

Coding alla scuola dell'infanzia con docente esperto della scuola primaria

Coding in preschool with expert primary school teacher

Giovanni Nulli and Margherita Di Stasio*

Istituto Nazionale di Documentazione, Innovazione e Ricerca Educativa, Firenze, Italy, g.nulli@indire.it, m.distasio@indire.it*

*corresponding author

HOW TO CITE Nulli, G., & Di Stasio, M. (2017). Coding alla scuola dell'infanzia con docente esperto della scuola primaria. *Italian Journal of Educational Technology*, 25(2), 59-65. doi:10.17471/2499-4324/902

1. INTRODUZIONE

Il pensiero computazionale è stato proposto come tema fondamentale per la scuola nei famosi articoli di Janet Wing (2006, 2008, 2010).

La sperimentazione qui presentata, parte del progetto *Coding@scuola*, contenuto nel Piano Triennale delle Attività 2015-2017 di Indire, ha l'obiettivo di studiare percorsi di pensiero computazionale nell'ultimo anno della scuola dell'infanzia in una prospettiva di verticalizzazione e focalizzazione sulle aree delle Indicazioni Nazionali per il Primo Ciclo. Tale prospettiva ci appare inoltre particolarmente fruttuosa se ricondotta all'azione #17 del Piano Nazionale Scuola Digitale, che individua proprio nella scuola primaria il segmento strategico per l'introduzione del pensiero computazionale a scuola.

2. POSIZIONAMENTO DEL PROBLEMA: QUALE APPROCCIO AL CODING?

In questa sede consideriamo il pensiero computazionale come una metodologia strutturata di *problem solving* che ha come oggetto e strumento l'uso di computer o facilitatori robotici.

Deriviamo questa definizione da ISTE in *Operational definition of computational thinking*¹ e dal documento di supporto con il quale il Department of Education ha accompagnato l'uscita del curriculum di *coding* in Inghilterra (Berry 2015).

Sia in ISTE che in Berry (2015) si evince che per pensiero computazionale si intende un processo di *problem solving*. ISTE lo dichiara esplicitamente, Berry (2015) lo definisce come un modo di guardare ai problemi (p. 6), affermando poi (p. 8) che il pensiero computazionale non è solo un “*new name for problem-solving-skills*”, ma ha a che fare con il processo rigoroso e creativo di scrivere codice.

ISTE identifica un set non chiuso di *skills*, che chiama anche *characteristics*, accompagnato da set ugualmente non chiuso di *attitudes*.

¹ <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>

Più articolato è il lavoro di Berry, volto a dare operatività al concetto di pensiero computazionale. Il “pensatore computazionale” si caratterizza per l’interiorizzazione di *concepts* e di *approach* all’attività pratica. Per Berry i *concepts* derivano da *processes*, ovvero da reti di azioni, tipiche del pensiero computazionale. Gli *approaches* sono modi di lavorare dei programmatori.

Possiamo quindi dire che per ambedue il pensiero computazionale ha a che fare con la risoluzione dei problemi attraverso i computer e con una serie di concetti, attitudini, abilità e modi di lavorare che riassumiamo come segue:

- saper analizzare il problema: sia in ISTE che in Berry sono presenti *skills* e *concepts* che si riferiscono alla rilevazione dei dati in funzione delle procedure da costruire;
- saper costruire procedure eseguibili da un computer: sia ISTE che Berry parlano di *skills* o *concepts* riferibili alla competenza nella scrittura di algoritmi;
- saper lavorare su problemi aperti e in situazioni complesse: ISTE identifica una specifica *skill* per la complessità, mentre in Berry l’intero concetto di *approaches* può essere ricondotto a una dimensione pratica ed ecologica, che implica di aver a che fare con approssimazioni, aggiustamenti, soluzioni, tipiche dei sistemi non lineari;
- saper lavorare in gruppo: sia ISTE che Berry identificano una *skill* o un *approach* in merito a ciò.

Questa accezione ci consente di tralasciare alcuni aspetti, solitamente evidenti ed evidenziati nelle pratiche di *coding* a scuola, che a nostro avviso sarebbero problematici, nonché secondari, rispetto alla sperimentazione oggetto di questo contributo, perché non riguardano direttamente gli ordini di scuola coinvolti.

In prima battuta tutto ciò che riguarda il pensiero computazionale in relazione all’avviamento professionale o verso una carriera informatica. Questo indirizzo è quello indicato dall’Unione Europea con iniziative quali la *CodeWeek*².

In questa sede, invece, riteniamo che introdurre in modo precoce pratiche legate alla programmazione possa essere utile in contesti molto diversi da quello del programmatore e legarsi a un metodo di lavoro centrato sullo studente.

Altra questione, collegata al tema del lavoro, è la scelta dei linguaggi di programmazione che si dimostrino adatti alla scuola. Infatti, insegnare precocemente uno specifico linguaggio pone due ordini di problemi:

- il primo legato all’effettiva difficoltà, in quanto tale linguaggio non è pensato come strumento didattico;
- il secondo legato al fatto che tra quindici anni tale linguaggio sarà quasi sicuramente superato³.

Di conseguenza, almeno per la scuola primaria e sicuramente per l’infanzia, ordini coinvolti in questo progetto, affrontare questo tipo di problemi non sarebbe né utile né fruttuoso.

Le cosiddette interfacce tangibili, come *Cubetto* o *Dash and Dot*, e alcuni software, come *Robot Turtles* o anche *Scratch Jr.*, stanno abbassando l’età a cui è possibile approcciarsi al *coding* (Firth, 2014). In particolare i *floor robot* e le interfacce tangibili - evoluzioni della *Logo Floor Turtle*, come *Bee e Blue Bot*, *Dash and Dot*, e anche lo stesso *Cubetto* - consentono di “manipolare e capire” (Manches & Plowman, 2017, p. 197).

Le pressioni affinché la scuola vada in questa direzione, come storicamente è avvenuto per molte tecnologie (Ranieri, 2011), possono provenire dall’esterno, da produttori di hardware e software, come dall’interno, da genitori o educatori. Ma al di là di questo vanno identificati i motivi pedagogici, se ce ne sono, per l’inserimento di tali supporti a età precoci. Oltre l’effettiva fattibilità, ne va dunque valutata l’opportunità, ponendosi domande specifiche: «quali strumenti offrono maggiori vantaggi e in quale fase dello sviluppo? Come

² http://ec.europa.eu/archives/commission_2010-2014/kroes/content/main-achievements-under-neelie-kroes-mandate-2010-2014.html. Si consideri come nel mandato 2010-2014 del commissario Neelie Kroes, che ha fortemente supportato la CodeWeek, questa venga inserita nel capitolo “Multiplying digital skill, jobs entrepreneurship”.

³ Penge, S. (2015). Piccolo è meglio? Sì, ma... Indire. Retrieved from <http://www.agenziascuola.it/content/index.php?action=read&id=1862&graduatorie=0>

supportano diversi concetti di programmazione? In che misura le capacità acquisite con uno strumento si trasferiscono agli altri o a idee più ampie all'interno e oltre la programmazione? Ora che la commercializzazione si rivolge a genitori e insegnanti, qual è il ruolo dell'adulto nell'utilizzo di questi strumenti con i bambini?» (Manches & Plowman, 2017, p. 197).

Per Bers (2010), l'uso di oggetti tangibili in processi di progettazione, flussi, *loop* e parametri, uso di sensori ecc. portano con sé idee potenti per l'apprendimento in età precoce; talvolta difficili a un primo approccio, possono essere veicolati attraverso concetti più familiari al mondo della scuola come “causa ed effetto”, “storytelling”, “senso del numero” o “osservazione scientifica”. In Kazakoff, Sullivan & Bers (2013), si dimostra come lavorare con mediatori robotici porti benefici allo *storytelling* e questo suggerisce come questo tipo di attività non sia solo un prerequisito per attività di informatica.

3. OBIETTIVI E METODOLOGIA

Abbiamo, su questa base, avviato un progetto di ricerca e sperimentazione con le scuole i cui obiettivi sono:

- la validazione della definizione di pensiero computazionale sopra esposta;
- la sua integrazione nella scuola in un'ottica di verticalizzazione;
- la sua applicabilità nella didattica quotidiana a sostegno del raggiungimento delle competenze indicate nelle Indicazioni Nazionali.

I tre Istituti Comprensivi coinvolti (IC Gobetti di Rivoli, Torino, IC di Rovoletto di Cadeo Piacenza, Secondo IC di Francavilla Fontana, Brindisi) in questo progetto si caratterizzano per alcuni aspetti comuni. Hanno lavorato attivamente su progetti, anche ad alto impatto tecnologico portati avanti a livello sistemico e che quindi possano affrontare anche a livello organizzativo i cambiamenti che questo tipo di esercizi possono comportare. Si tratta inoltre di istituti dove è presente un docente di scuola primaria con un'esperienza pregressa e strutturata di *coding*.

Abbiamo cercato di trovare un rapporto con la scuola ponendoci come sostegno, ma anche impostando una situazione di ascolto e comprensione reciproca. Per sostenere un'innovazione che vada al di là dei risultati conseguibili in una singola aula, il ricercatore deve comprendere la realtà in cui va ad operare, una realtà «*che chiede alla ricerca educativa di 'proporsi come consulenza efficace' per la 'soluzione di problemi educativi ben determinati'»* (Nigris, 2000, p. 164). Per questo abbiamo guardato in prima battuta al quadro della ricerca-azione, il cui «*scopo è promuovere un'azione di cambiamento su e per i soggetti coinvolti*» (Besozzi & Colombo, 1998, p. 112).

La valenza della ricerca azione trova ulteriore linfa nella misura in cui l'intervento è costruito come «*ricerca collaborativa tra ricercatori pratici e ricercatori di professione, che insieme costruiscono una 'comunità di ricerca'»* (Mortari, 2013, p. 94).

L'impostazione ha dunque guardato alla ricerca collaborativa che si presenta come particolarmente connessa al mondo della scuola e a quello della ricerca e, per la propria struttura, prevede un coinvolgimento di scuola, docenti, dirigente e personale interessato (Magnoler, 2012, p. 116), elemento che si mostrò indispensabile a innescare processi di miglioramento (si vedano ad esempio i progetti VSQ⁴ e Vales⁵) e a sostenere reali processi di innovazione, come nelle esperienze delle Avanguardie Educative⁶.

Abbiamo scelto di lavorare cooperando con le scuole per inserire il tema del pensiero computazionale nel vissuto

⁴ <http://for.indire.it/vsq2012/>

⁵ <http://for.indire.it/vales2014/>

⁶ <http://avanguardieeducative.indire.it/>

scolastico pratico e ponendo la verticalizzazione dell'esperienza come uno dei tratti caratterizzanti del progetto. La verticalizzazione rappresenta un tema di interesse da due punti di vista, quello organizzativo e quello didattico. Dal punto di vista organizzativo chiediamo ai docenti di lavorare insieme, definendo ruoli, sia a livello di progettazione che a livello di lavoro in classe. I docenti sono liberi di creare il sistema di collaborazione che preferiscono, che può andare dal tutoraggio semplice a sistemi di compresenza e collaborazione tra le classi. Questo può implicare il dover lavorare con l'orario, e trovare luoghi di collaborazione, e in questo la collaborazione della scuola come organizzazione è una necessità. La rilevazione di quanto avviene anche da questo punto di vista servirà alla descrizione della pratica.

Dal punto di vista didattico, oltre a mettere insieme due differenti modi di lavorare, si avvia un percorso sugli studenti, che nell'anno successivo potranno continuare il lavoro già avviato all'infanzia, nella scuola primaria. Sull'importanza di questo passaggio (e degli ulteriori sviluppi, magari alla scuola secondaria di primo grado), si potrà riflettere negli eventuali sviluppi futuri del progetto.

4. STRUTTURA E ATTUAZIONE

Il progetto è strutturato in più passaggi:

- un momento preparatorio in cui i ricercatori hanno identificato le linee fondamentali, già espone, dal punto di vista dell'idea di *coding* soggiacente al progetto e riguardo alla metodologia da utilizzare;
- un recupero del pregresso dei docenti in termini di esperienze di *coding* e di collaborazione in verticale infanzia primaria, da poter valorizzare come base per il lavoro;
- un avvio operativo, con un momento di condivisione, delle linee e delle esperienze, e di co-progettazione;
- la sperimentazione propriamente detta, prevista su tutta annualità scolastica 2016/2017, e la raccolta dati (in atto).

Per attuare il recupero dell'esperienza, in prima battuta abbiamo proposto ai docenti di raccontare il proprio pregresso attraverso un format narrativo con domande guida orientate a indagare il contesto; le motivazioni, la progettazione e la valutazione dell'attività.

Abbiamo quindi proposto un questionario in cui indagavamo vari aspetti quali: le precedenti esperienze di collaborazione e/o verticalizzazione; i traguardi di esperienza ritenuti collegati alle attività di *coding*; le modalità di rilevazione e verifica di quanto appreso sia in relazione ai traguardi di esperienza che in vista di attività centrate sul *coding*.

Il progetto è divenuto operativo con il coinvolgimento delle scuole ad aprile 2016 durante un seminario residenziale rivolto ai docenti, durante il quale è stato condiviso e validato il metodo di lavoro, ovvero l'unica parte strutturata in modo vincolante della proposta che i ricercatori hanno fatto ai docenti. Non sono infatti stati posti vincoli stretti né si è cercata altra uniformità al di là della verticalizzazione e della condivisione del metodo per poter, in coerenza con le premesse definite, porci nei confronti della scuola come un supporto utile alla loro crescita e per permettere l'emersione delle loro pratiche.

Il metodo di lavoro proposto è strutturato in fasi al cui centro ci sono compiti che dovranno essere elaborati secondo lo schema *Think, Make, Improve* (TMI), mutuato da Martinez e Stager (2013) e già utilizzato nel progetto Indire *Costruire giocattoli con la stampante 3d*⁷. Sulla scorta di questa esperienza pregressa, ipotizziamo che la tripartizione TMI offra ai docenti una struttura facilmente traducibile nella loro progettazione ed efficacemente attuabile nella pratica d'aula, dove può costituire un supporto alla metacognizione degli studenti. (Borri, Benassi, Guasti, & Nulli, 2015)

⁷ <http://www.indire.it/progetto/maker-a-scuola/costruire-giocattoli-con-la-stampante-3d/>

Nella fase *Think*, i docenti sono stati invitati a formulare accuratamente la consegna, in modo che sia comprensibile dagli studenti, ma sia espressa il più possibile senza contenere una soluzione diretta: i bambini pensano al problema e alla soluzione, sono invitati a discutere e a fare disegni. Ai docenti è stato chiesto di descrivere, nella progettazione, cosa pensano accadrà in classe. Nella fase *Make* ai docenti viene chiesto di pensare a cosa faranno loro e gli studenti in classe. Agli studenti viene chiesto di creare quanto hanno progettato con il software o con il mediatore robotico a seconda del compito nella prima fase. La fase *Improve* prevederà controllo, *testing* e discussione.

La parte centrale del metodo di lavoro (Figura 1) è il “compito”, che racchiude sia le fasi del TMI sia la “consegna problematica”, ovvero il modo in cui il problema, che genera l’attività del TMI, viene impostato. “Introduzione” e “avvicinamento” sono i due momenti che, con il compito costituiscono una fase. Questi due momenti servono ai docenti per preparare la classe; in particolare il momento di “introduzione” serve a creare le aspettative negli studenti attraverso l’uso di uno “sfondo narrativo” (preparato in precedenza), mentre nel momento di “avvicinamento” viene introdotto il tema del compito, senza l’uso dei mediatori tecnologici.

È anche previsto un momento di lavoro preliminare viene effettuata la progettazione delle fasi e la creazione di elementi, come lo sfondo narrativo, necessari all’attività vera e propria.

Lungo tutto l’arco del progetto è prevista poi una diaristica per i docenti e per i ricercatori che costituirà la base per i momenti di confronto, scambio e ed eventuale riprogettazione in coerenza con esigenze e situazioni inaspettate che andranno emergendo. Lo scambio informale tra i partecipanti è supportato dall’uso di strumenti di *social networking*.

Le tempistiche dei singoli compiti e delle specifiche attività verranno decise dai docenti in sperimentazione che potranno anche utilizzare le dotazioni in loro possesso. Indire ha messo a disposizione un *Cubetto*⁸, giudicato dai docenti in sede di seminario preliminare, come lo strumento più adatto da sperimentare.

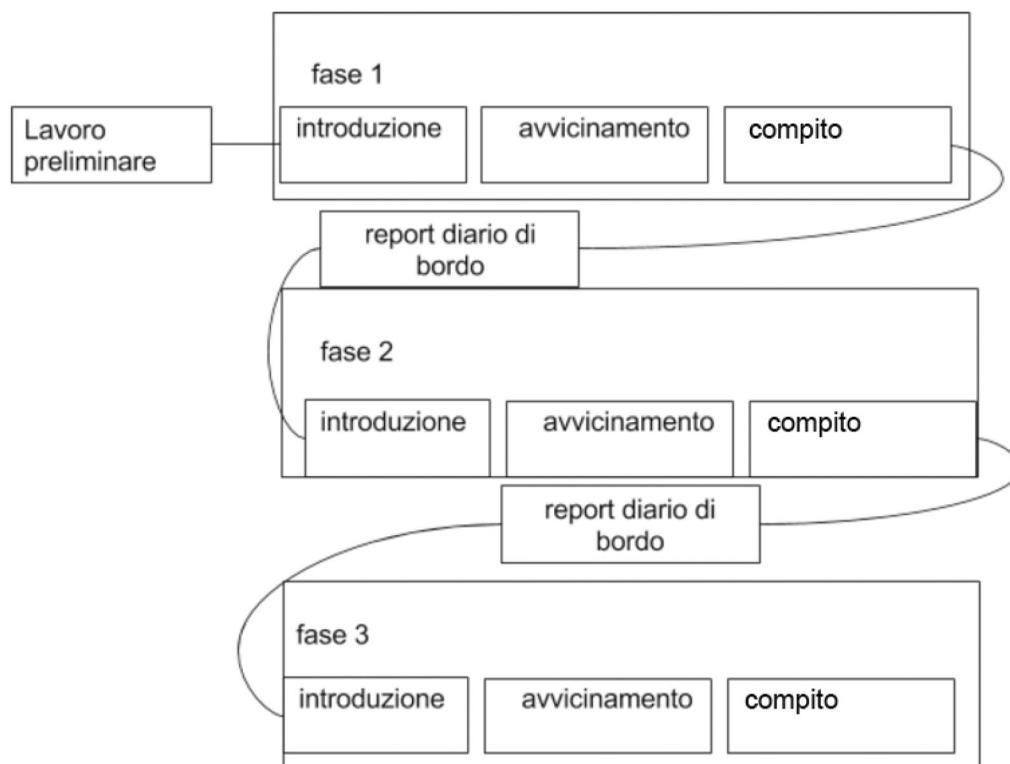


Figura 1. Strutturazione in fasi e loro articolazione interna.

⁸ <https://www.primotoys.com/>

5. RISULTATI OTTENUTI E ATTESI

In via generale, riteniamo che questo progetto porterà a osservare se e quanto proposto trova una validazione nella pratica; ha come ulteriore obiettivo quello di far emergere delle pratiche “ben descritte” e già sperimentate in classe che possano ambire ad essere facilmente riproducibili.

Allo stato attuale il primo risultato è stato verificare con i docenti che la sperimentazione possa essere avviata, quindi che per loro abbia un valore effettivo. Il secondo risultato è stata la discussione degli strumenti progettuali proposti, per arrivare a una soluzione che fosse più in linea possibile con le esigenze espresse dei docenti in sede di seminario.

Saranno oggetto di osservazione sul campo, di condivisione e confronto:

- la gestibilità e la tenuta delle fasi *TMI* così come ipotizzate in relazione alla realtà della pratica d’aula;
- l’impatto di queste attività in relazione alla definizione di *coding* proposta.

Auspichiamo inoltre che lavorare cooperando con le scuole ci consenta di inserire il tema del pensiero computazionale nel vissuto scolastico pratico e in particolare di indagare la funzionalità di questa metodologia in rapporto ai traguardi di esperienza e in relazione a percorsi di verticalizzazione.

6. BIBLIOGRAFIA

Berry, M. (2015). *Quick start Computing Primary Handbook*. Swindon, UK: BCS.

Bers, M. U. (2010). The Tangible K Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12(2). Retrieved from <http://ecrp.uiuc.edu/v12n2/bers.html>

Besozzi, E. & Colombo, M. (1998). *Metodologia della ricerca sociale nei contesti socioeducativi*. Milano, IT: Guerrini Studio.

Borri, S., Benassi, A., Guasti, L. & Nulli, G. (2015). *Costruire Giocattoli con la Stampante 3D nella Scuola dell’Infanzia. Spazi di utilizzo, sperimentazione e co-progettazione didattica*. In P. Calidoni & C. Casula. *Education 2.0: esperienze, riflessioni, scenari*. Cagliari, IT: Cuec Editrice.

Firth, N. (2014). Code generation. *New Scientist*, 223(2985), 38-41. Retrieved from <https://www.newscientist.com/>

Kzakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245-255. doi:10.1007/s10643-012-0554-5

Magnoler, P. (2012). *Ricerca e formazione. La professionalizzazione degli insegnanti*. Lecce, IT: Pensa MultiMedia.

Manches, A., & Plowman, L. (2017). Computing education in children’s early years: A call for debate. *British Journal of Educational Technology*, 48(1), 191-201. doi:10.1111/bjet.12355

Martinez, S. L., & Stager, G. (2013). *Invent to learn: Making, tinkering, and engineering in the classroom*. Torrance, CA: Constructing Modern Knowledge Press.

Mortari, L. (2013). *Ricercare e riflettere. La formazione del docente professionista*. Roma, IT: Carocci.

Nigris, E. (2000). *Un nuovo rapporto fra ricerca e educazione: la ricerca-azione*. In S. Mantovani (ed.), *La ricerca sul campo in educazione. I metodi qualitativi*. Milano, IT: Bruno Mondadori.

Ranieri, M. (2011). *Le insidie dell'ovvio. Tecnologie educative e critica della retorica tecno centrica*. Pisa, IT: ETS.

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

Wing, J.M. (2008). Computational Thinking and Thinking about Computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725.
doi:10.1098/rsta.2008.0118

Wing, J.M. (2010). Computational Thinking: What and Why?. *Link Magazine*. Retrived from:
<https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>