

Sviluppare il pensiero computazionale e la creatività nella scuola primaria

Un'analisi sistematica della letteratura

Francesco C. Ugolini, Università degli Studi Guglielmo Marconi
Panagiotis Kakavas, Università degli Studi Guglielmo Marconi

SOMMARIO

Since 2006, the year in which Jeannette Wing published her well-known seminal article, interest in developing programs of Computational Thinking in primary school has progressively grown. This concept does not rely exclusively on *coding* activities; it rather deals with problem solving activities, and therefore regarding creative thought processes. Furthermore, it can also be seen as a way to express oneself creatively. As computational thinking and creativity overlap in several ways, this paper aims to highlight teaching practices and learning frameworks used to support the development of both computational and creative thinking skills in primary level education students, by analyzing four scientific papers coming from a larger systematic review about computational thinking in primary school. Though to a different extent, depending of the specific methodology adopted, the papers show how the development of computational thinking enhances creativity as well, providing indications for integrated activities in primary school curricula.

KEYWORDS: *Coding, computational thinking, creativity, problem solving, systematic review*

Pensiero computazionale: definizioni e intrecci con la creatività¹

Anche se il concetto di pensiero computazionale non è nuovo – trae origine addirittura dagli anni '50 quando si parlava di pensiero algoritmico (Pérez-Marín et al., 2018) e il suo sviluppo presso un pubblico di bambini è stato promosso nei primi anni '80 da Seymour Papert (1980) – è però dal 2006, anno in cui l'informatica Jeannette Wing ha rilanciato la sua importanza, che si è imposto al centro del dibattito educativo sia scientifico sia politico². Spesso, tuttavia, tale

espressione è stata associata in modo quasi esclusivo al *coding*, ossia all'attività vera e propria di programmazione seppur con linguaggi pensati in modo specifico per gli alunni della scuola primaria³. A quale esigenza risponde dunque l'introduzione di simili attività fin dalla scuola primaria?

Non vi è dubbio che tra le righe delle politiche si risponda a un paradigma funzionalistico (Dufva, Dufva, 2016), volto, in prospettiva, a promuovere competenze digitali spendibili sul mercato del lavoro (Commissione Europea, 2016). Tuttavia ciò troverebbe scarsa giustificazione, sul piano pedagogico, per un simile dispendio di risorse. Per tale motivo, occorre definire meglio i contorni di quella che viene considerata soprattutto un'abilità di pensiero al di là della sua concretizzazione tecnologica strettamente intesa.

È innanzi tutto opportuno leggere con attenzione il contributo originario di Jeannette Wing (2006) per chiarire le intenzioni della scienziata. Se è vero che alcuni passaggi appaiono oggi ingenui – il riferimento alla “comprensione del comportamento umano” grazie ai concetti dell'informatica è stato abbondantemente smentito dalle teorie legate all'*embodied cognition* e alla *situatedness* (Varela, 1990; Gallese, Lakoff, 2005; Berthoz, 2009; Rivoltella, 2012; Sibilio, 2012; Rivoltella, Rossi, 2019) – altri costituiscono i mattoni fondamentali di un'accezione più appropriata di pensiero computazionale, su tutti il principio per il quale si tratta del modo di ragionare degli informatici e non dei computer. «Si riferisce a concettualizzare e non a programmare. Pensare come un informatico significa molto più che esser capaci a programmare il computer e richiede soprattutto di saper pensare a livelli multipli di astrazione. Si riferisce ad abilità fondamentali, non a capacità meccaniche di basso livello. Non è un tentativo di fare in modo che gli uomini pensino come i computer; i computer sono inintelligenti e noiosi mentre gli uomini hanno intelligenza e fantasia» (Wing, 2006)⁴.

Più in generale, il pensiero computazionale viene associato alle abilità di *problem solving*. Barr e Stephenson lo definiscono come «un approccio alla risoluzione di problemi in un modo che possa essere implementato con un computer» (Barr, Stephenson, 2011, p. 115). In questo modo, coinvolge abilità di pensiero quali la decomposizione, il riconoscimento di modelli, l'astrazione e la progettazione algoritmica (Anderson, 2016).

Un discorso a sé merita l'abilità di *testing and debugging*, associata in molte definizioni operative (Brennan, Resnick, 2012; Grover,

Pea, 2012; Anderson, 2016) al pensiero computazionale. Se, infatti, decomposizione, riconoscimento di modelli, astrazione e progettazione algoritmica, in qualche modo “precedono” logicamente l’attività vera e propria di *coding* attenendo piuttosto all’ambito dell’analisi e della progettazione del software, il *testing* e il *debugging* la seguono, o, meglio, sono integrate con essa Anderson (2016) le riconduce all’ambito della valutazione. Trincherò (2019) sottolinea l’importanza di questo aspetto in chiave autoregolativa: riprendendo il modello Risorse-Interpretazione-aZione-Autoregolazione (Trincherò, 2012). Il pensiero computazionale include tre momenti, tra di loro interrelati: uno di concettualizzazione, ossia di comprensione, astrazione, formulazione, riformulazione di problemi e soluzioni; uno di codifica delle concettualizzazioni prodotte in un linguaggio formalizzato intellegibile da un esecutore, non necessariamente informatizzato; uno di esecuzione e valutazione dei codici prodotti che rende evidenti gli effetti delle concettualizzazioni prodotte. I tre momenti sono interrelati perché le possibilità e i modelli di codifica impattano sulle concettualizzazioni producibili e sulle forme in cui i risultati di queste saranno visibili (Trincherò, 2019, p. 81).

L’intreccio tra pensiero computazionale e attività di *problem solving* interessa il tema della presente pubblicazione in quanto quest’ultimo «non riguarda problemi di routine risolvibili con procedure già conosciute e automatizzate ma problemi inediti in cui è necessario mobilitare *in modo creativo e originale* gli strumenti concettuali che si hanno a disposizione» (Trincherò, 2019, p.82, corsivo nostro).

L’aspetto creativo del pensiero computazionale emerge anche in un altro filone, ossia quello che vede la programmazione come un vero e proprio atto creativo (Verborgh, 2013, cit. da Olimpo, 2017, p. 22). «Computing is a creative human activity» (Grover, Pea, 2012, p. 2) e si colloca in quella prospettiva che vede il pensiero computazionale come opportunità per l’espressione creativa di sé (Brennan, Resnick, 2012, p. 10). Si tratta di un approccio che, nell’analisi sul significato sociale del “codice” operata da Dufva e Dufva (2016), segue un altro paradigma, quello “postmoderno”: secondo gli autori, «il *coding* creativo consente agli artisti di interrogare e criticare il codice e, nel contempo, di esprimersi tramite esso» (Trincherò, 2019, p.105). Analizzare come nelle ricerche empiriche internazionali viene trattato lo sviluppo congiunto di pensiero computazionale e creatività ci pare dunque viepiù significativo.

Analisi sistematica della letteratura su pensiero computazionale e creatività

L'obiettivo del presente lavoro è individuare le pratiche di insegnamento e le cornici teoriche per l'apprendimento presenti in letteratura funzionali allo sviluppo congiunto del pensiero computazionale e della creatività nella scuola primaria⁵.

A tal fine sono stati esaminati i contributi emersi da una rassegna sistematica più ampia riguardante interventi per lo sviluppo del pensiero computazionale nella scuola primaria. Sono stati esaminati i lavori (articoli in rivista, capitoli in volume, atti di Convegni internazionali) presenti in dieci database elettronici rilevanti per le scienze sociali, scienze dell'educazione e tecnologie informatiche, pubblicati tra gennaio 2006 e dicembre 2018, con i seguenti criteri di inclusione:

- a. contributi in lingua inglese;
- b. il termine “computational thinking” presente nell'abstract e/o nelle keywords;
- c. facenti riferimento a ricerche empiriche nella fascia K-6. Dalla rassegna sono emersi 53 studi.

Ai fini del presente lavoro, ci soffermeremo sui quattro che, tra questi, si concentrano anche sullo sviluppo della creatività attraverso framework teorici e attività appropriate.

I quattro lavori esaminati associano tutti il pensiero computazionale al pensiero creativo come approccio al problem solving, facendo riferimento a una visione dell'apprendimento socio-costruttivista e costruzionista (Jonassen, 1999; Harel & Papert, 1991; Vygotsky, 1974). In questa ottica lo sviluppo del pensiero computazionale viene abbinato a quello della creatività così come di altri tra quelli che vengono chiamati “21st century cognitive skills”.

Il primo lavoro si sofferma sulla valutazione dei processi di pensiero computazionale nell'ambiente classe con studenti di 10-11 anni di una scuola primaria londinese, che hanno preso parte a un progetto di creazione di giochi usando, per otto mesi, gli strumenti Scratch e Alice 2.4 (Allsop, 2018). Vengono presi in esame, in particolare, tre aspetti: i concetti computazionali (come le sequenze, i cicli, il parallelismo, gli operatori condizionali, gli eventi...), le pratiche metacognitive e i comportamenti di apprendimento.

Tra questi, viene inclusa anche la creatività, oltre alla collaborazione, la perseveranza, la comunicazione e il *problem solving*. Sul piano metodologico, lo studio adotta un approccio qualitativo, di impostazione etnografica longitudinale, e si sofferma pertanto sull'evoluzione nel tempo dei processi di ragionamento, con una triangolazione dei dati provenienti da diversi strumenti di rilevazione: osservazioni partecipanti, interviste semistrutturate, conversazioni informali, interviste in profondità, analisi dei prodotti finali, e *problem solving sheets*, intendendo con questi ultimi dei diari con guide semistrutturate riportanti la descrizione di come sono stati affrontati e risolti i problemi. Nello specifico del rapporto con la creatività, esso non è dunque stato oggetto di specifiche rilevazioni, ma dall'analisi dei prodotti finali è emerso come i bambini sviluppassero idee che implicavano il *decision-making*, il pensiero critico, il *problem solving*, il progettare soluzioni, aspetti questi che l'autrice riconduce alla creatività (Allsop, 2018, p. 8). Inoltre vengono riportati brani di interviste nei quali alcuni bambini, sollecitati su quanto avessero appreso durante l'attività proposta, evidenziavano in particolare il ruolo dell'espressione della propria immaginazione. Lo studio conclude dunque che la creatività, al pari degli altri comportamenti di apprendimento, fosse «visibile mentre i bambini stavano programmando i loro giochi» (Allsop, 2018, p. 9). Il secondo lavoro riporta i risultati di una ricerca valutativa riguardante un progetto di *Scalable Game Design* nel cantone svizzero di Solothurn finalizzato al suo inserimento, come attività di sviluppo del pensiero computazionale, nel curriculum scolastico nella scuola primaria (Lamprou et al., 2017). Il progetto riguardava sia la formazione degli insegnanti, sia attività degli studenti finalizzata allo sviluppo di *CT Concepts* tramite l'uso di due strumenti online: *AgentSheets* e *AgentCubes* (ambienti di programmazione di giochi e simulazioni 2-D e 3-D). Il risultato complessivo della ricerca mostra la fattibilità ma anche la piacevolezza di un'istruzione al pensiero computazionale usando ambienti di *Scalable Game Design*, dimostrando anche la sostenibilità e l'efficacia di tali strumenti nell'implementazione di un curriculum di informatica a livello scolastico. In questo quadro, durante le rilevazioni effettuate per valutare il progetto (sono stati intervistati 7 insegnanti, 133 studenti e 67 genitori), gli insegnanti hanno affermato come l'uso dei programmi *AgentSheets* e *AgentCubes* online promuove la creatività nei bambini, mentre la maggioranza di risposte di questi

ultimi hanno particolarmente evidenziato la possibilità, offerta da questi strumenti, di «usare liberamente la loro immaginazione, essere creativi e lavorare in maniera indipendente sui loro progetti/giochi» (Lamprou et al., 2017, p. 222).

Il terzo lavoro (Wong, Cheung, 2018) ha tra i suoi obiettivi quello di indagare come programmi di sviluppo del pensiero computazionale abbiano un impatto su altri costrutti tra i quali il pensiero creativo (gli altri sono il pensiero critico e il *problem solving*). In base alla teoria costruzionista, gli studenti (classe quarta e quinta di scuola primaria, classe prima di scuola secondaria) hanno partecipato ad attività di creazione di artefatti digitali usando Scratch e Kodu sotto forma di un progetto *game-based*, con l'obiettivo di coltivare nel contempo abilità di *problem solving* e pensiero critico, così come potenziare la creatività; i bambini, infatti, erano lasciati liberi di modificare e personalizzare i programmi. Le rilevazioni – avvenute per mezzo del test APASO-II comunemente usato in Cina, somministrato sia prima, sia dopo l'intervento – mostrano come vi sia in effetti un potenziamento della creatività, del pensiero critico e delle abilità di *problem solving*. Lo sviluppo della creatività, in particolare, appare maggiormente significativo rispetto anche alle altre due abilità indagate, in tutti e tre i gradi scolastici presi in esame⁶. Ai ragazzi sono anche state somministrate interviste semistrutturate sia prima sia dopo l'intervento; tra le domande, è stato chiesto quali abilità sono state sviluppate grazie alle attività di programmazione. La creatività è stata menzionata più spesso di quanto non lo siano state le abilità tecniche relative all'uso del computer, anche se, ad una netta prevalenza nella prima intervista, nella seconda, successiva all'intervento, le menzioni riguardanti abilità di pensiero e di *problem solving* sono state più numerose. Gli autori osservano che lo sviluppo della creatività dipende dalla libertà fornita nella formulazione dei compiti: se un certo ammorbidimento nei vincoli può consentire la messa in opera del pensiero creativo nello sviluppo degli artefatti digitali e dei programmi utili a raggiungere gli obiettivi, eccessiva libertà e flessibilità può al contrario penalizzarla in quanto gli studenti potrebbero non raggiungere gli obiettivi stessi (Lamprou et al., 2017, p. 9).

Infine il quarto lavoro si rivolge a 109 studenti⁷ di classe prima di scuola secondaria di primo grado e ha per obiettivo l'integrazione del pensiero computazionale con l'educazione artistica tramite l'uso di risorse digitali, sensori e minicomputer con un approccio

didattico *student-centered* (Sáez-López, Sevillano-García, 2017). In particolare, gli studenti hanno lavorato con Scratch e Raspberry Pi, che interfacciava il noto ambiente di programmazione a un microcomputer della misura di una carta di credito, che si collegava a una chitarra. Hanno creato una serie di progetti programmando a blocchi e usando in un secondo momento l'hardware e altre componenti. Lo studio ha mostrato che tali attività hanno favorito lo sviluppo del pensiero computazionale suonando e creando musica, dal momento che lavorare con il *coding* e con i dispositivi porta un vantaggio aggiuntivo nello sviluppo del pensiero computazionale e della competenza digitale. L'aspetto interessante ai fini del presente lavoro è l'incremento della creatività e della competenza artistica in relazione all'abilità di creare musica in attività e con strumentazioni proprie di un intervento tecnologico.

Dall'analisi di questi quattro studi, che forniscono una panoramica dello stato dell'arte sulle ricerche sul pensiero computazionale nella scuola primaria, appare in primo luogo come la ricerca sullo sviluppo combinato di pensiero computazionale e pensiero creativo sia ancora alle prime battute, seppure entrambi siano annoverati tra le abilità chiave di *problem solving* nel XXI secolo.

Dalla nostra rassegna, infatti, appare come solo quattro studi empirici con metodologia robusta siano stati condotti per sviluppare congiuntamente pensiero computazionale e creatività all'interno di quadri teorici appropriati. Tre tra questi coinvolgono gli studenti in attività di programmazione di giochi mentre il quarto propone attività collegate all'educazione artistica, tramite l'uso di risorse tecnologiche, sensori e minicomputers. Tutte e quattro, in proporzione rispetto ai diversi impianti metodologici, hanno mostrato lo sviluppo della creatività. Studi futuri dovrebbero interessarsi a come i concetti e le pratiche inerenti al pensiero computazionale possano essere sviluppati e valutati nei diversi gradi scolastici nell'ambito della creatività e vice versa; sarebbe di grande interesse indagare come approcci combinati possano essere adottati in ambiti diversi da quello delle scienze informatiche strettamente intese o in attività prive del vero e proprio *coding* o addirittura *unplugged*. In questo modo si potrebbero ottenere indicazioni importanti nella costruzione di curricula adatti per preparare gli studenti di oggi a essere al passo delle attuali evoluzioni digitali.

Note

¹ Francesco C. Ugolini è l'autore del paragrafo 1., mentre Panagiotis Kakavas è l'autore del paragrafo 2. I contenuti di questo secondo paragrafo riprendono e integrano parte del contributo *Computational thinking and creativity in K-6 education*, dello stesso Panagiotis Kakavas.

² Come fotografia delle politiche di inserimento di attività di *coding* nella scuola primaria in Europa, si può far riferimento al rapporto EUN del 2015. Per quello che riguarda l'Italia, possiamo citare il Piano Nazionale Scuola Digitale (MIUR, 2015), che prevede una specifica azione (la #17: «portare il pensiero logico-computazionale a tutta la scuola primaria») e nel quale si presenta la competenza digitale come «“alfabeto” del nostro tempo – al cui centro risiede il pensiero computazionale – una nuova sintassi, tra pensiero logico e creativo, che forma il linguaggio che parliamo con sempre più frequenza nel nostro tempo» (Ivi, p. 73).

³ Il più noto è forse Scratch, sviluppato dal MIT (scratch.mit.edu).

⁴ I passaggi riportati sono stati tradotti da Olimpo (2017, p. 16) che a sua volta ne ha sottolineato la significatività.

⁵ Si fa qui riferimento alla fascia denominata a livello internazionale con la sigla K-6, che comprende, cioè, la scuola dell'infanzia e i primi 6 anni di scuola del primo ciclo, corrispondenti in Italia, alla scuola primaria e alla prima classe della scuola secondaria di primo grado.

⁶ L'Effect Size, calcolato in base alla *d* di Cohen, risulta 0,47 per la creatività rispetto a 0,33 e 036 del pensiero critico e del *problem solving* rispettivamente

⁷ Altri 39 studenti costituivano il gruppo di controllo.

Bibliografia

Allsop Yasemin (2019), *Assessing computational thinking process using a multiple evaluation approach*, "International Journal of Child-Computer", V. 19, pp. 1-26

Anderson Nicole D. (2016), *A call for computational thinking in undergraduate psychology*, "Psychology Learning & Teaching", V.15, N. 3, pp. 226-234

Barr Valerie, Stephenson Chris (2011), *Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?*, V. 2, N. 1, "ACM Inroads", pp. 111-122

Berthoz Alain (2009), *La simplicité*, Paris, Odile Jacob

Brennan Karen, Resnick Mitchel (2012), *New frameworks for studying and assessing the development of computational thinking*. Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA'12), pp. 1-25

Commissione Europea (2016), *A new skills agenda for Europe. Working together to strengthen human capital, employability and competitiveness*. Gazzetta Ufficiale dell'Unione Europea.
ec.europa.eu/transparency/regdoc/rep/1/2016/EN/1-2016-381-EN-F1-1.PDF

Dufva Tomi, Dufva Mikko (2016), *Metaphors of code: structuring and broadening the discussion on teaching kids to code*, "Thinking Skills and Creativity", V. 22, pp. 97-110
<https://www.sciencedirect.com/science/article/pii/S1871187116301055>

EUN (2015), *Computer programming and coding. Priorities, school curricula and initiatives across Europe*, European Schoolnet
www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf

Gallese Vittorio, Lakoff George (2005), *The Brain's Concepts: The Role of the Sensory-Motor System in Reason and Language*, "Cognitive Neuropsychology", V. 22, N. 3-4, pp. 455-479

Grover Shuchi, Pea Roy D. (2013), *Computational Thinking in K-12: A Review of the State of the Field*, "Educational Researcher", V. 42, N. 1, pp. 1-6

Harel Idit, Papert Seymour (1991), *Constructionism*, Norwood, NJ, Ablex Publishing

Jonassen David H. (1999), *Designing constructivist learning environments*, in Reigeluth Charles M. (ed.), *Instructional-design theories and models: A new paradigm of instructional theory*, University Park, Pennsylvania State University, pp. 215-239

Lamprou Anna, Reppening Alexander, Escherle Nora A. (2017), *The Solothurn Project — Bringing Computer Science Education to Primary Schools in Switzerland*. Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '17, 3-5 July. Bologna, Italy, pp. 218-223

Olimpo Giorgio (2017), *Dal mestiere dell'informatico al pensiero computazionale*, "Italian Journal of Educational Technology", V. 25, N. 2, pp. 15-26

Papert Seymour (1980), *Mindstorms. Children, computers and powerful ideas*, New York, Basic Books

Pérez-Marín Diana, Hijón-Neira Raquel, Baceo Adrián, Pizarro Celeste (2018), *Can computational thinking be improved by using a methodology based on metaphors and Scratch to teach computer programming to children?* "Computers in Human Behaviour"

Rivoltella Pier Cesare (2012), *Neurodidattica. Insegnare al cervello che apprende*, Milano, Raffaello Cortina

Rivoltella Pier Cesare, Rossi Pier Giuseppe (2019), *Il corpo e la macchina. Tecnologia, cultura, educazione*, Brescia, Morcelliana

Sáez-López José-Manuel, Sevillano-García María-Luisa (2017), *Sensors, programming and devices in Art Education sessions. One case in the context of primary education*, "Cultura y Educación", V. 29, N. 2, 350-384

Sibilio Maurizio (2012), *Il corpo e il movimento nella ricerca didattica. Indirizzi scientifico-disciplinari e chiavi teorico-argomentative*, Napoli, Liguori

Trincherò Roberto (2012), *Costruire, valutare, certificare competenze. Proposte di attività per la scuola*, Milano, Franco Angeli

Trincherò Roberto (2019), *Problem solving e pensiero computazionale. Costruire sinergie tra concettualizzazione e codifica a partire dalla scuola primaria*, "Form@re", V. 19, N. 1, pp. 78-90

Varela Francisco J. (1990), *Il corpo come macchina ontologica*, in Ceruti Mauro, Preta Lorena, *Che cos'è la conoscenza*, Bari, Laterza

Vygotskij Lev Semënovič (1978), *Mind in society: The development of higher psychological processes*, Rev. ed. by Cole Michael, John-Steiner Vera, Scribner Sylvia, Souberman Ellen [With Introduction; Biographical Note by Cole Michael, Scribner Sylvia; Afterword by John-Steiner Vera, Souberman Ellen], Cambridge MA, Harvard University Press
https://books.google.it/books?id=RxjIUefze_oC&pg=PA1&hl=it&source=gbs_toc_r&cad=3#v=onepage&q&f=false

Wing Jeannette M. (2006), *Computational thinking*, "Communications of the ACM", V. 49, N. 3, pp. 33-35
https://www.researchgate.net/publication/274309848_Computational_Thinking

Wong Gary Ka-Wai, Cheung Ho-Yin (2018), *Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming*, "Interactive Learning Environments", V. 26, 25 October, pp. 1-14

