

Coding, pensiero computazionale e informatica: quale evoluzione possibile?

Michael Lodi and Simone Martini

Dipartimento di Informatica - Scienza e Ingegneria
Alma Mater Studiorum - Università di Bologna
Bologna, Italia
Laboratorio CINI Informatica e Scuola
{michael.lodi, simone.martini}@unibo.it

Abstract

Analizziamo, dal punto di vista dell'informatica e della didattica dell'informatica, l'espressione "coding, pensiero computazionale e informatica" utilizzata in recenti documenti ministeriali e più in generale per parlare dell'introduzione dell'informatica a scuola.

1 Introduzione

"Coding, pensiero computazionale e informatica: quale evoluzione possibile?", così recita il titolo dell'ultimo paragrafo delle "Linee Guida STEM" [1] emanate a ottobre 2023, che mirano, anche nel contesto del PNRR, a integrare nell'offerta formativa delle scuole azioni per potenziare le competenze matematico-scientifico-tecnologiche e digitali.

Negli ultimi anni queste espressioni sono entrate a far parte, oltre che del linguaggio della scuola e della formazione docenti, anche di alcune leggi e decreti. Tuttavia sono spesso usate senza essere definite e possono essere fraintese dai non addetti ai lavori. Inoltre, esistono interpretazioni diverse anche tra gli esperti.

In questo articolo, forniamo il nostro punto di vista da informatici e ricercatori in didattica dell'informatica, indicando come naturale evoluzione il pieno riconoscimento dell'informatica come disciplina autonoma e la sua introduzione nei curricula scolastici, con la stessa dignità delle altre discipline scientifiche.

Le nostre riflessioni potranno essere utili nel valutare quanto un'attività didattica possa essere considerata di "coding, pensiero computazionale e informatica".

2 Informatica

Chiedendo a gruppi di studenti di diverse età che cos'è la biologia, moltissimi risponderanno correttamente che è la scienza che studia la vita. Lo stesso con l'astronomia: la scienza che studia i corpi celesti e il cosmo. Per l'informatica, qualcuno dirà che studia i computer. Nessuno dirà che è una scienza. Ma, come suggerito dall'etimologia (*informat(ion)* e (*automat)ique*) e semplificando la definizione di Denning [9], potremmo dire che l'informatica è la scienza che studia l'elaborazione (descrizione, memorizzazione, trasmissione...) automatica dell'informazione. Si tratta certo di una scienza, e di una disciplina, giovane, ma riconosciuta e autonoma (di informatica troviamo dipartimenti universitari, corsi di laurea, materie scolastiche, titoli di professori universitari e docenti).

3 Coding

Nel mondo delle professioni informatiche, *coding* viene sempre più usato come sinonimo di *programmazione* (es. si trovano annunci di lavoro che parlano di *coders* e *coding languages* invece di *programmers* e *programming languages*).

Nell'ingegneria del software *coding* indica solo una delle varie fasi dello sviluppo software (insieme ad analisi dei requisiti, design, test, ...): la più noiosa e meccanica, quella in cui si "pigiano" materialmente i tasti sulla tastiera per scrivere codice e quindi implementare il programma.

Il termine viene usato, più o meno formalmente, anche in altre aree dell'informatica: nel senso di "codifica" dell'informazione (più propriamente *encoding*) o cifratura (più propriamente *encrypting*).

Tuttavia, di recente è emersa una accezione didattica: *coding* è la "descrizione più giocosa e non intimidatoria della programmazione per principianti" [18, traduzione nostra]. In Italia, in particolare, si è diffuso l'uso secondo cui "il coding (in quest'ambito) è l'applicazione intuitiva di principi, metodi e strumenti di programmazione al fine di sviluppare il pensiero computazionale" [5], spesso inteso come metodologia didattica trasversale.

Questa recente accezione didattica spiega anche espressioni che possono apparire ossimoriche, come "coding unplugged". Nella didattica dell'informatica si parla di attività unplugged come di "attività cinesesiche e divertenti per insegnare i concetti di informatica senza computer" [6, 7, nostra traduzione]), cioè che fanno uso di materiali facilmente reperibili in classe o di giochi in giardino. Tradizionalmente, queste attività non si concentrano sulla programmazione, ma su altri concetti informatici quali algoritmi notevoli, strutture dati, codifica (*encoding*) dell'informazione, e così via. Se si mantiene l'accezione informatica (e non quella didattica) di *coding* come sottoinsieme della – o equivalente alla – programmazione, non è pensabile che sia svolto *unplugged*. L'espressione "coding unplugged" si riferisce dunque in realtà ad attività unplugged per introdurre concetti propedeutici alla programmazione, quali introduzione agli algoritmi, strutture di controllo espresse in pseudocodice, e così via.

L'espressione "programmazione informatica (coding)" è anche presente nella legge 159/19 e nel Decreto Interministeriale 1612 del 28/10/21, che l'hanno resa parte della formazione degli insegnanti di ogni ordine, grado e disciplina. Nella legge 233/21 (PNRR), art. 24-bis, si indica inoltre che la "programmazione informatica (coding)" dovrà essere introdotta nell'ambito degli insegnamenti già esistenti nelle scuole di ogni ordine e grado, anche tramite l'aggiornamento di Indicazioni Nazionali e Linee Guida. Sebbene la formulazione suggerisca che si stia usando *coding* come sinonimo di programmazione informatica, questo non viene mai esplicitato. Ma leggendo il dibattito che ha portato a diverse mozioni approvate l'11 marzo 2019 [2], che hanno impegnato il governo proprio a emanare le leggi di cui sopra, si può chiaramente dedurre l'uso di *coding* con accezione di metodologia didattica trasversale.

Le linee guida STEM utilizzano "programmazione informatica (coding)" quando si riferiscono ai documenti precedentemente citati, introducono l'espressione *coding unplugged* già discussa, oppure usano inscindibilmente l'espressione "coding, pensiero computazionale e informatica".

Tuttavia, l'Accademia della Crusca sconsiglia l'uso di questa terminologia per supportare l'introduzione dell'informatica a scuola. Discutendo della traduzione di *coding* suggerisce che

Codifica risulterebbe quindi un traducevole più legittimo. In realtà, se tale traducevole è adatto in ambito informatico, non sembra possibile applicarlo all'insegnamento della materia scolastica. Quando si parla di *coding* a scuola, infatti, si fa riferimento alle attività di introduzione all'informatica. [...] In ambito scolastico, sarebbe dunque più adeguato [...] favorire un sostantivo italiano che è sicuramente noto, trasparente e facilmente applicabile alle diverse situazioni: l'insegnamento dell'*informatica*. [11, p. 100, corsivo originale]

e poi conclude che

sembra più opportuno seguire la strada intrapresa dagli istituti superiori ed estendere tale proposta alle scuole di ogni ordine e grado; si potrebbe quindi chiamare la materia con una delle seguenti denominazioni: *principi di informatica*, *basi di informatica* o *fondamenti di informatica*. [11, p. 101, corsivo originale]

4 Pensiero Computazionale

Come già sottolineato, l'espressione *pensiero computazionale* è utilizzata negli ultimi anni in abbinamento a *coding* per parlare dell'introduzione dell'informatica nella scuola. Sebbene l'espressione sia stata utilizzata già da Papert nel 1980 (per una prospettiva storica si vedano [16, 17]), il pensiero computazionale è stato definito da Wing prima informalmente nel 2006 come “pensare come un informatico per risolvere problemi” [19] e poi come “il processo mentale coinvolto nella formulazione dei problemi e delle loro soluzioni, di modo che tali soluzioni siano rappresentate in una forma che possa essere eseguita in modo effettivo da un agente che processa informazioni” [20, traduzione nostra].

Ancora una volta l'accademia della Crusca conferma questa interpretazione:

Potremmo quindi considerare il *pensiero computazionale* come una modalità di pensiero che coinvolge particolari facoltà cognitive umane e che prevede uno specifico approccio alla formulazione e risoluzione di problemi complessi, utile alla comprensione della società contemporanea, consentendo di analizzare ed elaborare dati con metodi e strumenti che si acquisiscono con lo studio dell'informatica. [11, p. 95, corsivo originale]

L'espressione è già presente (non definita) nella legge 107/15 (La “Buona Scuola”), e nel D.M. 741/17 sull'Esame di Stato conclusivo del primo ciclo di istruzione. La definizione presente nel documento “Indicazioni Nazionali e Nuovi scenari” [3], che va ad integrare le Indicazioni Nazionali per il curriculum della scuola dell'infanzia e del primo ciclo di istruzione, definisce *pensiero computazionale* come “un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia”. Il documento omette il collegamento diretto con la disciplina scientifica sottostante (i metodi e gli strumenti sono specifici *dell'informatica*) e colloca il pensiero computazionale tra le competenze trasversali alle altre discipline. In tempi recenti inoltre l'interpretazione restrittiva del termine *computazionale* come “legato al calcolo” [13, pp. 2-3] e l'inclusione del *computational thinking* tra le competenze di matematica valutate nell'OCSE PISA [4] ha avvicinato il concetto alla matematica, allontanandolo dall'informatica.

In altre discipline è normale parlare di “pensiero disciplinare” (es. *pensiero matematico*, *ragionamento matematico*, *problem solving matematico*). Come riconosciuto da Li e colleghi [14], i “pensieri disciplinari” contengono sia elementi specifici della disciplina sia elementi più trasversali.

Purtroppo notiamo [15] la tendenza di insegnanti e policymaker a focalizzarsi solo sugli aspetti trasversali (come creatività, collaborazione, perseveranza e tolleranza per l'ambiguità) o su aspetti più specifici ma letti in chiave di problem solving generale (scomposizione di problemi, astrazione, approcci iterativi...). Sebbene è ovvio che questi aspetti, facenti parte delle complesse abilità di problem solving umane, siano condivisi tra diverse discipline, non dobbiamo dimenticare che l'informatica, come ogni altra disciplina, apporta contributi specifici che istanziano queste skill (es. l'astrazione è molto diversa in informatica, in matematica e in fisica [10, p. 21]) e formano il suo modo di pensare, il “pensare informatico”.

Uno tra tutti: il problem solving di cui parla Wing è il cosiddetto *problem solving computazionale* (da intendersi non come “legato al calcolo” ma “legato all'elaborazione automatica dell'informazione” – *problem solving informatico*). Uno degli elementi caratterizzanti è la presenza di un interprete (esterno, automatico) con un insieme limitato di istruzioni che comprende e che può eseguire effettivamente, per risolvere problemi espressi in un linguaggio non ambiguo [12, 8].

5 Evoluzione possibile

Da quanto detto dovrebbe risultare evidente la nostra posizione: esiste la disciplina dell'informatica, fondamentale per comprendere ed essere attori protagonisti nella società digitale in cui viviamo, e quindi è questa disciplina che dovrebbe essere imparata e insegnata a scuola, a tutti i livelli. Lo studio dell'informatica (così come fanno le altre discipline) permette di acquisire un particolare modo di pensare disciplinare, il “pensiero computazionale”, che permette di risolvere problemi di elaborazione automatica dell'informazione. Il linguaggio della scienza informatica è la programmazione (volgarmente detta coding), e per questo – sebbene non sia l'unico aspetto – introdurla è fondamentale per lo studio di tale disciplina.

Chiamare “coding” o “informatica” attività che non includano, insieme alle cruciali componenti di competenze trasversali, anche gli aspetti disciplinari fondamentali – quale quello di interprete – danneggia la battaglia per il riconoscimento dell'informatica come disciplina da insegnare a scuola e preclude agli studenti la possibilità di impararne i principi fondamentali.

Fondi

Il lavoro di M. Lodi è supportato dallo Spoke 1 “FutureHPC & BigData” del Centro Nazionale di Ricerca in “High Performance Computing, Big Data and Quantum Computing” (ICSC) finanziato da MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S (M4C2-19) - Next Generation EU (NGEU).

Il lavoro di S. Martini è parzialmente finanziato da: (a) SERICS (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; (b) Italian PRIN 2020 2022BP7K3 Learning informatics; (c) INdAM GNSAGA.

Bibliografia

- [1] URL: <https://www.miur.gov.it/documents/20182/0/Linee+guida+STEM.pdf/2aa0b11f-7609-66ac-3fd8-2c6a03c80f77?version=1.0&t=1698173043586> (visitato il giorno 13/07/2024).
- [2] URL: <https://www.cs.unibo.it/~michael.lodi2/mozionecoding-annotata.pdf> (visitato il giorno 13/07/2024).
- [3] URL: <https://www.miur.gov.it/documents/20182/0/Indicazioni+nazionali+e+nuovi+scenari/> (visitato il giorno 13/07/2024).
- [4] URL: <https://pisa2022-maths.oecd.org/ca/index.html#Interpret> (visitato il giorno 13/07/2024).
- [5] Alessandro Bogliolo. *Un modello per il coding a scuola*. 2016. URL: <https://codemooc.org/un-modello-per-il-coding-a-scuola/> (visitato il giorno 13/07/2024).
- [6] Tim Bell e Michael Lodi. «Constructing Computational Thinking Without Using Computers». In: *Constructivist Foundations* 14.3 (2019), pp. 342–351. URL: <https://constructivist.info/14/3/342.bell>.
- [7] Tim Bell et al. «Computer Science Unplugged: school students doing real computing without computers». English. In: *New Zealand Journal of Applied Computing and Information Technology* 13.1 (2009), pp. 20–29. ISSN: 1174-0175.

- [8] Carlo Belletini et al. «To Be Or Not To Be . . . An Algorithm: The Notion According to Students and Teachers». In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 102–108. ISBN: 9798400704239. DOI: [10.1145/3626252.3630950](https://doi.org/10.1145/3626252.3630950). URL: <https://doi.org/10.1145/3626252.3630950>.
- [9] Peter J. Denning et al. «Computing as a discipline». In: *Communications of the ACM* 32.1 (feb. 1989), pp. 9–23. DOI: [10.1145/63238.63239](https://doi.org/10.1145/63238.63239).
- [10] Andrea A. diSessa. «Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education». In: *Mathematical Thinking and Learning* 20.1 (2018), pp. 3–31. DOI: [10.1080/10986065.2018.1403544](https://doi.org/10.1080/10986065.2018.1403544). URL: <https://doi.org/10.1080/10986065.2018.1403544>.
- [11] Lucia Francalanci. «Coding». In: *Italiano digitale* 12.1 (mar. 2020), pp. 95–101. ISSN: 2532-9006. DOI: [10.35948/2532-9006/2020.3311](https://doi.org/10.35948/2532-9006/2020.3311). URL: <http://dx.doi.org/10.35948/2532-9006/2020.3311>.
- [12] Robert R. Korfhage. «Algorithm». In: *Encyclopedia of Computer Science*. GBR: John Wiley e Sons Ltd., 2003, pp. 36–38. ISBN: 0470864125.
- [13] Yeping Li et al. «Computational Thinking Is More about Thinking than Computing». In: *Journal for STEM Education Research* 3.1 (apr. 2020), pp. 1–18. ISSN: 2520-8713. DOI: [10.1007/s41979-020-00030-2](https://doi.org/10.1007/s41979-020-00030-2). URL: <http://dx.doi.org/10.1007/s41979-020-00030-2>.
- [14] Yeping Li et al. «On Thinking and STEM Education». In: *Journal for STEM Education Research* 2.1 (feb. 2019), pp. 1–13. ISSN: 2520-8713. DOI: [10.1007/s41979-019-00014-x](https://doi.org/10.1007/s41979-019-00014-x). URL: <http://dx.doi.org/10.1007/s41979-019-00014-x>.
- [15] Michael Lodi. «Informatical Thinking». In: *OLYMPIADS IN INFORMATICS* (dic. 2020), pp. 113–132. ISSN: 1822-7732. DOI: [10.15388/ioi.2020.09](https://doi.org/10.15388/ioi.2020.09). URL: <http://dx.doi.org/10.15388/ioi.2020.09>.
- [16] Michael Lodi. «Pensiero Computazionale: dalle “scuole di samba della computazione” ai CoderDojo». In: *Atti del convegno DIDAMATICA 2018*. In Italian. Cesena, Italy: AICA, apr. 2018. URL: https://www.aicanet.it/documents/10776/2101882/didamatica2018_paper_57.pdf.
- [17] Michael Lodi e Simone Martini. «Computational Thinking, Between Papert and Wing». In: *Science & Education* 30.4 (apr. 2021), pp. 883–908. ISSN: 1573-1901. DOI: [10.1007/s11191-021-00202-5](https://doi.org/10.1007/s11191-021-00202-5). URL: <http://dx.doi.org/10.1007/s11191-021-00202-5>.
- [18] Kiki Prottzman. *Coding vs. Programming - Battle of the Terms!* 2015. URL: https://www.huffpost.com/entry/coding-vs-programming-bat_b_7042816 (visitato il giorno 13/07/2024).
- [19] Jeannette M. Wing. «Computational thinking». In: *Communications of the ACM* 49.3 (mar. 2006), p. 33. DOI: [10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215).
- [20] Jeannette M. Wing. «Research Notebook: Computational Thinking—What and Why?». In: *The Link Magazine* (2011). URL: <https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>.