

Il Pensiero Computazionale, questo sconosciuto

Fabrizio Basciani

Istituto Comprensivo Pescara 2,
Docente di Tecnologia,
Via Vincenzo Cerulli 15, 65126, Pescara, Italy
fbasciani@wavecon.it

Sunto

Da qualche anno, nella Scuola, si sente parlare di Coding e di Pensiero Computazionale. Come spesso accade, tuttavia, quando il MIUR intende introdurre forti innovazioni in tempi relativamente brevi nelle Metodologie Didattiche, tutti ne parlano, ma pochi comprendono esattamente di cosa si tratti e, soprattutto, pochissimi lo utilizzano nella quotidianità operativa delle lezioni.

Perché? Scarsità di Formazione? Poche Risorse Digitali? Resistenza al cambiamento da parte dei docenti? Proviamo a dare una risposta a queste domande, con l'obiettivo di trasmettere il messaggio che non bisogna arrendersi dinanzi agli ostacoli della Professione Docente, in quanto gli studi nazionali ed internazionali sull'applicazione didattica del Pensiero Computazionale dimostrano che, in termini di efficacia, vale la pena impegnarsi nella sua divulgazione, sia come strumento di potenziamento delle competenze più sviluppate, sia per il recupero e l'inclusione degli alunni più deboli, o con problemi di apprendimento.

Parole chiave: Pensiero Computazionale, Coding, Metodologie Didattiche Innovative, Scuola Digitale, Coding unplugged, Coding inverso.

1. Introduzione

L'Italia è uno dei primi Paesi al mondo che ha iniziato, in via sperimentale, l'introduzione sistematica nelle scuole dei concetti di base del Pensiero Computazionale, adottando il Coding come metodologia didattica e semplici risorse software come strumenti.

Ciò è avvenuto a seguito dell'importante spinta del MIUR, che, ad esempio, nella Circolare Prot. N. 9759 del 08/10/2015, dichiarava: *"Nel mondo odierno i computer sono dovunque e costituiscono un potente strumento per la comunicazione. Per essere culturalmente preparato a qualunque lavoro uno studente vorrà fare da grande è*

indispensabile quindi una comprensione dei concetti di base dell'informatica. Esattamente com'è accaduto nel secolo passato per la matematica, la fisica, la biologia e la chimica. Il lato scientifico-culturale dell'informatica, definito anche pensiero computazionale, aiuta a sviluppare competenze logiche e capacità di risolvere problemi in modo creativo ed efficiente, qualità che sono importanti per tutti i futuri cittadini. Il modo più semplice e divertente di sviluppare il "pensiero computazionale" è attraverso la programmazione (coding) in un contesto di gioco."

Anche il Piano Nazionale Scuola Digitale, introdotto dalla Legge n. 107 / 2015, prevede *"un'appropriata educazione al pensiero computazionale, che vada al di là dell'iniziale alfabetizzazione digitale, essenziale affinché le nuove generazioni siano in grado di affrontare la società del futuro non da consumatori passivi ed ignari di tecnologie e servizi, ma da soggetti consapevoli di tutti gli aspetti in gioco e come attori attivamente partecipi del loro sviluppo"*.

Questi concetti, assolutamente condivisibili dal punto di vista della preparazione al mondo del lavoro e della missione dell'Istituto dell'Istruzione, che sempre più si configura come scuola di vita, oltre che come Ente di formazione nelle varie Discipline, vanno tuttavia analizzati da molteplici punti di vista, poiché, come sempre accade, quando si pianifica un Progetto, soprattutto un Progetto importante come quello che implica l'ennesima piccola rivoluzione dell'impostazione metodologica del Primo Ciclo, bisogna preliminarmente verificare se tutti i tasselli dello stesso sono disponibili e pronti per essere utilizzati: tempi, costi, qualità del servizio, risorse umane, risorse tecniche, capacità di comunicazione, analisi dei rischi connessi.

E l'impressione che si ha, accostandosi alla scuola italiana del 2018, come docenti, ma anche come genitori e/o alunni, è che il MIUR abbia fatto, per così dire, i conti senza l'oste.

Ciò nonostante, non bisogna arrendersi dinanzi agli ostacoli della Professione Docente, perché gli studi nazionali ed internazionali sull'applicazione didattica del Pensiero Computazionale dimostrano che vale la pena impegnarsi nella sua divulgazione, sia come strumento di potenziamento delle competenze più sviluppate, sia per il recupero e l'inclusione degli alunni più deboli, o con problemi di apprendimento.

E siamo sicuri che, come da sempre accade, la personale contribuzione dei docenti, avvezzi da sempre ad organizzare con passione percorsi formativi coinvolgenti con le risorse a disposizione, nonché una sintetica ma sostanziale formazione sul tema (sia essa formale, informale, sul campo) riusciranno a vincere la classica paura della pagina bianca, che tormenta tutti coloro che si apprestano a scrivere nuove pagine di storia della Scuola.

Questo breve articolo, lungi dall'essere il depositario della verità sul Pensiero Computazionale, mira a focalizzare l'attenzione su due aspetti:

1. Fungere da lente di ingrandimento sulla metodologia del Coding, ripercorrendo le fasi di sviluppo della Pedagogia e della Didattica che hanno portato ad eleggerlo come principale palestra del Pensiero Computazionale;
2. Suggestire una lettura del Coding alternativa a quella prevalentemente

informatica, attualmente diffusa, che ne enfatizza, spesso, l'aspetto più commerciale e, a volte, ne rappresenta una devianza, concentrandosi invece sulle dinamiche di logica e di ragionamento che la pratica del Pensiero Computazionale riesce ad indurre negli allievi.

2. Analisi delle Risorse della Scuola per lo sviluppo del Pensiero Computazionale

Di Pensiero Computazionale nella Scuola si sente parlare, per vie ufficiali, da pochi anni. E il tempo di maturazione, si sa, nel settore pubblico, è una variabile fondamentale affinché burocrazia, organizzazione, atteggiamento mentale delle risorse coinvolte si predispongano ad accettare l'ennesima novità, in un ambiente lavorativo, ricordiamolo, letteralmente martoriato da riforme e controriforme che hanno riguardato non solo i contenuti della didattica, ma soprattutto gli aspetti pratici della docenza quotidiana: numero di ore lavorate, incertezza degli assetti logistici, variazione delle autonomie decisionali. È evidente che l'educatore medio italiano, da anni, non abbia esattamente la tranquillità mentale necessaria ad approcciare a metodologie didattiche troppo innovative, poiché impegnato su altri fronti, personalmente e professionalmente più impellenti.

Di costi pur si deve parlare, perché la formazione su nuove tematiche, soprattutto se basate sulle TIC, necessita di investimenti, sia sul capitale umano che sulle strutture necessarie ad accogliere e a sostenere nuove attività laboratoriali, nonché per acquistare le tecnologie, informatiche e non, e le attrezzature utili a rendere un Progetto realmente fruibile, evitando che rimanga un bel concetto fine a se stesso, seppur sapientemente enunciato con un gessetto sulla cara vecchia lavagna di ardesia. Ancora oggi, invece, la maggior parte delle aule è impostata con la classica disposizione frontale dei banchi, anziché l'innovativa flipped classroom, e spesso la LIM è uno strumento di difficile reperimento. Non parliamo poi delle metodologie BYOD (Bring Your Own Device, in sintesi la possibilità degli alunni di portare a scuola il proprio dispositivo informatico, connettendolo alla rete della scuola, per utilizzarlo nelle attività didattiche proposte dal docente), che inducono nel contesto scolastico serie problematiche di infrastrutture, di privacy e di sicurezza digitale.

Del resto, in merito alla formazione di tipo informatico, non bisogna tralasciare un aspetto fondamentale, spesso trascurato: gli adolescenti degli anni '80 e '90, che oggi sono diventati docenti, e che hanno vissuto la diffusione massiva, nel mondo del lavoro e della vita privata, del computer, di internet e della tecnologia informatica in genere, si trovarono allora nella condizione obbligata di dover necessariamente impararne da sé l'utilizzo, pena l'esclusione dai rispettivi contesti sociali.

Questo processo di implementazione culturale, vissuto in un'età in cui si è ben disposti ad affrontare le sfide, oltre che interessati a migliorare la qualità della vita e a differenziare le proprie conoscenze, ha generato una classe di utenti che, tutto sommato,

con il computer se la cavano, poiché nati “informaticamente” insieme alla materia, in un periodo in cui, per ottenere un risultato dalla macchina, era necessario scrivere righe di codice (e quindi conoscerne il significato), capirne il funzionamento, risolvere criticamente e pragmaticamente i (numerosi) problemi.

La generazione di utenti che, invece, nel periodo indicato, era più avanti negli anni, e che oggi è ancora attiva nell’insegnamento a scuola, si è spesso lasciata scivolare addosso il problema, ritenendo di non aver bisogno dell’informatica per migliorare la propria professionalità.

Paradossalmente, persino la generazione successiva ha problemi con l’“informatica ragionata”, perché, se è vero che gli alunni dell’attuale Primo Ciclo sono “millennials”, o “nativi digitali” che dir si voglia, è anche vero che essi, con il computer, non sanno andare al di là del click, senza porsi minimamente il problema delle conseguenze tecniche, meccaniche, elettriche, sociali, e soprattutto dei rischi che quel click comporta. Pertanto, alla luce del panorama di insegnanti ed alunni sin qui descritto, trattare il Pensiero Computazionale nella Scuola, basandone l’apprendimento solo sull’informatica, non è un’attività di immediata ed automatica implementazione.

Ma allora come fare? È semplice: iniziamo, come vedremo, con il recuperare abilità dimenticate.

3. Il Pensiero Computazionale: cos’è

Il Pensiero Computazionale è un processo di elaborazione mentale, costituito di istruzioni elementari successive non ambigue, che porta alla soluzione di un problema complesso.

Facciamo un esempio. Ci troviamo dinanzi a due mele A e B, una più grande ed una più piccola. Se un interlocutore ci chiede: “Qual è la mela più grande?”, il nostro cervello effettua una serie di operazioni elementari in sequenza, ormai talmente automatiche e scontate che nemmeno ce ne rendiamo conto: a) confronta la mela A con la mela B; b) se $B > A$, sceglie la mela B; c) se $A > B$, sceglie la mela A.

Banale. Ma cosa succederebbe se, anziché tra due mele, ci venisse chiesto di esprimere il nostro giudizio su un cesto da 1000 mele? Beh, il processo di elaborazione mentale resterebbe lo stesso e potremmo descriverlo, in lingua italiana, nel modo seguente: “Confronto le mele a due a due, e scelgo sempre la più grande, finché le mele non sono finite”.

Abbiamo già compreso, dunque, un primo aspetto del Pensiero Computazionale: partendo da un esempio, modellizzare il problema per riuscire a risolvere una categoria di problemi simili. Infatti, volendo risolvere la tipologia di problemi “trova l’elemento più grande in un gruppo di n elementi”, la frase precedente potrebbe essere riscritta così: “Confronta gli elementi a due a due e scegli sempre il maggiore, finché gli elementi non sono finiti”. Siano gli elementi mele, numeri, libri.

L’essere umano apprende questo meccanismo molto presto, da bambino in età

Il Pensiero Computazionale, questo sconosciuto

prescolare, e lo interiorizza, utilizzandolo nella quotidianità dei gesti, finché esso diventa talmente automatico che... lo dimentica! O meglio, gli viene talmente naturale che, se gli viene chiesto di spiegare il processo mentale che lo ha portato al risultato, non sa farlo, o perlomeno non sa farlo con la stessa naturalezza con cui ha fornito il risultato.

Peraltro, dinanzi alla domanda sulle mele, non tutti avrebbero fornito lo stesso procedimento. Ad esempio, si potrebbe pensare di dividere il cesto iniziale in due gruppi, facendo così una prima distinzione tra oggetti grandi e oggetti piccoli, e poi procedere per suddivisioni successive fino ad identificare il più grande.

Ecco chiarito un secondo aspetto del Pensiero Computazionale: svilupparlo, non significa soltanto trovare una soluzione ad un problema, ma anche individuare il procedimento migliore per trovarla. Ma non il migliore in assoluto, bensì il migliore per se stessi.

E' quindi chiaro come il Pensiero Computazionale aiuti fin da piccoli a ragionare di più e a pensare in modo creativo, stimolando la curiosità attraverso quello che apparentemente può sembrare solo un gioco.

Compreso questo, dobbiamo chiederci: cosa c'entra il Pensiero Computazionale con l'Informatica? Beh, c'entra, ed anche parecchio, perché l'unico modo per dare istruzioni ad un calcolatore elettronico è proprio quello di fornirgli "istruzioni elementari successive non ambigue", che è, lo ricorderemo, la definizione di Pensiero Computazionale!

Il MIUR, pertanto, sulla scia della didattica innovativa, perseguendo l'obiettivo dello sviluppo delle competenze digitali, ha pensato bene di prendere due piccioni con una fava: promuovere l'uso dell'informatica nelle scuole come strumento e contestualmente implementare lo sviluppo delle logiche che ne sono alla base e che rappresentano l'obiettivo ultimo dell'intervento proposto. Sia chiaro, quindi, in maniera definitiva, che quando viene proposto, a tutti i livelli, un Corso di Formazione sul Coding (di cui a breve daremo l'esatta definizione), l'obiettivo non è creare schiere di docenti ed alunni super esperti di informatica: l'obiettivo è, e resterà sempre, quello di spingere i partecipanti a risvegliare in sé le sopite attitudini all'uso del Pensiero Computazionale. E poiché esso va attivato trasversalmente ai molteplici aspetti della vita quotidiana e della didattica, abbiamo così dissolto un falso mito che aleggia, spesso, intorno a Coding e Pensiero Computazionale: non sono materie da riservare alle discipline tecnico scientifiche.

Vedremo infatti, nei prossimi paragrafi, applicazioni del Coding ad alcune materie, comprese l'italiano, la storia, l'arte.

4. Il Coding

Il Coding è l'utilizzo didattico di strumenti e metodi di programmazione informatica, estremamente semplificati, in modo da essere accessibili anche ai non esperti, tramite l'introduzione di blocchi logici visuali che si incastrano perfettamente tra loro, in modo da fornire l'esatta sequenza di istruzioni semplici e non ambigue, già citate nella definizione di Pensiero Computazionale, necessarie a far eseguire ad una macchina automatica (un computer, un personaggio virtuale su uno schermo, oppure un robot) una serie di attività, siano esse semplici o molto complesse.

Tornando ad esempio al confronto tra due elementi per determinare quale sia il maggiore, abbiamo utilizzato istintivamente dei blocchi di comando, che rappresentano i cardini del Coding e, in generale, della programmazione informatica:

- Un Operatore Logico di Scelta Condizionata, del tipo “*se...allora...altrimenti*”:
se A > B, allora scegli A, altrimenti scegli B
- Un Operatore “ciclo” del tipo “*ripeti finché*”: *ripeti* la scelta tra due elementi *finché* gli elementi non sono finiti

Scratch, il più famoso software di coding didattico, sviluppato dal MIT (Massachusetts Institute of Technology) nel 2003 ed ormai ampiamente diffuso da tempo anche in Italia, utilizza, per questi due blocchi di comando da impartire ad un personaggio (detto “sprite”), la seguente grafica:



Negli spazi vuoti inglobati dal blocco di comando vanno inseriti i dati relativi allo specifico problema, ma essi sono utilizzabili per risolvere modelli di problemi infiniti.

L'utente è aiutato, nella compilazione delle istruzioni e nella scelta dei blocchi, dalla presenza di dentellature e di forme geometriche, non tutte compatibili tra loro: se due blocchi sono geometricamente compatibili, vuol dire che quelle due istruzioni possono essere inviate consecutivamente alla macchina, che sarà in grado di eseguirle.

Ora che abbiamo capito cosa è il Coding, vale la pena fare una precisazione: abbiamo detto che il Coding è il cavallo di battaglia che il MIUR ha scelto come strumento di divulgazione del Pensiero Computazionale. Ma non è l'unico! Anche imparare una lingua diversa da quella madre ne aiuta lo sviluppo. Sarà per questo motivo, chissà, che una volta si suggeriva lo studio del Greco e del Latino agli alunni più volenterosi? “Il Liceo Classico forgia la *forma mentis*”, era la frase ricorrente degli adulti che suggerivano ai ragazzi il miglior percorso di studi.

Ebbene, quella cosiddetta forma mentis è strettamente imparentata, a quanto sembra, con il Pensiero Computazionale. La logica di formazione delle frasi, basata sulle

desinenze delle declinazioni nelle parole e dei verbi nelle coniugazioni, funziona, nel cervello umano, allo stesso modo degli incastri geometrici dei blocchi visuali, impegnandolo a capire “quale parola è compatibile con quale parola”. Una dinamica di formazione di frasi di senso compiuto, questa, molto ben diversa da quella italiana, basata sulle logiche posizionali: quale studente non ricorderà il giochino, proposto dai professori, basato proprio su questo aspetto: “I Vitelli dei romani sono belli: è italiano o latino? Traducete!”.

Anche lo studio della Musica, altra forma di comunicazione complessa, incentiva lo sviluppo del Pensiero Computazionale. Nell'apprendimento di qualsiasi strumento musicale, le fasi di studio sono le seguenti:

- a) l'allievo impara a riconoscere i suoni;
- b) l'allievo impara a riconoscere la posizione delle note sul pentagramma
- c) l'allievo impara a riconoscere la posizione delle note sullo strumento.

...e con l'esercizio, saprà suonare il DO sullo strumento anche senza leggerlo su pentagramma, esattamente come quando, nella lingua italiana, non ha bisogno di leggere una parola per scriverla, ma basta ascoltarla.

5. Nuove teorie o Sviluppo della Ricerca Pedagogica?

Sarebbe un errore pensare che il Pensiero Computazionale rappresenti una rottura con la tradizionale Pedagogia. Al contrario, esso è la naturale evoluzione del Principio della Zona di Sviluppo Prossimale (ZPS) elaborato da Vygotskij, secondo il quale la distanza tra il livello di sviluppo potenziale e quello attuale di un allievo può essere colmata con l'aiuto di altre persone, che siano adulti o pari dell'alunno con un livello di competenza maggiore. Nella sua tipica forma di didattica laboratoriale, in cui ogni problema viene scomposto in problemi più semplici sotto la guida di un soggetto (il docente) dotato di conoscenza superiore o di pari che si trovano in una zona di sviluppo superiore alla sua (i compagni), il Coding rappresenta proprio la messa in pratica di questo Principio.

Anche le teorie di Piaget trovano buona aderenza applicativa al Pensiero Computazionale: la logica della visuale a blocchi, infatti, serve proprio a questo: saper usare immagini riprodotte (lo sprite di scratch, ad esempio) ed immagini anticipatorie (prevedere il prossimo passo in una sequenza di istruzioni) per arrivare alla soluzione di un problema mediante un proprio percorso, diverso da alunno ad alunno.

Per finire, nel Coding si ripercorrono sempre le tre fasi della teoria dell'apprendimento di Bruner, partendo da quella esecutiva (si affronta il problema reale, nella sua complessità fatta di tante azioni), passando per quella iconica (modellizzazione del problema e scomposizione in sottoproblemi più piccoli) e poi quella simbolica (si eseguono le fasi del processo creato, modellizzato mediante simboli e convenzioni condivise).

6. Il Coding unplugged

Una buona notizia per i docenti che non amano il computer, o per coloro che non ne dispongono a scuola: ci si può esercitare con il Coding anche senza l'ausilio di attrezzature informatiche! Al contrario, lo scrivente ritiene che l'esercizio del Coding offline, o, come viene definito in gergo, "unplugged" (ossia senza l'ausilio dell'elettricità, e quindi senza computer) sia il modo migliore, più divertente e più proficuo per approcciare il pensiero computazionale, nonché il meno traumatico per gli utenti completamente digiuni della materia.

I classici esercizi base di Coding unplugged propongono all'allievo percorsi direzionali per orientarsi su un piano di gioco a caselle (una specie di battaglia navale). Si fornisce all'allievo una serie di istruzioni in sequenza e si chiede allo stesso, partendo da una posizione nota, di indovinare quale sarà la posizione finale. Questa dinamica di gioco può essere vissuta sia in prima persona, costruendo sul pavimento il terreno di gioco e utilizzando l'alunno come "pedina" per i movimenti (modalità molto adatta alla Scuola Primaria), sia su un tabellone da tavolo, o da computer, sul quale far muovere un personaggio (un robot, un segnaposto orientabile, modalità più adatta alla Scuola Secondaria di Primo Grado).

					
E					
D					
C					
B					
A					
	1	2	3	4	5

Molto interessante anche il Coding inverso: si forniscono ad un robot le istruzioni necessarie a raggiungere un traguardo noto, avendo a disposizione solo due comandi: ad esempio, "avanza di una casella" e "gira a sinistra sul tuo posto"; quindi, si chiede all'allievo di scrivere la sequenza corretta di comandi per ottenere il risultato desiderato. Questa modalità è fortemente consigliata nelle situazioni in cui l'obiettivo è sviluppare la progettualità individuale, mentre nel Coding "diretto" si predilige lo sviluppo delle capacità esecutive dell'alunno.

7. Il Coding applicato alle Discipline didattiche: alcuni esempi

Si riportano qui di seguito alcune proposte didattiche, di diversi gradi di difficoltà, per iniziare ad usare il Pensiero Computazionale a Scuola.

a) Italiano: creare la mappa concettuale di una pagina di un libro di testo.

Proposta del docente: riusciamo a scrivere l’algoritmo necessario ad un robot ad ottenere una mappa concettuale, talmente “modellizzato” e “generalizzato” da essere applicabile ad ogni pagina di ogni libro, a prescindere dalla materia?

Soluzione

Istruzione 1: leggi un capoverso, fino al punto

Istruzione 2: sottolinea i concetti più importanti

Istruzione 3: Riporta i concetti più importanti su un foglio bianco, riquadrandoli con blocchi rettangolari

Istruzione 4: Ripeti le istruzioni da 1 a 3 finché i capoversi non sono finiti

Istruzione 5: Collega i blocchi rettangolari tra di loro, utilizzando connettori logici (“conseguenza logica”, “sequenza temporale”, ecc...)

b) Storia: creare un processo che descriva tutti gli eventi relativi ad una guerra

Proposta del docente: proviamo a scrivere un processo che, ordinando cronologicamente tutti gli eventi preliminari alla prima battaglia e poi tutte le battaglie, descriva compiutamente le fasi di una guerra.

Soluzione

Istruzione 1: evidenzia tutte le date relative agli eventi riportate nell’argomento

Istruzione 2: ordina le date in modo crescente

Istruzione 3: associa ad ogni data una breve descrizione dell’evento associato

Istruzione 4: elimina gli eventi di importanza minore

c) Arte e Immagine: Pixel Art. La Pixel Art è una tecnica per costruire immagini che segue le orme della Corrente del Puntinismo, sostituendo, però, al classico puntino realizzato con un pennello, il pixel, un quadretto colorato, seguendo istruzioni successive tipiche degli algoritmi informatici in codice binario. Si riporta qui di seguito un esempio semplice, in cui, fornendo all’alunno, riga per riga, il numero e la sequenza di quadretti da riempire con un determinato colore, si ottengono due immagini sul tema dell’autunno.

Proposta del docente:

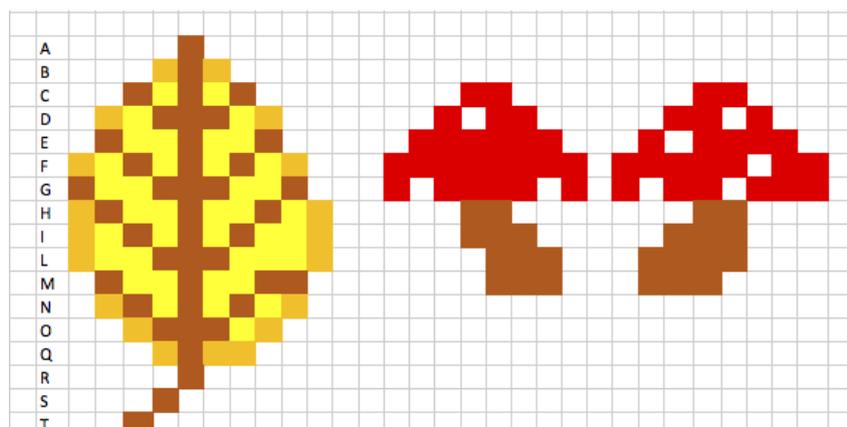
A – 4Bianchi, 1 marrone

B – 3 bianchi, 1 arancione, 1 marrone, 1 arancione

C – due bianchi, 1 marrone, 1 giallo, 1 marrone, 1 giallo, 1 marrone, 8 bianchi, due rossi, 7 bianchi, 2 rossi

D – ecc....

Soluzione



Le applicazioni, come abbiamo visto, sono innumerevoli. Anche la Grammatica, che può sembrare la materia più complessa da piegare alle regole del Pensiero Computazionale, può offrire invece spunti di esercitazione: ad esempio, si può chiedere agli allievi di creare un algoritmo che sia in grado di estrarre da un testo tutte le proposizioni subordinate, da elencare in una tabella, associando ad ognuna la tipologia di proposizione (condizionale, consecutiva, ecc ...)!

Persino la vita quotidiana può servire da palestra di allenamento: chiediamo ai nostri alunni di creare, ad esempio, una lista di istruzioni da far eseguire ad un robot per inserire nello zaino tutto il materiale scolastico da portare a scuola il giorno seguente.

Attualmente, purtroppo, non abbiamo in dotazione robot del genere, ma, magari, insegnando ai nostri bambini a pensare in modo logico ed organizzato, saranno proprio essi ad inventarne uno! Alla peggio, avranno imparato a non dimenticare a casa gli strumenti scolastici. E sarà stata comunque una vittoria!

8. Potenziali sviluppi e Conclusioni

Quali possono essere gli sviluppi del Coding? Quale la frontiera successiva da affrontare per stare al passo con la costante necessità di potenziare il Pensiero Computazionale, una volta che la Scuola avrà digerito questo primo step, ed il Coding sarà diventato un argomento di routine?

In realtà, già in alcuni Istituti italiani la Robotica ha fatto il suo timido ingresso, e sul mercato sono già disponibili diversi kit di assemblaggio che permettono di costruire scariate tipologie di robot, completi di tutti i dispositivi, seppur in via semplificata, necessari a “dare vita” alla macchina, quali attuatori elettrici, sensori di movimento, sensori di prossimità, unità di controllo, ecc...E, nel kit, è anche compreso il software necessario a programmare il robot appena assemblato per comunicargli le istruzioni da eseguire. In sostanza, si passa dalla programmazione di un soggetto virtuale, come lo sprite di Scratch, alla programmazione di un oggetto reale, che si muove in uno spazio fisico e interagisce con esso, ad esempio, cambiando direzione quando avverte un

Il Pensiero Computazionale, questo sconosciuto

ostacolo dinanzi al suo percorso, sollevando un braccio meccanico quando avverte un peso, invertendo la direzione di marcia dopo un tempo prestabilito, e così via.



Anche il Tinkering (dall'inglese "arrangiarsi", "arrabattarsi") ha già qualche cultore tra i docenti italiani. La Tinkering School è nata nel 2005 all'Exploratorium di San Francisco, ed è una scuola che spinge ad imparare facendo, nella maggior parte dei casi giocando, e non viene richiesto uno studio classico o teorico approfondito, ma una buona capacità logica e manuale. Lo scopo del Tinkering è costruire oggetti (anche robotici) con oggetti di partenza completamente diversi, eventualmente provenienti dal riciclaggio di parti di macchine o suppellettili di tutt'altro iniziale utilizzo, con lo scopo non solo di sviluppare la creatività e l'imprenditorialità degli allievi, ma anche di sensibilizzarli al riuso e alla riconversione dei prodotti che sono arrivati a fine ciclo vita, in favore di uno sviluppo ecosostenibile e di un impatto ambientale ridotto.

Facile, pertanto, ipotizzare che, una volta che l'alunno avrà imparato come è fatto un robot, potrà passare dall'assemblaggio guidato dello stesso mediante un kit predefinito, alla realizzazione di un robot realizzato "da zero" con pezzi meccanici a sua scelta e/o con materiali di uso comune (cartone, bottiglie di plastica, assi di legno, ecc...).

Come non citare, infine, i droni, tanto richiesti sugli scaffali della grande distribuzione organizzata. Grazie alla visuale aerea che queste macchine riescono ad offrire, e che prima potevano essere solo ammirate in televisione, oggi è possibile approcciare alla didattica in modalità molto più concreta.



Si pensi, ad esempio, a come potrà cambiare una lezione per la Scuola Primaria, quando si dovrà affrontare il tema del territorio e dell'ambiente: come è fatta la nostra città?

Quanti metri è alta la nostra scuola? Che temperatura c'è a 20 metri di altezza? Cos'è la forza di gravità? Click, e grazie ai molteplici sensori di cui un drone è dotato, e grazie al fatto che esso sarà pilotabile comodamente dall'aula, grazie ad una lista di istruzioni dettate con un codice visuale a blocchi, tutte queste domande avranno una risposta molto più tangibile rispetto a quella ricavabile da un libro di testo.

Siamo alle conclusioni. Arrestare la divulgazione del Pensiero Computazionale non è possibile, oltre ad essere controproducente, in virtù del fatto che le metodologie e le tecnologie didattiche che si possono mettere in campo a seguito dell'acquisizione di questa competenza sono talmente efficaci ed efficienti che sarebbe davvero una grave perdita, per tutti, non utilizzarle: per gli alunni, perché consentono un apprendimento più agevole ed inclusivo, nonché più accattivante e meno noioso, basato sul "fare" e sul problem solving, e per i docenti, per i quali, a fronte di un piccolo sforzo nell'implementazione del proprio aggiornamento professionale, faranno meno fatica nella divulgazione delle discipline ministeriali.

Non resta, quindi, che armarsi della solita pazienza, unita ad una sana curiosità per l'innovazione didattica, qualità che da sempre caratterizzano il docente italiano, e affrontare di petto, seppur gradualmente e passo dopo passo, questa nuova competenza del millennio, che promette di essere una delle più concrete e risolutive della storia della Scuola Italiana.

Buon Pensiero Computazionale a tutti!

Bibliografia

- Circolare MIUR Prot. N. 9759 del 08/10/2015, avente ad oggetto: *"Il pensiero computazionale a scuola – al via il secondo anno dell'iniziativa "Programma il Futuro": insegnare in maniera semplice ed efficace le basi dell'informatica"*.
- *Piano Nazionale Scuola Digitale*, introdotto dalla Legge n. 107 / 2015
- Paolo Ferragina, Fabrizio Luccio, 2017, *"Il pensiero computazionale. Dagli algoritmi al coding"*, Ed. Il Mulino
- Alessandro Bogliolo, 2016, *"Coding in your Classroom, Now!"*, Ed. Giunti.
- Olga Liverta Sempio, 1998, *"Vygotskij, Piaget, Bruner. Concezioni dello sviluppo"*, Raffaello Cortina Editore.

Sitografia

- <https://scratch.mit.edu>
- www.code.org
- <https://www.tinkeringschool.com>