteriormente Koch, Cantor y Peano entre otros, definieron objetos catalogables dentro de esta categoría, pero no reconocidos como tales.

El concepto de fractal se puede abordar desde distintos puntos de vista, sin embargo se acepta comúnmente que un fractal es un objeto geométrico compuesto de elementos también geométricos de tamaño y orientación variable, pero de aspecto similar. Con la particularidad de que si un objeto fractal lo aumentamos, los elementos que aparecen vuelven a tener el mismo aspecto independientemente de cual sea la escala que utilizamos, y formando parte, como en un mosaico de los elementos mayores. Es decir estos elementos tienen una estructura geométrica recursiva. Si observamos dos fotografías de un objeto fractal con escalas diferentes (una en metros y otra en milímetros, por ejemplo) sin nada que sirva de referencia para ver cuál es el tamaño, resultaría difícil decir cuál es de las ampliaciones es mayor o si son distintas. El que cada elemento de orden mayor esté compuesto, a su vez, por elementos de orden menor, como sucede con las ramas de un árbol es lo que da estructura recursiva a los fractales.

Para representar gráficamente un fractal basta por tanto encontrar la relación o la ley de recursividad entre las formas que se repiten. Es decir encontrar el objeto elemental y la ley de formación y establecer el algoritmo gráfico. Es por esto que lenguajes como LOGO se avienen tan bien para representar fractales.

Volviendo a los fundamentos del lenguaje y de la función de pensar, Rosas (2012) en la glosa que hace le artículo de Corballis (2011) *The Recursive Mind. The Origins of Human Language, Thought, and Civilization*, nos habla, en el contexto del resurgimiento del Programa Minimalista de Chomsky, más allá de una búsqueda formal centrada en el código y en las reglas que subyacen en éste, de una búsqueda desde un plano superior, más abstracto, que observa la capacidad lingüística, insertándola en un sistema funcional que la explica mediante determinados principios no específicos de esta facultad, como son la arquitectura estructural y a las restricciones de su desarrollo (como es la eficiencia computacional) o los relativos al análisis de datos. Que en un plano multidisciplinar hace consideraciones a criterios que no son propios de ese dominio.

Rosas (2012) entiende que “eso ha dado lugar a la aparición de una nueva capacidad en la que pocas veces se había reparado (sic) y mucho menos ocupado el foco principal de estudio: la recursividad.

En un artículo muy citado en este sentido, Hauser, Chomsky y Fitch (2002 y 2005) proponen la recursividad como la única capacidad estrictamente merecedora de la denominación “lenguaje”, de ahí el nombre que le otorgan: “Facultad del lenguaje en sentido estricto”. Ésta sería exclusiva de la especie humana y estaría presente sólo en el lenguaje.

En el primer capítulo, de *The Recursive Mind: The Origins of Human Language, Thought, and Civilization*, respondiendo a la pregunta ¿qué es la recursividad?, Michael C. Corballis (2014) sostiene que en 1637, el filósofo francés René Descartes cuando escribió la frase inmortal “Je pense, donc je suis” o “Cogito, ergo sum”, en esencia al hacer esta declaración, Descartes lo que hizo no sólo fue pensar, que estaba pensando en el pensamiento, y que le llevó a la conclusión de que existía, sino que estaba constatando la existencia del pensamiento como recursividad. Esta consideración le da paso a Corballis para pensar en un papel más general de la recursividad en nuestros procesos mentales, y a sostener que es la característica principal que distingue a la mente humana de la de otros animales. Es la base de nuestra capacidad no sólo para reflexionar sobre nuestra propia mente, sino también para simular la mente de otros. Mediante ella se nos permite viajar mentalmente en el tiempo, a realizar una inserción de la conciencia del pasado o del futuro en la conciencia presente. La recursividad es pues el ingrediente principal para distinguir el lenguaje humano de todas las otras formas de comunicación animal.

### **Iteración.-**

---

Siempre que hablamos de iteración pensamos en procedimientos repetitivos como los que utilizamos cuando aprendimos o cuando enseñábamos BASIC, Pascal, LOGO, o C++, y más recientemente Java o Phyton. Lo asociamos a bucles, a instrucciones *FOR TO, while, do-while, repeat,...* y a diagramas de flujo. En definitiva, era difícil hablar de iteración sin pensar en la construcción de algoritmos repetitivos. Sin embargo pocas veces pensamos que hay aprendizajes básicos, en las primeras etapas de desarrollo, donde se pone en marcha un sistema de pensamiento de este tipo. Pensemos por ejemplo en la adquisición que hacen los niños de las ideas sobre fracciones, números racionales, o incluso en los números reales, en la representación decimal, en la notación decimal de números reales, y en su representación ¿qué son sino más que procedimientos iterativos? También podríamos pensar en sistemas de medición, de magnitudes de peso, masa, volumen, superficie,... ¿qué son estos procesos sino sistemas de representación conceptual iterativas?



Fig. 3 La escalera de Bramante es un claro ejemplo de iteración en el diseño arquitectónico.

Son particularmente importantes las aportaciones en este sentido de Mack (2001); Olive & Vomvoridi (2006). E igualmente las de Steffe & Olive (2010) para el desdoblamiento de operaciones, que juega un papel crítico en la construcción de esquemas de fracciones por los estudiantes, tales como el esquema de fracción iterativo. Otras investigaciones en este sentido fueron las de Hackenberg, (2007) y Steffe (2004).

Jesse L.M. Wilkins, Anderson Norton y Steven J. Boyce (2011) prueban empíricamente la validez de unos esquemas iterativos llamados *partitive unit fraction scheme* (PUFS), y de forma más general *partitive fraction scheme* (PFS), para el aprendizaje de fracciones. Aunque este estudio ha sido criticado por Norton & Wilkins (2010), debido a la existencia de caracterizaciones del esquema divergentes.

La iteración es una componente pues importante del pensamiento computacional, con una extensa proyección en otras representaciones cognitivas y en procedimientos que son la base de importantes actividades y tareas, como por ejemplo lo que hemos mencionado en relación con la medida y la representación de magnitudes y valores.

Pero no sólo es relevante a ese nivel, la iteración es la base de procedimientos complejos y está en la resolución de problemas con más alcance o más impacto que lo que supone su definición en una primera aproximación.

Hay un problema conocido por el problema del viajante (TSP por las siglas de *Traveling Salesman Problem*). Sucintamente consiste en, *dada una lista de poblaciones, ¿cuál es la ruta más corta posible que pasa por cada localidad exactamente una vez y termina en el lugar de salida?*. Hay muy variados métodos de resolución y todos son iterativos. El primero es el procedimiento *a mazazos*, también llamado “*búsqueda de fuerza bruta*” (*Brute-force search*). Como el nombre sugiere se trata de obtener como itinerarios posibles todas las permutaciones de las ciudades, con la distancia recorrida en cada caso, y obtener el mínimo de ellas. Es un método exacto y factible, con muy poca programación, en un ordenador. **Es un ejemplo de método**

**iterativo.** Tiene problemas de tiempo porque es una función de un factorial, pero hay algoritmos para abreviar el tiempo.

Este es un método heurístico. Se podría pues, como vimos, incluir en varias categorías en cuanto a tipo de pensamiento. Casi nunca son puros.

Hay otros métodos heurísticos también recursivos como es el del vecino más cercano (NN, *nearest neighbor*) (también llamado algoritmo voraz), que como el propio nombre indica en cada caso elige la ciudad más cercana sin visitar como su próximo movimiento.

Luego hay métodos algorítmicos con hamiltonianos y todo eso.

Pero lo importante para nuestro propósito de ver la importancia de la iteración en métodos basados en grafos es al menos traer la referencia de los métodos *op*:

El intercambio par a par simple, o técnica *2-opt*, que supone en cada iteración la eliminación de dos aristas que se cortan y su reemplazo por con dos aristas diferentes, que no se crucen, y que reconecten los fragmentos creados por la eliminación de las aristas por este método, obteniendo así un camino más corto.

Éste es un caso especial del método general heurística *k-opt* o heurística Lin-Kernighan. Donde en un recorrido dado se eliminan *k* aristas que se cortan en al menos un punto por un conjunto que no se corta en ningún punto.

Un ejemplo interesante lo podemos ver este caso

Este problema y los algoritmos que se han desarrollado apartir de él han permitido un enfoque interesante para otro problema, el del embarque en aviones, a Jason Steffen (2008), y al desarrollo de un algoritmo muy práctico y de indudable interés económico para la secuencia de embarque de pasajeros en los aeropuertos.

El cambio que supone en la práctica modificar la forma en que la gente embarca en los aviones, el ahorro de tiempo multiplicado por el número de embarques que se producen en todo el mundo, a lo que se unen los tiempos de espera en vuelos combinados, suponen una solución de gran envergadura directa, a la economía de tiempo, y de dinero, de sueldos, de energía,... Y esto sólo si lo planteamos directamente. Pero es que además hay un ahorro indirecto y un bienestar añadido: El que supone que la gente llegue a tiempo a su trabajo, a su negocio, a sus vacaciones o a estar con su familia. Ventajas que son muy de tener en cuenta..., todo por una cuestión de iteración. Y como consecuencia de una forma de pensar que primero se planteó exclusivamente como pensamiento computacional.

### **Métodos por aproximaciones sucesivas. Ensayo – error.**

---

El método de resolución de problemas por aproximaciones sucesivas, o por ensayo-error, constituye igualmente un procedimiento que utilizamos, confrontando las ideas que nos formamos con la realidad tal como la percibimos, en acciones percepciones y en la formación de modelos cognitivos, de ideas. Sucede así en el ser humano a lo largo de toda la vida, desde las primeras etapas de desarrollo, en la que los niños comienzan a conocer la realidad, el mundo que les rodea. Utilizan los sentidos, la experimentación y la representación de las ideas obtenidas de las experiencias, para aceptar o rechazar el conocimiento que la realidad les ofrece y para inducirlo. Ese mecanismo forma parte del

desarrollo humano, pero también lo encontramos en los fundamentos de la ciencia. Así lo encontramos en multitud de ámbitos y dominios del saber y de la técnica. Constituye la base de las ideas de Popper (1934) que fundamentan el método científico. Lo encontramos igualmente como uno de los procedimientos que más frecuente utilizan los programadores, de forma espontánea y subyacente, en casi todas las fases de su trabajo. También constituye la esencia de la ayuda pedagógica que los maestros y tutores hacen a sus alumnos para guiarles en estos procesos de ensayo error y que no se pierdan o se distraigan por caminos inapropiados.

A Karl R. Popper se le considera el padre del método científico tal como se conoce en la actualidad, pero sobre todo es uno de los pensadores contemporáneos más influyentes, cuyas teorías epistemológicas y sociopolíticas han ido más allá del estricto ámbito del método científico. Hasta él el método que utilizaba la ciencia era eminentemente deductivo. A partir de él todo cambia: La ciencia sigue siendo inductiva, pero su gran aportación ha sido que esta inducción ha avanzado a través del método hipotético-deductivo.

Así según Popper (1934), el método científico no usa un razonamiento inductivo, sino un razonamiento hipotético-deductivo (que simplificado se conoce como método de ensayo error o por aproximaciones sucesivas). Como en el caso del razonamiento inductivo, se pasa desde los datos que contrastan una hipótesis a una conclusión sobre ésta, es decir va de lo particular a lo general, en dirección inductiva. Sin embargo el método no es el de la inducción como razonamiento o inferencia. Sostiene que materialmente no es posible inducir o verificar todas las hipótesis o teorías (no es posible explorar todas las situaciones posibles para ver si la teoría se mantiene), ni siquiera hacerlo con las más probables. Además, los científicos en general buscan teorías altamente informativas. Otra cosa son los informáticos, que aplican con frecuencia la navaja de Occam.

La cuestión clave en la ciencia es **qué criterio guía la búsqueda o el avance a través de las hipótesis que se eligen sucesivamente**. En esta cuestión tiene bastante que decir la creatividad y el pensamiento divergente, según vimos en otra ocasión.

En el aprendizaje, el mecanismo en esencia es el mismo. Pero en este caso es el papel que juega el tutor lo esencial, como veremos, sin despreciar los elementos naturales de motivación que el método que utilizamos en cada caso posee para el alumno. De esta forma el tutor ha de guiar de forma adecuada y sin ser invasivo el procedimiento para que el alumno tampoco desista, y este proceder es distinto en cada caso.

Pero volviendo a Popper y al método hipotético deductivo. Lo que se hace realmente en el proceso, en cada paso, es proponer una hipótesis como solución tentativa del problema particular, confrontar la predicción deducida, mediante la hipótesis, con la experiencia, y evaluar si la hipótesis se rechaza o no por los hechos (contraste de hipótesis). La cuestión es que con este método no verificamos las teorías, sólo las aceptamos cuando resisten el intento de rechazarlas. Por tanto, el contraste radica en la crítica o, si estamos en ciencias sociales, en el intento serio de *falsación*, es decir, la eliminación de la parte del error dentro de una teoría, para rechazarla, si es falsa, y sustituirla por otra. Como hemos dicho el objetivo del método es la búsqueda de teorías verdaderas.

Según Popper (1934) este método, el actualmente aceptado como método científico, utiliza sólo y de forma sistemática reglas metodológicas (no lógicas), para tomar decisiones. Reglas o principios metodológicos que tiene como base casi exclusivamente



dos principios: La creatividad y la crítica. Hay que ser creativo y crítico. Hay que proponer hipótesis audaces y someterlas a tests experimentales rigurosos. La lógica juega un papel fundamental como elemento que rige las decisiones y la elaboración de hipótesis que, mediante su contraste, confrontarán los hechos con las teorías convirtiéndolas en evidencias.

David Wood y Heather Wood (2006) en su ya clásico libro *Vygotsky, Tutoring and Learning* señalan como justificación de la acción tutorial en el aprendizaje la investigación que con frecuencia citamos como referencia y que hemos rescatado de Bloom (1984) para el aprendizaje personalizado con tecnología, nos referimos al problema de dos sigmas. Como ya vimos (Zapata-Ros, 2013 y Bloom, 1984) se trata de un análisis comparativo de la eficacia, relativa, entre la enseñanza en clase convencional y la tutoría uno a uno (humana). Anderson (1993) y sus colegas sugiere que algunos de los beneficios señalados por Bloom podrían obtenerse por medio de sistemas informáticos diseñados para individualizar el aprendizaje. Bloom concluyó que la instrucción mentorizada (uno a uno) conduce a una mejora, en la campana de Gauss del aprendizaje, en dos desviaciones típicas sobre la enseñanza de clases convencional (es decir, alrededor del 98% de los alumnos que se enseñan de forma individual puntúan por encima de la media para el grupo convencional). Anderson sostenía que los sistemas basados en tecnología, con tutoría personal o asistida, podrían ser diseñadas para conseguir en parte ese beneficio.

### **Contingencia e inmediatez.-**

En este planteamiento juegan un papel clave las aproximaciones sucesivas a los objetivos educativos, es decir la acción tutorial. Veamos: Una de las características centrales de lo que supone la tutoría uno a uno (y lo que en la época de Anderson eran los sistemas de tutoría inteligente), es que pueden proporcionar instrucción en el contexto de la actividad en tiempo real al alumno. En términos de Anderson, son útiles al principio: proporcionan instrucción en el contexto de la resolución del problema. En situaciones de grupo, también funciona la tutoría como ayuda en el acercamiento a la resolución de problemas por ensayo error: El alumno puede ensayar públicamente, en ambiente de clase, métodos de resolución y puede ser asesorado por el profesor y por los propios alumnos. El profesor puede ofrecer orientaciones sucesivas sobre la forma de resolver los problemas, y hacerlo sobre la marcha.

En cualquier caso es importante la idea de contingencia: la sensación de que el problema puede ser resuelto o no en función del camino elegido.

De esta forma, en la tutoría, el cuándo y el dónde la ayuda pueden ser ofrecidas por el tutor es la clave. Ha de hacerse en los momentos pertinentes, es decir, de manera contingente. Además es importante hacerlo así porque el estudiante puede ser sensible a la pérdida de tiempo y perder la motivación cuando ésta, la pérdida, sea excesiva o también porque la actividad sea confusa y sin fruto. Para evitar este efecto, el de la confusión, la respuesta debe ser inmediata a los errores de los alumnos (otro principio del ensayo error: que no se pierda la inmediatez).

El tutor debe detectar, en el lugar y en el momento que se produzca, la dificultad de aquel aprendiz que comprenda insuficientemente el tema que es objeto de aprendizaje. De esta forma el tutor puede tener que intervenir con frecuencia para reparar el error y mostrar al alumno qué hacer. Así el objetivo es apoyar en la resolución en aproximaciones sucesivas hasta un desempeño competente. Por último el tutor no debe interferir con la actividad de estudiante exitoso. Y en la medida que el alumno aprenda,

la acción del tutor se desvanece. Estas capacidades del tutor que Anderson prescribía para el tutor inteligente, o para el sistema experto, se obtenía a partir de la modelización de los tutores humanos. Por esa misma razón ese conjunto de prescripciones siguen siendo válidas en el caso real, y constituyen una aplicación en este caso de los principios popperianos de las aproximaciones sucesivas.

### **Métodos colaborativos.- ¿Hacer cosas juntos o entender cosas juntos?**

---

En este recorrido toca llegar al final, aunque quizá haya más componentes del pensamiento computacional de los que hemos visto, y esperamos futuras críticas y propuestas. Vamos a concluir con el trabajo colaborativo.

Expresiones como trabajo colaborativo o aprendizaje colaborativo son lugares comunes en la práctica de la enseñanza y en las teorías del aprendizaje. Tienen su origen remoto en los métodos socráticos, en el aprendizaje vicario y más recientemente en las teorías de Vygostky, en las del aprendizaje situado de Merrill y en el socioconstructivismo. Y han adquirido plena vigencia en los entornos conectados de aprendizaje. Si bien las aportaciones más fecundas en el mundo del aprendizaje con la ayuda de la tecnología se deben a David Jonassen, Mark Davidson, Mauri Collins, John Campbell, y Brenda Bannan Haag (1995).

En el mundo computacional: La complejidad de desarrollos y arquitecturas hace inconcebible el trabajo aislado. Tienen que producirse fuertes flujos de trabajo y de comunicación que hagan posibles proyectos comunes en equipos amplios. De hecho se ha desarrollado una ética, casi una mística, conocida y popularizada por Pekka Himanen (2002) como la ética del hacker, basada en la emoción por compartir más que en el valor económico del trabajo propio de la ética de Weber, la ética protestante del trabajo.

En una buena parte esta disposición a compartir y al trabajo colaborativo constituye un elemento para la formación en valores del pensamiento computacional. Pero también implica un desafío, no todo el mundo de forma inicial acepta compartir, implica un compromiso e implica una técnica.

La definición más amplia pero igualmente imprecisa e insatisfactori, de “trabajo colaborativo” es la que da Dillenbourg (1999): Trabajo colaborativo es el que se produce en una situación en la que dos o más personas aprenden o intentan aprender algo juntos.

Es obvio que al menos hay tres imprecisiones en los elementos de esta definición, que se pueden interpretar de diferentes maneras:

“Dos o más” es ¿un par?, ¿un pequeño grupo (3-5 individuos)?, ¿una clase (20-30 sujetos)?, una comunidad (unos pocos cientos o miles de personas), ¿un MOOC?, ¿una sociedad (varios miles o millones de personas) ... ¿cualquier nivel intermedio?. Esto da lugar a situaciones de aprendizaje completamente distintas, cada una de las cuales lleva aparejado un análisis que de forma no simple es muy diverso. Los entornos de los que estamos hablando y que permiten un trabajo fecundo son aquellos que permitan de forma eficiente a cada individuo procesar la información que genera el resto.

“Aprender algo” puede ser interpretado como “seguir un curso con provecho”, es decir cumpliendo los objetivos de aprendizaje previstos, o también se puede referir de forma laxa a aprender (en el sentido de comprender solo y memorizar de forma comprensiva) el “material del curso de estudio”, o bien “realizar actividades de aprendizaje tales como

la resolución de problemas”, y en su caso óptimo que de ellas se desprenda conocimiento o elaboración, igualmente puede ser “aprender de la práctica del trabajo” que se realiza entre varios y en el que interviene la interacción.

Y en esto último es cuando interviene el último elemento de la definición: “juntos”. Que en cualquier caso implica y se debe interpretar como como una referencia a diferentes formas de interacción que, por la forma física de realizarse, origina distintos entornos y proceso cognitivos: Cara a cara, grupo o videogrupo (*hangout*), mediada por entornos de red, sociales (web social), sincrónicas o no, frecuentes en el tiempo o no, si se trata de un esfuerzo verdaderamente conjuntado y coordinado, si el trabajo se divide de una manera sistemática en un entorno colaborativo, híbrido y organizado con *affordances* a ese fin.

Combinados de múltiples formas y en relación con fines de distinta naturaleza, pero todos ellos conducentes a aumentar el material cognitivo de los participantes, y el común, constituyen ambientes que se encuentran bajo la etiqueta de “aprendizaje colaborativo”. Podemos reconocer así a parejas o tríos de aprendizaje, que se desenvuelven a través de trabajos intensivos y cortos para resolver un problema de forma conjunta y síncrona, durante una o dos horas, a grupos de estudiantes que utilizan el correo electrónico o Facebook durante un curso, o durante un año, a las comunidades de profesionales que se desarrollan vinculadas a una cultura específica a través de generaciones de participantes.

La actividad que se produce es singular y la capacitación o las competencias para obtener el máximo rendimiento son necesarias para el mundo computacional, con rasgos específicos (por ejemplo con pasarelas de datos y de resultados), pero igualmente para el resto de actividades que conducen a algún tipo de aprendizaje o de desarrollo. El análisis tiene al menos tres dimensiones como hemos visto, cada una constituye no de forma excluyente ni por separado del resto, un dominio de estudio y de investigación: la dimensión de la situación de colaboración (tamaño del grupo, período de tiempo, *affordances*,...), el tipo y características del “aprendizaje” y las formas de “colaboración”.

Referencias útiles en un primer acercamiento son:

Sobre la noción de escala en relación con las modalidades y los objetivos del trabajo colaborativo, “... los paradigmas de investigación basadas en distinciones entre lo social y lo cognitivo”, se encuentra el trabajo de Perret-Clermont, Perret y Bell (1991). Sobre las teorías de la cognición distribuida (Salomon, 1993) en que el grupo es visto como un sistema cognitivo. El proceso de construcción de un grupo de micro-cultura es estudiado por Baker, Traum, Hansen y Joiner (1999), y por Hansen, Lewis, Rugelj y Dirckinck-Holmeld (1999).

En resumen la cuestión no es tanto aprender técnicas para trabajar juntos como encontrar una cultura común, unas referencias y unas experiencias que hagan que esa forma de trabajar fluya.

### **Patrones.-**

---

Los patrones constituyen una herramienta para el análisis de la programación con una doble singularidad: Evitan el trabajo tedioso que supone repetir partes de código o de diagramas de flujo o de procedimientos que en esencia se repiten pero aplicados a contextos y situaciones distintas, y por otro lado exige la capacidad de distinguir lo que

tienen de común situaciones distintas. Esta facultad es útil en la programación pero igualmente en multitud de situaciones de la vida o de las actividades científica y profesionales, de hecho nacieron como tales en la arquitectura. Un ejemplo de actividad muy alejada a la computación es la enseñanza, la pedagogía. En ese apartado hablo de un patrón pedagógico, el que se basa en el aprendizaje activo. Hay muchas disciplinas o dominios de aprendizaje muy distintos por su objeto de tratamiento, la naturaleza de lo que se aprende y los objetivos: el qué, de lo que se aprende, pero que tienen en común que se aprende mediante la actividad. Pues bien el diseñar esas actividades para que se produzca el aprendizaje es común a los distintos ámbitos de aprendizaje. Y el patrón consiste en las pautas comunes para todos esos casos y situaciones.

El concepto de patrón y su práctica se aplica, en la computación y en otros dominios, a estructuras de información que permiten resumir y comunicar la experiencia acumulada y la resolución de problemas, tanto en la práctica como en el diseño.

De esta forma un patrón puede entenderse como una plantilla, una guía, un conjunto de directrices o de normas de diseño. Los patrones pueden entenderse desde dos perspectivas: La propia del dominio en el que estamos trabajando (la arquitectura, el diseño industrial, el diseño instruccional, etc.), o bien desde la perspectiva de los lenguajes y las técnicas computacionales que permiten el desarrollo de patrones.

Un patrón pues permite la adquisición de “buenas prácticas” y sirve como referencia para nuevas aplicaciones y casos. El almacenamiento y proceso sistemático de estos patrones permite construir corpus de información o bases de datos de referencias documentadas a las que los distintos profesionales o investigadores pueden dirigirse para sus trabajos específicos.

Los patrones tienen su origen en los patrones de diseño, o en los patrones genéricos, y sirven para aplicar en un campo cualquiera de la actividad de creación y de desarrollo, donde se quiere optimizar el trabajo intelectual haciendo más eficaz el trabajo empleado, o bien donde se quiere comunicar una parte operativa del diseño independientemente del dominio técnico del que se trate. Originalmente los patrones de diseño se deben al arquitecto Christopher Alexander (...). Posteriormente estas técnicas se han adoptado en el campo de la ingeniería de software, y de allí se han incorporado al diseño instruccional tecnológico.

Un patrón (Alexander et al., 1977) “describe un problema que ocurre una y otra vez en nuestro entorno y, a continuación, describe el núcleo de la solución de ese problema, de tal manera que el usuario puede utilizar esta solución un millón de veces más, sin tener que hacerlo de la misma manera dos veces”.

Especial importancia merecen los patrones instruccionales, aunque el término igualmente acuñado puede ser el de patrones pedagógicos (Pedagogical Patterns

Project, 2008), porque sirven de comunicación en equipos pluridisciplinarios en los que concurren técnicos en computación, diseñadores instruccionales y profesores.

Se utiliza indistintamente el término patrón y plantilla como traducciones del término original *pattern*. Sin embargo creemos que es más apropiado patrón que pauta o plantilla. Pauta o conjunto de pautas hace referencia a procedimientos a aplicar de forma algorítmica o automática en un contexto donde se cumplan determinadas condiciones. Mientras que patrón como hemos visto hace referencia a un esquema completo de resolución de un problema teniendo solo en presencia la perspectiva del patrón (por ejemplo si a un problema de aprendizaje queremos aplicar el patrón de “aprendizaje autónomo”, tendremos que aplicar, para la resolución del problema, la alternativa de las posibles que implique al máximo la participación del alumno sin ayuda externa).

El objetivo en todo caso es captar la esencia de la buena práctica de una forma resumida e independiente (abstrayendo los elementos más significativos) de manera que pueda ser fácilmente comunicada a los que la necesitan en un contexto de condiciones distinto. En su naturaleza el patrón puede ser igualmente la presentación de esta información (de las buenas prácticas, los conocimientos expertos, las soluciones a problemas,...) de una forma accesible y sistematizada, de manera que para cada nuevo diseñador pueda aprender o tener en cuenta lo que se conoce por expertos que hayan resuelto ya el problema en cuestión, y sea fácil la transferencia de conocimiento dentro de la comunidad.

De esta forma en esencia un patrón resuelve un problema. Este problema debe ser una naturaleza tal que se repita en distintos contextos. En el ámbito de la educación tenemos muchos problemas de esta naturaleza: Son por ejemplo los que tienen que ver con la motivación, o con determinadas motivaciones, de los estudiantes, la elección de los materiales para cada situación de aprendizaje y la secuencia de los contenidos, los conocimientos previos, las ideas implícitas, los criterios de evaluación de los estudiantes,...

*Pedagogical Patterns Project* (2008) establece una clasificación de los patrones pedagógicos en tres categorías, basados en otras tantas formas de aprendizaje:

- Aprendizaje activo. Un patrón de este tipo se basaría en un conjunto de actividades que involucren a los alumnos de manera activa. El patrón se construye utilizando algún problema concreto que a menudo pueden ocurrir en un entorno de enseñanza y que maximice la atención del alumno por estar implicado en la resolución, o por sus experiencias pasadas o presentes, por su carácter real etc.
- Aprendizaje experimental. Un patrón de este tipo se basaría en lo que es necesario aprender mediante la experimentación o bien mediante las experiencias pasadas de los alumnos estudiantes.
- Enseñando desde diferentes perspectivas. Un patrón de este tipo se basaría en la bondad de los aprendizajes que supone por los alumnos el estudio de los recursos educativos desde diferentes perspectivas, tratando los siguientes problemas: preparar al estudiante para el mundo real, hacer uso de diferentes perspectivas por pares (por ejemplo utilizando personal profesional de empresa).

A los que podríamos añadir un patrón pedagógico para el “aprendizaje autónomo”, con abundancia de elementos de metacognición.

La naturaleza de estos problemas (los que originan el patrón) es, como vemos, que se pueden repetir de forma diferente (en las partes no sustanciales) cada vez. Y cuando aparece un problema de este tipo conlleva consideraciones a tener en cuenta para tomar las decisiones en la selección del procedimiento a utilizar para la resolución. Estas consideraciones son las que influyen en los expertos a optar por una u otra resolución. Son las consideraciones que el patrón presenta para que las tengamos en cuenta en nuestra elección y que nos pueden acercar o alejar de una buena solución al problema.

Un patrón presenta un problema y una solución. O bien el criterio de la solución. De tal forma que los criterios que deben aplicarse deben hacer que la solución sea la más acertada para el problema planteado.

Para desarrollar un patrón se utiliza un “lenguajes de patrón”. Su naturaleza y principales características están descritas como buena parte de este apartado en un artículo dedicado exclusivamente a patrones (Zapata-Ros, 2011).

Un ejemplo sencillo e inevitable de patrón y de lenguaje es el que exponen DeLano, D.E. y Rising, L. (1997). Consiste en cómo fabricar galletas (cookies) con trozos de chocolate

Otro ejemplo es el de las wikis. De las cuales la más conocida es Wikipedia. Las wikis constituyen el ejemplo más importante de construcciones utilizando lenguajes de patrón, y de hecho cada wiki se ha desarrollado utilizando un patrón concreto: El patrón de las wikis.

El origen de las wikis está en la comunidad de patrones de diseño Portland Pattern, cuyos integrantes, informáticos las utilizaron para escribir patrones de programa de ordenador. La primera wiki llamada WikiWikiWeb fue creada por Ward Cunningham, quien creó y dio nombre al concepto wiki, además implementó el primer servidor WikiWiki, y con él creó el primer servicio de este tipo, para el repositorio de patrones del Portland (Portland Pattern Repository) en 1995.

En el citado artículo (Zapata-Ros, 2011) reproducimos el método que utilizó Ward Cunningham para diseñar la Wiki original como un ejemplo concreto para expresar los lenguajes de patrón de forma efectiva (<http://c2.com/cgi/wiki>).

### **Sinéctica.-**

---

La Sinéctica es un punto de confluencia de las teorías que tratan de explicar y estudian la creatividad, de las técnicas de trabajo en grupo como medio para exteriorizar flujos e impulsos que de otra forma no serían observables y por tanto analizados, mejorados y

compartidos, y los procesos de sistematización y racionalización de esos flujos e impulsos.

Como consecuencia de esta naturaleza y de estos procesos, la Sinéctica también puede considerarse como una teoría para la resolución de problemas.

Así (Gordon,1961) “la Teoría Sinéctica estudia cómo organizar la integración de los diversos individuos que componen un grupo para la resolución de problemas. Es pues una teoría operacional que orientada al uso consciente de los mecanismos psicológicos preconscientes que hay presentes en la actividad creadora humana.”

Reigeluth (2012) **considera la Sinéctica dentro de los Métodos Situados de Instrucción.**

Una de las posibles dimensiones que puede considerarse al estudiar los principios y los métodos de enseñanza, dice, son los diversos niveles de precisión (Reigeluth y Carr Chellman, 2009b). Por ejemplo, en el nivel menos preciso, Merrill (2009) indica que la instrucción debe provenir del entrenamiento. En el extremo opuesto, en un nivel de alta precisión, siguiendo con los ejemplos, "al enseñar un procedimiento, si un alumno se salta un paso durante la ejecución del procedimiento, se debe inducir al alumno hacia la identificación del paso omitido mediante preguntas que lo guíen hasta llegar al reconocimiento de la omisión". De esta manera cuando proporcionamos mayor precisión sobre un principio o sobre un método instruccional, por lo general descubrimos que hace falta que éste sea diferente para diferentes situaciones. Reigeluth (1999a) se refirió a los factores contextuales que influyen en los efectos de los métodos como "escenarios". En definitiva se trata de métodos situados.

Reigeluth y Carr-Chellman (2009a) proponen dos principales tipos de escenarios que requieren conjuntos fundamentalmente diferentes de métodos: Escenarios basados en distintos enfoques de la enseñanza (medios), y escenarios basados en diferentes resultados de aprendizaje (fines). Entre los primeros incluye, entre otros, al juego de rol (role-playing), resolución de conflictos, aprendizaje entre iguales, aprendizaje basado en problemas, aprendizaje por simulación, y también a la sinéctica

Los capítulos en las Unidades de 2 y 3 del libro de Reigeluth y Carr-Chellman (2009c) *Teorías y Modelos de Diseño Instruccional, (Volumen III: Construyendo una base de conocimientos en común)* describen una "base de conocimientos" para esos conjuntos de métodos.

La Teorías Sinécticas tiene su fuente empírica en las historias de casos que ilustran el uso de mecanismos operativos, que de esta forma se llaman *mecanismos sinécticos*, y en cómo operan estos en los procedimientos, que se estudian detalladamente, para la organización y funcionamiento de los grupos, llamados *grupos sinécticos*, fundamentalmente en contextos industriales. En estos procesos se hace especial énfasis en el papel que juega en la actividad creativa la metáfora y en su análisis.

La sinéctica se ha presentado por sus creadores, y así se ha aceptado, como una metodología de resolución de problemas que estimula los procesos de pensamiento de los cuales el sujeto puede no ser consciente.

Este método fue desarrollado por George M. Prince (5 abril 1918 a 9 junio 2009) y William JJ Gordon, originarios de la Arthur D. Little Unidad Invención Diseño en la década de 1950.

Inicialmente el método consistió en grabar en audio y en vídeo reuniones en las que se hablaba sobre experimentos, su desarrollo y el análisis de los resultados, haciendo interpretaciones de ellos. Después el método se completaba con el análisis de las grabaciones. Se discutía sobre formas alternativas de resolución del problema y se procuraba llegar a soluciones de compromiso sobre lo que se consensuaba como una solución creativa.

Esto es muy parecido a lo que hacen los entrenadores de fútbol en las sesiones de preparación de un encuentro, o de análisis del anterior.

Es un tipo de práctica muy útil para muchas cosas y en la que nuestros jóvenes debieran estar instruidos. Además la aceptan bastante bien, siempre que sean dirigidos a encontrar aspectos novedosos o creativos como una vía de éxito en el proceso de resolver el problema.

El nombre sinéctica, viene del griego, a través del inglés synectic, y significa "la unión de elementos diferentes y aparentemente irrelevantes."

La Sinéctica, fundada como una teoría por William JJ Gordon y George M. Prince, adquiere su sentido como una práctica. Los autores la implementan a través de su compañía Synectics, que es la que ejerce la exclusiva para formar a ejecutivos según los principios de esta teoría. Por tanto sinéctica es una teoría y una marca: Es una palabra estándar para describir una forma de encontrar soluciones creativas de problemas en contextos de grupos.

Como teoría la Sinéctica nos ofrece procedimientos para utilizar las habilidades creativas, en la resolución de problemas, de una manera racional. De esta forma (Gordon, 1961) "(...), tradicionalmente, el proceso creativo ha sido considerado después de los hechos. Los estudios sinécticos han intentado investigar el proceso creativo en vivo, mientras que está pasando."

Según Gordon (1961), la investigación sinéctica tiene tres premisas básicas:

- El proceso creativo puede ser descrito y aprendido;
- Los procesos de invención en las artes y en las ciencias tienen una naturaleza análoga y son impulsados por los mismos procesos "psíquicos";
- La creatividad Individual y la creatividad del grupo son análogas.

Hay un elemento clave en la conceptualización de la creatividad que habitualmente se acepta sin discutir y que está considerada como aparentemente irrelevante: Habitualmente se privilegia la emoción sobre el intelecto y lo irracional sobre lo racional. En cambio la teoría sinéctica considera que a través de la comprensión de los elementos emocionales e irracionales de un problema o de una idea, un grupo puede tener más éxito en la solución de un problema.

Otras cuestiones prácticas e ideas que se aportan son:

- La importancia de la conducta creativa en la reducción de las inhibiciones y la liberación de la creatividad inherente en los individuos y la importancia en el proceso de la interacción grupal.



- El estudio de la configuración de grupos y el diseño de prácticas específicas y de estructuras de reuniones que ayudan a las personas a que sus intenciones constructivas son experimentados positivamente por los otros.
- El uso de recursos basados en la creatividad aplicada al comportamiento se extiende, con la aplicación de métodos sinécticos a situaciones que van más allá de las sesiones pensadas para la resolución de problemas en la invención (en particular se manifiesta útil en la resolución constructiva de conflictos).
- Hace hincapié en la importancia de recursos y "procesos metafóricos 'para hacer lo familiar extraño y lo extraño familiar'".

Hay un principio central de proceso como es considerar y animar a considerar "las cosas que habitualmente no se estiman como familiares y objetivizar las cosas en las que habitualmente confiamos." Esto posibilita un análisis donde surgen soluciones que habitualmente no se consideran

Con la sinéctica se inventa una técnica nueva, el "springboarding", para obtener ideas creativas comienzo. Es la "lluvia de ideas", que posteriormente se profundiza y ensancha con el uso de la metáfora; y por último con el análisis y al evaluación de esas ideas. Se destaca la importancia de estos procesos en el desarrollo y eclosión de ideas creativas y sobre todo con el compromiso por apropiación en las personas que van a aplicarlas.

Destaca en todo el proceso la importancia de un personaje dinamizador: El facilitador .

La sinéctica es aplicable igualmente en los procesos de la computación por concurrir en ellos muchos de los elementos que los creadores, impulsores e investigadores encontraron en los procesos donde existe creación. Tienen una naturaleza análoga.

Podemos pues considerar la sinéctica como una componente más del pensamiento computacional

En la tabla 1, al final del artículo, distinguiremos en qué fases de la creación de los códigos interviene la sinéctica, como el resto de las componentes.

### **Metacognición.-**

---

En las tareas de codificación los aspectos procedimentales en cómo afrontar un problema y cómo resolverlo por los alumnos adquieren una importancia clave.

Hace siete años decíamos que, cuando el **concepto de estrategias** se incorpora a la psicología del aprendizaje y a la educación, inevitablemente se ve resaltado el carácter procedimental que tiene todo aprendizaje (Esteban y Zapata-Ros, 2008). Con ello además se está aceptado que los procedimientos utilizados para aprender constituyen una parte muy decisiva del propio aprendizaje y del resultado final de ese proceso. Eso no quiere decir que con anterioridad se ignorase la importancia decisiva de las formas de aprender aportados por el aprendiz, sobre todo por los buenos "maestros". Simplemente no existía una formulación ni una conceptualización tan explícita y con términos específicos sobre las operaciones cognitivas que se ponen en marcha. El concepto de estrategia de aprendizaje es, pues, un concepto que se integra adecuadamente con los principios de la psicología cognitiva, desde la perspectiva constructivista del conocimiento y del aprendizaje. Lo hace además con la importancia

atribuida a los elementos procedimentales en el proceso de construcción de conocimientos y, asimismo, teniendo en cuenta aspectos diferenciales de los individuos. Aspecto tan caro a la psicología cognitiva sobre todo en el caso del aprendizaje en adolescentes, adultos, expertos y novatos.

Conviene pues destacar en primer lugar esta visión del aprendizaje de las habilidades propias del pensamiento computacional.

Hay algunos de los aspectos de las estrategias que pueden resultar más relevantes: El propio concepto de estrategia implica una connotación **finalista e intencional**. Toda estrategia conlleva, de hecho **es**, un **plan de acción para realizar** una tarea que requiera una actividad cognitiva en el aprendizaje. No se trata, por tanto, de la aplicación de una técnica concreta, por ejemplo de aplicar un método de lectura o un algoritmo. Se trata de un plan de actuación que implica habilidades y destrezas –que el individuo ha de poseer previamente- y de una serie de técnicas que se aplican en función de las tareas a desarrollar, sobre las que el alumno decide y sobre las que tiene una intención de utilizar consciente. Por tanto lo más importante de esta consideración es que para que haya intencionalidad **ha de existir conciencia** de:

a) **la situación** sobre la que se ha de operar (problema a resolver, datos a analizar, conceptos a relacionar, información a retener, etc.). Esta conciencia y esta intencionalidad presupone, como una cuestión clave desde el punto de vista del aprendizaje, **la representación de la tarea** que se realiza, sobre la que el aprendiz toma la decisión de qué estrategias va a aplicar; y

b) **de los propios recursos** con que el aprendiz cuenta, es decir, de sus habilidades, capacidades, destrezas, recursos y de la capacidad de generar otros nuevos o mediante la asociación o reestructuración de otros preexistentes.

En todos estos puntos, decisiones y representaciones, ha de existir en definitiva la conciencia de los propios recursos cognitivos con que cuenta el aprendiz. Eso es lo que se ha denominado **metacognición**.

Así pues no es sólo una estrategia o un conjunto de estrategias. Es la condición necesaria para que pueda darse cualquier plan estratégico. Lo contrario serían simplemente algoritmos o incluso estrategias pero donde, al no haber intencionalidad, no habría la valoración que conlleva la adopción de un plan con previa deliberación de la situación y de los recursos.

La metacognición y el estudio de los estilos de aprendizaje son dos cosas que van íntimamente ligadas.

Los psicólogos del aprendizaje descubrieron que los alumnos tenían distintas estructuras cognitivas que afectaban a sus formas personales de aprender, a los procedimientos que cada individuo ponía en marcha de forma espontánea cuando intentaba aprender algo, que determinaba, en definitiva, las estrategias de aprendizaje de los alumnos. Descubrieron además que estos procedimientos constituían racimos (*clusters*) que se repetían con frecuencia de forma análoga o parecida en distintos individuos. Estas estructuras tenían un origen en el que no entraron en polémica, se habían formado a lo largo de su vida en función de distintos factores ambientales, genéticos, culturales... y de experiencias personales, que se aceptaba sin más. Formaba parte de la identidad, de la configuración cognitiva, por así decirlo, del alumno. Les llamaron estilos de aprendizaje.

Y se han definido de forma consistente con esta idea

Un estilo de aprendizaje es una forma consistente que tiene un estudiante de responder a los estímulos que se producen en un contexto de aprendizaje (Clark, 2014).

Keefe (1979) define los estilos de aprendizaje como la

"combinación de características cognitivas, afectivas, y de factores fisiológicos que sirven como indicadores relativamente estables de cómo un aprendiz percibe, interactúa con él, y responde al ambiente de aprendizaje."

Stewart y Felicetti (1992) definen los estilos de aprendizaje como lo que determina las

"condiciones educativas en las que un estudiante es más probable que aprenda."

En todos los casos los autores ponen énfasis en que los estudios de los estilos de aprendizaje no están realmente preocupados por *lo que* los alumnos aprenden, sino más bien por *la forma en que* prefieren aprender. Como decíamos anteriormente, hay un factor de intencionalidad.

El paso siguiente fue intentar determinar de manera eminentemente mecanicista qué formas de organizar la instrucción era la más eficiente para cada uno de estos estilos, al menos para los más frecuentes, con el propósito de mejorar los aprendizajes. Lo cual constituye por otro lado el principal objetivo de la calidad de la enseñanza. En ese planteamiento se consideró a los estudios y a los resultados de las investigaciones sobre estilos de aprendizaje como uno de los principales conjuntos de elementos que iluminaban, informaban y fundamentaban el diseño instruccional.

En nuestro ámbito de trabajo, y en esa hipótesis tendría sentido plantearse qué estilos de aprendizaje son los más propicios para el pensamiento computacional o van ligados a él, y adaptar el diseño instruccional a esos estilos de aprendizaje, y a los alumnos que presentan estos perfiles de aprendizaje. Afortunadamente las investigaciones han ido por otro lado y han puesto en evidencia que ésta era una idea equivocada.

La expresión "afortunadamente" se justifica cuando supusiéramos, por ejemplo, que ciertos estilos de aprendizaje fuesen asociados a estereotipos humanos. Pensemos que hay estilos vinculados con el género (por ejemplo estilos de aprendizaje femeninos) o con componentes étnicas, o de clase social. Esto nos llevaría a distintos tipos de exclusiones.

No obstante los estilos de aprendizaje se han manifestado como de utilidad en otro sentido: son los puntos a lo largo de una escala que nos ayudan a descubrir las diferentes formas de representaciones mentales. Sin embargo, no son buenas como caracterizaciones de lo que los alumnos son o de cómo son. Las conclusiones de los estudios como veremos, no deben llevarnos a dividir a la población en un conjunto de categorías estancas con respecto a cómo organizar la educación o a elaborar las estrategias educativas (como sucede por ejemplo en casos estereotipados: educación para "individuos visuales", educación para "individuos auditivos", etc. De esta forma se ha intentado asignar a las personas un punto en un continuo, similar a la medición de la altura o peso).

En otras palabras: se trata de no encasillar a los alumnos, y de dejar claro que todos somos capaces de aprender a partir de casi cualquier estilo de aprendizaje, no importa

cuales sean nuestras preferencia sobre como representar el conocimiento, o de cómo aprender, de forma inicial.

Finalmente se demostró con evidencias empíricas que adaptando el diseño instruccional a los distintos estilos de aprendizaje o teniéndolo como referencia de alguna forma para organizar la enseñanza no se obtenían mejores resultados (Marzano, 1998) (Coffield, et. al., 2004).

La literatura especializada básicamente indica que hay un amplio consenso y aceptación del concepto de estilos de aprendizaje. Incluso hay un estudio que demuestra que existen los estilos de aprendizaje (Thompson-Schill, Kraemer, Rosenberg, 2009). Sin embargo, **no hay acuerdo sobre cuáles son los mejores estilos de aprendizaje en función de los objetivos de aprendizaje deseados para cada caso, ni de la forma de establecer un criterio de eficiencia** (Coffield, Moseley, Hall, Ecclestone, 2004). Así mientras que los especialistas han reconocido desde hace tiempo la necesidad de actividades de enseñanza innovadoras que se relacionen con los diversos estilos de aprendizaje de los alumnos (lo cual los haría útiles en la mentorización), hay dudas razonables en cuanto a que sean significativos y en cómo lo sean a la hora de determinar el ambiente de aprendizaje.

Es decir, la mayoría de los investigadores coinciden en que los alumnos tienen diferentes estilos de aprendizaje, sin embargo, la investigación manifiesta un claro acuerdo en que es relativamente poco importante en el diseño de programas de aprendizaje. Es mucho más importante para este fin temas tales como la naturaleza de los contenidos, de las actividades, la significatividad general, la relevancia, la complejidad de la tarea, etc. y utilizar estrategias y contextos adecuados que emparejar métodos de enseñanza con preferencias o estilos individuales (Coffield, et. al., 2004).

Como ejemplos de estudios empíricos se pueden citar:

En un gran meta-estudio realizado por Marzano (1998) se encontró que ciertas representaciones de los contenidos tuvieron efectos positivos en los resultados del aprendizaje, independientemente de las modalidades con que los alumnos aprender, de sus preferencia o del estilo de aprendizaje.

También se cita el estudio de Constantinidou y Baker (2002) donde se demostró que la presentación visual mediante el uso de imágenes adecuadas era ventajoso para todos los adultos, independientemente de su estilo de aprendizaje. Incluso en el caso de aquellos con una fuerte preferencia por el procesamiento verbal.

Sin embargo, como hemos dicho, eso no significa que los estilos de aprendizaje no sean importantes. Como escribió Coffield (Coffield, et. al., 2004): "La unidad de consideración y de uso didáctico de los estilos de aprendizaje debe ser el individuo más que el grupo".

Por lo tanto, aquellos que son responsables de ayudar a otros a aprender, tales como mentores, instructores o entrenadores debieran atender a los estilos y ajustarse en algunos casos a ellos, mientras que los que diseñan la instrucción o enseñan a grupos, debieran ver los estilos de aprendizaje con relativa importancia.

Y aquí es donde entra la metacognición. Si los alumnos son conscientes, primero, de su propio estilo de aprendizaje, fuesen conscientes, segundo, de la necesidad del cambio de los procedimientos que constituyen su estilo o de consolidarlos y potenciarlos, y por último de sus propias capacidades para llevar de forma autónoma ese cambio o de la necesidad de adquirirlas (capacidades metacognitivas) estaríamos en presencia de la

cuestión clave para abordar el resto de competencias del pensamiento computacional para la mayor parte de los alumnos.

El argumento para señalar la importancia de la metacognición, su papel clave, en palabras de David Merrill (2000) es que la mayoría de los estudiantes no son conscientes de sus estilos de aprendizaje y si se deja a sus propios medios, no es probable que empiecen a aprender de nuevas maneras. Por lo tanto, el conocimiento de los estilos de aprendizaje de uno mismo puede ser utilizado para aumentar la autoconciencia acerca de las fortalezas y debilidades como aprendices que cada uno tiene y por consiguiente para mejorar en el aprendizaje.

Si bien todas las ventajas que se atribuyen a la metacognición (ser consciente de los propios procesos de pensamiento y aprendizaje) pueden ser adquiridas alentando a los estudiantes a adquirir conocimientos acerca de su propio aprendizaje y el de los demás (Coffield, et. Al., 2004), lo importante es estudiar e investigar cómo los alumnos pueden adquirir este conocimientos, formar en habilidades cognitivas.

En el caso del pensamiento computacional la cuestión es cómo los estudiantes pueden adquirir las habilidades metacognitivas específicas, cuáles son las mejores estrategias y cómo pueden detectar cuales son las debilidades y las fortalezas de sus propios estilos y cambiarlas o potenciarlas.

Establecer en qué medida es posible formar en estas habilidades y cómo llevar a cabo este meta-aprendizaje.

Con este apartado dedicado a metacognición concluimos la serie de los catorce elementos/componentes del pensamiento computacional reseñados en la fig. 2

Como dijimos al principio queda por desarrollar pormenorizadamente los contenidos en un corpus útil a las distintas modalidades y niveles de formación, así como para la formación de maestros y profesores que los impartan.

### **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. Ahora queremos estudiarlo desde el punto de vista del pensamiento computacional: qué elementos de este pensamiento, de los que hemos visto en las antradas anteriores, están presentes en cada una de estas fases.

La propuesta cuales son las fases diferenciadas en el proceso de creación de un código es

- Detección y delimitación del problema y de su naturaleza
- Delimitación de métodos y disciplinas en la resolución del problema
- Organización de la resolución, *feed back* e investigación formativa
- Diseño de la resolución
- Algoritmia/ diagrama de flujo.- Incluye la discusión
- Elaboración del código (programa).- Incluye codificación, ejecuciones e implementación, documentación, etiquetas, modularización
- Prueba/ Validación .- Incluye implementación y depuraciones de errores,

Luego, además, podemos considerar un conjunto de competencias necesarias en todas las fases, sin que tengan que ver directamente y de forma exclusiva con algunas de ellas, como puede ser la metacognición.

La segunda cuestión es qué componentes están presentes en las distintas fases o en todas.

La propuesta que se presenta es (tabla 1):

	Competencias necesarias en todas las fases , sin que tengan que ver directamente y de forma exclusiva con algunas de ellas	Detección y delimitación del problema y de su naturaleza	Delimitación de métodos y disciplinas en la resolución del problema	Organización de la resolución, <i>feed back</i> e investigación formativa	Diseño de la resolución	Algoritmia/ diagrama de flujo	Elaboración del código (programa)	Validación
Metacognición	X							
Sinéctica		X	X	X	X			X
Análisis descendente				X		X	X	
Análisis ascendente						X	X	
Recursividad				X		X	X	
Método por aprox. sucesiv. Ensayo – error						X	X	X
Heurística					X	X	X	
Iteración						X	X	
Pensamiento divergente					X			
Creatividad		X	X	X	X			
Resolución de problemas		X	X	X	X	X	X	X
Pensamiento abstracto	X							
Métodos colaborativos		X	X	X	X			
Patrones	X							

Tabla 1

## Discusión.-

Falta por determinar con evidencias empíricas si, como parece, la codificación es una competencia compleja o un conjunto de competencias, así como establecer en términos diferenciados cuáles son esas competencias y determinar el diseño y los términos de las investigaciones que diesen lugar a estas delimitaciones.

Faltaría pues definir qué es codificación en un sentido pluridisciplinar que implique a profesionales de la psicología del Aprendizaje y del Desarrollo, los especialistas en Educación (Pedagogía del pensamiento computacional, curriculum, etc)

*Code*, o codificación como lo hemos traducido, o *programming code* (programación de códigos) consiste en elaborar códigos fuente de programas de ordenador que puedan ser interpretados y/o compilados por un interface para decirle a un sistema informático cómo se resuelve un problema o cómo se realiza un procedimiento de forma eficaz.

En los documentos utilizados para elaborar este trabajo se ha definido (Balanskat y Engelhardt, October 2014 p. 5) como

**una competencia clave** que tendrá que ser adquirida por todos los jóvenes estudiantes y cada vez más por los trabajadores en una amplia gama de actividades industriales y profesiones. La codificación es parte del razonamiento lógico y representa **una de las habilidades** clave que forma parte de lo que ahora se llaman “habilidades del siglo 21”.

No obstante en el informe citado, donde se ponen énfasis en esta necesidad (de hecho es el documento base para la integración de las enseñanzas para la adquisición de las competencias para la codificación) como prioridad de la UE:

- a) No se plantea como una idea de un curriculum integral y sistémico que abarque desde las etapas preescolares hasta la educación universitaria.
- b) Se dedica a describir las experiencias y el estado de la cuestión en los países europeos, donde solo se constatan situaciones de inclusión en otras materias o de materias específicas de programación del tipo que hemos señalado.

Sin embargo en este trabajo hemos puesto de relieve que la codificación es una competencia compleja o más bien un complejo de habilidades de las que participan posiblemente, entre otras, las 14 que hemos glosado, y al que en conjunto es lo que llamamos pensamiento computacional, de manera que su adquisición quedaría incompleta si faltase alguno de estos elementos.

Por último quedaría analizar además cuáles serían los pasos siguientes para determinar el curriculum y las características de la formación de los profesores y maestros.

Presentación del artículo: 15 de julio de 2015  
Fecha de aprobación: 5 de septiembre de 2015  
Fecha de publicación: 15 de septiembre de 2015

Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. <i>RED. Revista de Educación a Distancia</i> , 46(4). 15 de Septiembre de 2015. Consultado el (dd/mm/aa) en <a href="http://www.um.es/ead/red/46">http://www.um.es/ead/red/46</a>
---



## Referencias.-

- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A Pattern Language: Towns, Buildings, Construction* (Center for Environmental Structure).
- Alexander et al., 1977, *A Pattern Language*, Oxford University Press, px.
- Ausubel, D. P. (1963). *The psychology of meaningful verbal learning; an introduction to school learning*. New York: Grune & Stratton.
- ANDERSON, J.A. (1993) *Rules of the Mind* (Hillsdale, NJ, Erlbaum).
- Baker, M., Hansen, T., Joiner, R., & Traum, D. (1999). The role of grounding in collaborative learning tasks. In P. Dillenbourg (Ed.), *Collaborative Learning: Cognitive and Computational Approaches*. (pp. 31-63; 223-225). Elsevier Science.<http://www.uio.no/studier/emner/matnat/ifi/TOOL5100/v08/leseliste/F9/baker99role.pdf>
- Balanskat, A. & Engelhardt, K. (October, 2014). *Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe*. European Schoolnet (EUN Partnership AISBL) [http://www.eun.org/c/document\\_library/get\\_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887](http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887)
- Bawden, D. (2001). Information and digital literacies: a review of concepts. *Journal of Documentation*, 57(2), 218–259.
- Bawden, D. (2008). Origins and concepts of digital literacy. *Digital literacies: Concepts, policies and practices*, 17-32. <http://sites.google.com/site/colinlankshear/DigitalLiteracies.pdf#page=19>
- Bergin, J. (2008) *Pedagogical Patterns Project* [en línea]. Disponible en: <http://www.pedagogicalpatterns.org/>
- Blikstein, (2013). *Seymour Papert's Legacy: Thinking About Learning, and Learning About Thinking*. <https://tltl.stanford.edu/content/seymour-papert-s-legacy-thinking-about-learning-and-learning-about-thinking>
- Bono, E. D. (1968). *New think: the use of lateral thinking in the generation of new ideas*. Basic Books.
- Bono, E. D. (1970). *Lateral Thinking. A Textbook of Creativity*. Londres: Ward Lock Educational.
- Bono, E. DE (1986):*El pensamiento lateral: manual de creatividad*. Editorial Paidós.
- Bloom, B.S. (1984). *The 2 Sigma Problem: The Search for Methods of Group Instruction as effective as One-to-One Tutoring*, Educational Researcher, 13:6 (4-16). <http://www.comp.dit.ie/dgordon/Courses/ILT/ILT0004/TheTwoSigmaProblem.pdf>

- Clark, D. (2014) *Learning Styles & Preferences*.  
<http://www.nwlink.com/~donclark/hrd/styles.html>
- Coffield, F., Moseley, D., Hall, E., Ecclestone, K. (2004). *Learning Styles and Pedagogy in Post-16 Learning: A systematic and critical review*.  
www.LSRC.ac.uk: Learning and Skills Research Centre. Retrieved  
from: <http://www.lsd.org.uk/files/PDF/1543.pdf>
- Constantinidou, F., Baker, S. (2002). Stimulus modality and verbal learning performance in normal aging. *Brain and Language*, 82(3), 296-311.
- Corballis, M. C. (2007). Pensamiento recursivo. *Mente y cerebro*, 27, 78-87.  
<http://amscimag.sigmaxi.org/4Lane/ForeignPDF/2007-05CorballisSpanish.pdf>
- Corballis, M. C. (2014). *The recursive mind: The origins of human language, thought, and civilization*. Princeton University Press. <http://press.princeton.edu/titles/9424.html>
- Csikszentmihalyi, M. (1996). *Creativity: Flow and the psychology of discovery and invention*.
- Csikszentmihalyi, M. (2009). *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Collins.
- Csikszentmihalyi, M. (1998). *Creatividad: el flujo y la psicología del descubrimiento y la invención*. Ed. Paidós.
- DeLano, D.E. y Rising, L. (1997). Introducing Technology into the Workplace. *Proceedings PLoP'97 Conference*. Consultado en  
<http://hillside.net/plop/plop97/Proceedings/delano.pdf>
- Dillenbourg, P. (1999). What do you mean by collaborative learning?. *Collaborative-learning: Cognitive and Computational Approaches.*, 1-19. <https://halshs.archives-ouvertes.fr/file/index/docid/190240/filename/Dillenbourg-Pierre-1999.pdf>
- Eggleston, J. (1982). *Sociología del currículum*. Ed. Troquel. Buenos Aires.
- Esteban, M. y Zapata, M. (2008, Enero). Estrategias de aprendizaje y eLearning. Un apunte para la fundamentación del diseño educativo en los entornos virtuales de aprendizaje. Consideraciones para la reflexión y el debate. Introducción al estudio de las estrategias y estilos de aprendizaje. *RED. Revista de Educación a Distancia*, número 19. Consultado (día/mes/año) en <http://www.um.es/ead/red/19>
- Eshet, Y. (2002). Digital literacy: A new terminology framework and its application to the design of meaningful technology-based learning environments, In P. Barker and S. Rebelsky (Eds.), *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 493-498 Chesapeake VA: ACE, Retrieved November 30, 2007, from  
<http://infosoc.haifa.ac.il/DigitalLiteracyEshet.doc>

- Eshet-Alkalai, Y. (2004), Digital literacy: a conceptual framework for survival skills in the digital era, *Journal of Educational Multimedia and Hypermedia*, 139(1), 93–106. Available at: [http://www.openu.ac.il/Personal\\_sites/download/Digital-literacy2004-JEMH.pdf](http://www.openu.ac.il/Personal_sites/download/Digital-literacy2004-JEMH.pdf)
- Fitch, T., Hauser, M. & Chomsky, N. 2005. The evolution of the language faculty: Clarifications and implications. *Cognition*. 97.179-210
- Fricke, A. y Voelter, M. (2000). SEMINARS: A Pedagogical Pattern Language about teaching seminars, [en línea]. *Proceeding EuroPLoP 2000*. Disponible en: <http://www.voelter.de/publications/seminars.html> [2008, 2 diciembre].
- Gilster, P. (1997). *Digital literacy*. New York: Wiley.
- Gordon, W. J. (1961). *Synectics: The development of creative capacity*.
- Hauser, M., Chomsky, N., & Fitch, T. 2002. The faculty of language: what is it, who has it, and how did it evolve. *Science* 198. 1569-79
- Hackenberg, A. J. (2007). Units coordination and the construction of improper fractions: A revision of the splitting hypothesis. *Journal of Mathematical Behavior*, 26(1), 27–47.
- Hansen, T., Dirckinck-Holmfeld, L., Lewis, R., & Rugelj, J. (1999). Using telematics to support collaborative knowledge construction. *Collaborative learning: Cognitive and computational approaches*, 169-196. [http://www.researchgate.net/publication/228559912\\_Using\\_telematics\\_to\\_support\\_collaborative\\_knowledge\\_construction/file/60b7d523962ffc2db3.pdf](http://www.researchgate.net/publication/228559912_Using_telematics_to_support_collaborative_knowledge_construction/file/60b7d523962ffc2db3.pdf)
- Hayman-Abello SE, Warriner EM (2002). (2002). Child clinical/pediatric neuropsychology: some recent advances. *Annual review of psychology*, 53(1), 309-339.
- Himanen, P. (2002). *La ética del hacker y el espíritu de la era de la información*. <http://eprints.rclis.org/12851/>
- Jonassen, D., Davidson, M., Collins, M., Campbell, J., & Haag, B. B. (1995). Constructivism and computer-mediated communication in distance education. *American journal of distance education*, 9(2), 7-26. <http://www.tandfonline.com/doi/abs/10.1080/08923649509526885>
- Keefe, J.W. (1979) Learning style: An overview. NASSP's *Student learning styles: Diagnosing and proscribing programs* (pp. 1-17). Reston, VA. National Association of Secondary School Principals..
- Koch, T., & Denike, K. (2009). Crediting his critics' concerns: Remaking John Snow's map of Broad Street cholera, 1854. *Social science & medicine*, 69(8), 1246-1251. <http://www.albany.edu/faculty/fboscoe/papers/koch2009.pdf>
- Lanham, R.A. (1995). Digital literacy, *Scientific American*, 273(3), 160–161.

- Lankshear, C. and Knobel, M. (2006). Digital literacies: policy, pedagogy and research considerations for education. *Digital Kompetanse: Nordic Journal of Digital Literacy*, 1(1), 12–24.
- Leeder, D., Boyle, T., Morales, R., Wharrad, H., & Garrud, P. (2004). To boldly GLO-towards the next generation of Learning Objects. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Vol. 2004, No. 1, pp. 28-33).
- Mack, N. K. (2001). Building on informal knowledge through instruction in a complex content domain: Partitioning, units, and understanding multiplication of fractions. *Journal for Research in Mathematics Education*, 32(3), 267–296.
- Mandelbrot, B. (1982). *The fractal geometry of nature*. W. H. Freeman.
- Mandelbrot, B. (1977). *Fractals, form, chance and dimension*. W. H. Freeman.
- Marzano, R.J. (1998). *A theory-based meta-analysis of research on instruction*. Mid-continent Regional Educational Laboratory, Aurora, CO.
- Merrill, D. (2000). Instructional Strategies and Learning Styles: Which takes Precedence? *Trends and Issues in Instructional Technology*, R. Reiser and J. Dempsey (Eds.). Prentice Hall.
- Merrill, M. D. (2009). First principles of instruction. In C. M. Reigeluth & A. A. Carr-Chellman (Eds.), *Instructional-design theories and models: Building a common knowledge base* (Vol. III, pp. 41-56). New York: Routledge.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc. <http://www.arvindguptatoys.com/arvindgupta/mindstorms.pdf>
- Pólya, George (1945). *How to Solve It*. Princeton University Press.
- Pólya, G. (1989). *Como plantear y resolver problemas* Ed. Trillas. (Primera edición 1965)
- Popper, Karl (1934). «*The Logic of Scientific Discovery*». Consultado el 08-09-2007. <http://books.google.es/books?hl=es&lr=&id=LWSBAgAAQBAJ&oi=fnd&pg=PP1&dq=popper+scientific+methods>
- Popper, Karl (1934). *La lógica de la investigación científica*. Traducido por Víctor Sánchez de Zavala (1ª edición). Madrid: Editorial Tecnos (publicado el 1962). ISBN 84-309-0711-4.
- Popper, Karl (1934). *The Logic of Scientific Discovery*. New York: Routledge (publicado el 2009).
- Olive, J., & Vomvoridi, E. (2006). Making sense of instruction on fractions when a student lacks necessary fractional schemes: The case of Tim. *Journal of Mathematical Behavior* 25(1), 18–45.

- Raja, T. (2014). *We can code it!*.  
<http://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education>.
- Reigeluth, C. M. (2012). Instructional theory and technology for the new paradigm of education. *RED, Revista de Educación a Distancia*, 32, 1-18.  
<http://www.um.es/ead/red/32/reigeluth.pdf>.
- Reigeluth, C. M., & Carr-Chellman, A. A. (2009a). Situational principles of instruction. In C. M. Reigeluth & A. A. Carr-Chellman (Eds.), *Instructional-design theories and models: Building a common knowledge base* (Vol. III, pp. 57-68). New York: Routledge.
- Rosas, M. J. M. (2012). Recensión de “The recursive mind. The origins of human language, thought, and civilization”, de Michael C. Corballis. *Teorema: Revista internacional de filosofía*, 31(1), 151-154. <http://dialnet.unirioja.es/descarga/articulo/4349918.pdf>
- Salomon, G. (1993). *Distributed cognitions. Psychological and educational considerations* (pp. 111-138) Cambridge, USA: Cambridge University Press.
- Siemens, G. (December 12, 2004). *Connectivism: A Learning Theory for the Digital Age* Consultado el 18/8/2011 en <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.3793&rep=rep1&type=pdf> el 30/08/2012
- Steffe, L. P., & Olive, J. (2010). *Children's fractional knowledge*. Springer: New York.
- Steffe, L. P. (2004). On the construction of learning trajectories of children: The case of commensurate fractions. *Mathematical Thinking and Learning*, 6(2), 129–162
- Steffen, J. H. (2008). Optimal boarding method for airline passengers. *Journal of Air Transport Management*, 14(3), 146-150. <http://www.sciencedirect.com/science/article/pii/S0969699708000239>
- Stewart, K.L., Felicetti, L.A. (1992). Learning styles of marketing majors. *Educational Research Quarterly*, 15(2), 15-23.
- Thompson-Schill, S., Kraemer, D., Rosenberg, L. (2009). *Visual Learners Convert Words To Pictures In The Brain And Vice Versa, Says Psychology Study*. University of Pennsylvania. News article retrieved from <http://www.upenn.edu/pennnews/news/visual-learners-convert-words-pictures-brain-and-vice-versa-says-penn-psychology-study>
- Valverde-Berrocoso, J., Fernández-Sánchez, M.R., Garrido-Arroyo, M.C. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *RED, Revista de Educación a Distancia*. Número 46. Número monográfico sobre «Pensamiento Computacional». Septiembre de 2015. Consultado el (dd/mm/aa) en <http://www.um.es/ead/red/46>

- Washington, US Congress of Technology Assessment, OTA CIT-235 (April 1984). *Computerized Manufacturing Automation: Employment, Education and the Workplace*, page 234. <http://ota-cdn.fas.org/reports/8408.pdf>
- Wilkins, J. L. M., & Norton, A. (2011). The splitting loope. *Journal for Research in Mathematics Education*, 42(4), 386–416
- Wilkins, J. L., Norton, A., & Boyce, S. J. (2013). Validating a Written Instrument for Assessing Students' Fractions Schemes and Operations. *Mathematics Educator*, 22(2), 31-54.
- Wing, J.M. (March 2006). Computational Thinking. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. *COMMUNICATIONS OF THE ACM* /Vol. 49, No. 3. <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Wood, D., & Wood, H. (1996). Vygotsky, tutoring and learning. *Oxford review of Education*, 22(1), 5-16. <http://www.jstor.org/stable/1050800>
- Wood, D., & Wood, H. (1996). Vygotsky, tutoring and learning. *Oxford review of Education*, 22(1), 5-16. [http://www.tandfonline.com/doi/abs/10.1080/0305498960220101#.VI3EvyuG-\\_1](http://www.tandfonline.com/doi/abs/10.1080/0305498960220101#.VI3EvyuG-_1)
- Zapata-Ros, M. (1996a). *Integración de la GEOMETRÍA FRACTAL en las Matemáticas, y en la Informática, de Secundaria*. [http://platea.pntic.mec.es/~mzapata/tutor\\_ma/fractal/fracuned.htm# Pero... ¿qué son los fractales?](http://platea.pntic.mec.es/~mzapata/tutor_ma/fractal/fracuned.htm# Pero... ¿qué son los fractales?)
- Zapata-Ros, M. et al (1996b). Integración de la GEOMETRÍA FRACTAL en las Matemáticas, y en la Informática, de Secundaria. *Materiales para la Enseñanza Secundaria: área de Matemáticas y área de Educación Física. Documentos CEP*. Núm. 47. CEP Murcia II. <http://hdl.handle.net/11162/645>.
- Zapata-Ros, M. (2009): Objetos de aprendizaje generativos, competencias individuales, agrupamientos de competencias y adaptatividad . *RED. Revista de Educación a Distancia*, número monográfico X. Consultado (DD/MM/AA) en <http://www.um.es/ead/red/M10>. Pág. 5.
- Zapata-Ros, M. (2011). Patrones en elearning. Elementos y referencias para la formación. 15 de julio de 2011. *RED, Revista de Educación a Distancia. Número 27*. Consultado el [dd/mm/aaaa] en <http://www.um.es/ead/red/27/>
- Zapata-Ros, M. (2013a). ¿Por qué nos gustan las cosas hermosas? La belleza está escrita en lenguaje matemático mucho antes de que se descubra. *Blog Redes Abiertas*. <http://redesabiertas.blogspot.com.es/2013/03/por-que-nos-gustan-las-cosas-hermosas.html>
- Zapata-Ros, M. (2013b). *El “problema de 2 sigma” y el aprendizaje ayudado por la tecnología en la Educación Universitaria*. <http://red.hypotheses.org/287>
- Zapata-Ros, M. (2014). La fundamentación teórica y científica del conectivismo. *RED-Hypotheses*. <http://red.hypotheses.org/688>

Zapata-Ros, M. (2014a). Enseñanza Universitaria en línea: MOOC, aprendizaje divergente y creatividad (II).RED-Hypotheses. <http://red.hypotheses.org/416>

Zapata-Ros, M. (2014b). Enseñanza Universitaria en línea: MOOC, aprendizaje divergente y creatividad (II).RED-Hypotheses. <http://red.hypotheses.org/427>