

Violetta Lonati *

Sull'informatica come scienza e sul perché insegnarla nella scuola del primo ciclo

L'informatica non ha per oggetto i computer, proprio come l'astronomia non ha per oggetto i telescopi.

Edsger Dijkstra, 1970 (attribuzione incerta)

Nel dibattito sulla scuola spesso si sente parlare di “tecnologie digitali”, del loro ruolo nell'apprendimento, dell'importanza di sviluppare competenze digitali fin dai primi anni di formazione; altrettanto spesso, queste tecnologie vengono associate all'informatica. Ciò si riscontra anche nei documenti ministeriali, in cui espressioni come “competenze digitali”, “*coding*”, “pensiero computazionale” e “informatica” appaiono quasi come interscambiabili e vengono usate in maniera impropria.

Con questo contributo si intende chiarire innanzitutto perché la tendenza a ridurre a sinonimi queste espressioni sia fuorviante, se non addirittura dannosa, e svelare l'informatica in quella che è invece la sua accezione più alta, di disciplina scientifica che studia i principi e i metodi per l'elaborazione automatica delle informazioni.

Infatti, al di là di ciò che le nuove tecnologie ci consentono di fare negli ambiti più disparati e del loro impatto economico, sociale, o nella vita di tutti i giorni, il contributo culturale offerto dall'informatica risulta troppo spesso sottovalutato o comunque debolmente percepito. Al contrario, l'informatica è una materia affascinante che (beninteso, assieme alle altre) può fornire peculiari strumenti concettuali e modelli interpretativi della realtà, i quali a buon diritto dovrebbero entrare a far parte del patrimonio cognitivo di ogni cittadino.

Ne consegue dunque l'importanza di proporre agli alunni, a partire dal primo ciclo di istruzione e con i dovuti adattamenti richiesti dai diversi contesti, i principi e i concetti fondanti dell'informatica. Quali siano questi fondamenti e in che modo sia possibile proporli a scuola sono le domande a cui si cercherà di dare risposta di seguito.

* Ricercatrice presso il Dipartimento di Informatica dell'Università degli Studi di Milano.

Cosa informatica non è

Prima di parlare di informatica nella sua accezione di disciplina scientifica, è necessario chiarire l'ambiguità del termine, cominciando a dire cosa informatica non è.

Il termine “informatica” è diventato una categoria-ombrello usata per racchiudere cose molto diverse. Ad esempio, si parla di informatica in ambito economico-produttivo (le aziende che utilizzano determinati strumenti e tecnologie), oppure parlando dell'uso del *personal computer* o degli apparecchi *post-PC* come *smartphone* e *tablet*. Saper usare le tecnologie digitali non vuol dire conoscere l'informatica; in un laboratorio scolastico attrezzato con *computer* non si fa necessariamente informatica; dotare ogni studente di un *tablet* non può essere descritto come “potenziamento informatico”; saper installare programmi, configurare dispositivi, usare le applicazioni, finanche riparare i computer, non fa di una persona un informatico.

Questa identificazione tra la disciplina scientifica e i suoi strumenti (ben evocata dall'epigrafe all'inizio dell'articolo) accade anche in altre materie: ad esempio, non è raro vedere la matematica ridotta a mero “far di conto”. Tuttavia questa confusione sembra affliggere in particolare l'informatica: se entrate in libreria, nel settore “Scienze” trovate saggi di divulgazione scientifica sulla matematica, l'astronomia, le scienze naturali, ecc; c'è poi di solito un settore “Informatica” in cui si trovano manuali per imparare a usare i *computer* e i loro programmi o, nel migliore dei casi, trattati sull'impatto sociale di questi strumenti. Ancora, se volete comprare una bilancia di certo non cercherete nei negozi un reparto “Fisica e Chimica”; invece, i *computer* li trovate negli scaffali marcati “Informatica”. E si potrebbero fare ancora moltissimi altri esempi.

In tutti i contesti sopraccitati, sarebbe più opportuno parlare, invece, di dispositivi e competenze digitali, per riferirsi all'utilizzo di queste tecnologie e al loro impatto nella vita di tutti i giorni, nella società o nell'ambito economico-produttivo o lavorativo.

L'informatica come disciplina scientifica

A differenza di quanto descritto sopra, l'informatica può essere più propriamente definita come *la scienza che studia i principi e i metodi per l'elaborazione automatica delle informazioni*. Vediamo di chiarire i termini usati e approfondire i concetti implicati da questa definizione.

- Usiamo la parola **informazione** per riferirci a qualcosa che prima non si sapeva, qualcosa che aumenta la nostra conoscenza. Un'informazione prende forma e/o si comunica tramite *dati*, che di per sé non sono informativi, ma lo diventano se interpretati da chi li riceve, anche in funzione delle sue conoscenze pregresse. Per poter essere elaborati, i dati vanno rappresentati in qualche forma; in particolare quando i dati sono

- espressi usando dei simboli (ad esempio cifre o lettere), si parla di *representazione digitale* dell'informazione, o di *codifica* dell'informazione.
- L'aggettivo **automatica** si riferisce al fatto che l'elaborazione di questi dati, rappresentati opportunamente in forma digitale, avviene ad opera di un *esecutore* automatico (chiamato anche *interprete*). Questo agisce sui dati sulla base di istruzioni impartite precedentemente tramite dei programmi. L'esecuzione dei programmi si svolge in maniera determinata e ripetibile e, affinché questo possa avvenire, la descrizione delle azioni che devono essere svolte deve essere a sua volta rigorosa, precisa e non ambigua, cosa che sarebbe impossibile usando il linguaggio naturale (e che invece si può ottenere usando un linguaggio di programmazione).
 - L'ultimo concetto è quello di **elaborazione**. Qui la chiave sono gli *algoritmi*, ovvero i metodi che, elaborando dati, consentono di ricavarne nuova informazione, risolvere problemi, svolgere compiti. Un approccio molto diffuso esemplifica l'algoritmo con una ricetta di cucina, nella quale si descrive come combinare certi ingredienti per ottenere una pietanza. Pur essendo molto evocativa, questa, come tutte le analogie, è valida solo fino a un certo punto. Una delle criticità fondamentali sta nel fatto che l'algoritmo per essere tale ha bisogno di essere descritto in maniera estremamente precisa, cosa che sarebbe impossibile senza fare riferimento a un interprete automatico.

In breve: i dati sono portatori di informazione, purché siamo in grado di raccogliarli, memorizzarli, analizzarli, trasformarli. In un'unica parola: elaborarli. Il processo è particolarmente interessante, fertile ed efficiente se l'elaborazione avviene in maniera automatica, cioè tramite interpreti "meccanici". Ma non basta avere un interprete meccanico per elaborare i dati, serve anche (o soprattutto) conoscere e sviluppare principi e metodi su cui basare l'elaborazione. Di questo si occupa l'informatica!

È importante osservare che, rispetto alle altre discipline, l'informatica propone un punto di vista complementare, che poi è la sua caratteristica distintiva. Come altre discipline scientifiche anch'essa si pone l'obiettivo di trattare nuovi problemi, e quindi svolgere un'attività di *problem solving*. La sua peculiarità sta nel modo con cui i problemi vengono affrontati: l'informatica non si accontenta di trovare una soluzione o indicare un metodo per risolvere un problema (come fa, ad esempio, la matematica), affronta invece i problemi creando dei sistemi (programmi, dispositivi) che ne automatizzano la risoluzione, in modo poi da poterla affidare a degli agenti automatici (privi di intelligenza) esterni, terzi o comunque indipendenti da chi ha progettato o realizzato il sistema stesso. In questo processo di automatizzazione è inclusa anche la necessità preliminare di codificare, ovvero rappresentare digitalmente, le informazioni

necessarie a risolvere il problema o svolgere il compito. Come ben spiegato da Lodi, Martini e Nardelli¹:

È proprio l'aver messo l'esecutore al centro del proprio approccio a far sì che l'informatica apporti, rispetto alla matematica, un mutamento radicale di punto di vista. Il cambiamento di paradigma concettuale è costituito dal passaggio dal risolvere i problemi al far risolvere i problemi. Questa transizione [...] colloca l'informatica come disciplina a sé stante tra le altre scienze.

La programmazione è dunque un ambito centrale dell'informatica, tuttavia sarebbe sbagliato equiparare i due termini. Per “programmazione” si intende l'insieme delle attività di ideazione e realizzazione di un programma o di un'applicazione in grado di svolgere un certo tipo di compito (o risolvere un certo tipo problema). Tra queste attività sono incluse anche l'impostazione dell'algoritmo che porta allo svolgimento del compito o alla risoluzione del problema, e la sua scrittura in un linguaggio di programmazione specifico. Si tratta di un'attività complessa, articolata, sfaccettata, che richiede sia capacità logico/analitiche, che competenze creative e di progettazione. Al termine “programmazione” è spesso preferita, soprattutto in ambito scolastico, la parola “*Coding*”². In effetti, le attività di *coding* si focalizzano però sull'atto di scrittura di programmi secondo una data sintassi (spesso “visuale”, più che linguistico-formale) che dunque rischia di risultare meccanica, nonché riduttiva rispetto all'attività così ricca, complessa e creativa che è la programmazione³.

Perché insegnare i principi dell'informatica

La possibilità di elaborare informazioni per via automatica, come descritto nella sezione precedente, è onnipresente e ha effetti pervasivi nella nostra quotidianità. Per non subirla (ed eventualmente contribuire allo sviluppo digitale) non basta acquisire capacità operative, che oltretutto rischiano di diventare velocemente obsolete: ciò che sarà normale usare in un futuro prossimo sarà probabilmente molto diverso da ciò che è normale adesso. È invece fondamentale, cominciando fin da piccoli, sviluppare *competenze informatiche* e modelli interpretativi che permettano di comprendere e di orientarsi su ciò che riguarda l'elaborazione automatica delle informazioni. Queste “competenze informatiche” sono soprattutto concettuali e cognitive, dunque si parla di qualcosa di molto diverso dalle “competenze digitali”. Fare informatica a scuola deve mirare allo sviluppo di queste competenze informatiche e di questi modelli interpretativi.

¹ LODI M., MARTINI S. e NARDELLI E., *Abbiamo davvero bisogno del pensiero computazionale?*, Mondo Digitale, AICA, 2017, pp. 1-15.

² Si veda su questo l'articolo di GABBARI M., GAGLIARDI R., GAETANO A., SACCHI D., *Integrare “Coding e Pensiero computazionale” nella didattica*, su questo stesso numero della rivista OPPIinformazioni.

³ LONATI V., MALCHIODI D., MONGA M. e MORPURGO A., *Is coding the way to go?*, in BRODNIK A., VAHRENDHOLD J. (a cura di), *Proceedings of ISSEP 2015 - 8th International conference on informatics in Schools: Situation, Evolution, and Perspective*, Lecture Notes in Computer Science 9378, Springer, 2015, pp. 165-174.

In altre parole, fare informatica consentirebbe di sviluppare quelle capacità che a volte si trovano riassunte nell'espressione "pensiero computazionale", coniata da Seymour Papert nel 1980 nel suo celebre libro *Mindstorms*⁴ e resa popolare, anche se in un'accezione abbastanza diversa, da Jeanett Wing nel 2006⁵.

Benché non esista tra la comunità accademica una definizione unanimemente accettata di "pensiero computazionale", qui lo si può definire come *l'insieme delle abilità cognitive e dei processi mentali che mette in atto un informatico nella sua tipica attività di problem solving*. In particolare, il pensiero computazionale include le seguenti capacità che, benché tipiche di questa disciplina, hanno anche una chiara valenza trasversale⁶:

- formulare i problemi in modo che possano essere risolti in maniera automatica da agenti autonomi,
- organizzare e analizzare logicamente le informazioni,
- rappresentarle attraverso modelli e astrazioni,
- automatizzare lo svolgimento di compiti tramite sequenze di passi ordinati (gli algoritmi),
- identificare e analizzare possibili soluzioni algoritmiche usando la migliore combinazione di passi e risorse,
- implementare gli algoritmi con linguaggi di programmazione che la macchina/l'interprete automatico possa comprendere,
- generalizzare e trasferire processi risolutivi a una grande varietà di situazioni diverse.

Si noti che avere competenze informatiche non è importante solo per chi vorrà intraprendere una carriera in questo settore. Consideriamo ad esempio il contesto della pubblica amministrazione o dei grandi enti (come banche, assicurazioni, compagnie aeree e via dicendo): sempre più spesso, l'interazione umana con l'operatore allo sportello o via telefono è sostituita da un sistema automatico che, per sua stessa natura, non sa gestire il caso particolare imprevisto (nel senso letterale di "non previsto") con la flessibilità, il buon senso o la creatività che potrebbe avere un operatore umano gentile e volenteroso. Il sistema automatico che sostituisce queste interazioni è stato presumibilmente realizzato da programmatori aderendo alle definizioni fornite dal dirigente responsabile del servizio. Ma il disegno di quella procedura non può che essere profondamente diverso dal disegno di una procedura che deve essere eseguita da operatori umani, capaci di riempire gli spazi vuoti non definiti e di trovare

⁴ PAPERT S., *Mindstorms: children, computers, and powerful ideas*, Basic Books, Inc., USA, 1980.

⁵ WING J., *Computational thinking*, Communications of the ACM, 49 (3), pp. 33-35.

⁶ Si veda la definizione riassunta in ISTE (International Society for Technology in Education) e CSTA (Computer Science Teachers Association), *Computational thinking for K-12 education*, 2010, <<https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>> (ultimo accesso, aprile 2020).

soluzioni creative per risolvere i casi ambigui o speciali. Una maggiore capacità di *pensiero algoritmico* da parte del dirigente del servizio lo avvantaggerebbe in questo lavoro di progettazione, e questo andrebbe naturalmente a beneficio degli utenti.

Non c'è in realtà bisogno di pensare alla progettazione di sistemi estremamente complessi per capire l'importanza di sviluppare competenze informatiche; anche nel nostro piccolo è possibile incontrare esempi significativi. Immaginate di voler rinnovare il sito *web* della vostra scuola, o associazione, o attività commerciale. Probabilmente vi affidereste a un esperto, ma certamente il risultato ne gioverebbe se foste in grado di analizzare e spiegare all'esperto quali informazioni volete rendere disponibili, quali sono le relazioni che le collegano, e come è opportuno organizzarle per renderle più fruibili. Tutti aspetti tipici dell'informatica, in questo caso legati in particolare all'*analisi e all'organizzazione dei dati*.

Obiettivi di apprendimento in informatica

Verso la fine del 2017, nell'ambito del gruppo di lavoro “Informatica e Scuola” del consorzio CINI è stato elaborato e diffuso un documento intitolato “Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola”⁷, che indicheremo qui di seguito per semplicità “Indicazioni per l'Informatica”. Con questo documento ci si è posti l'ambizioso obiettivo di richiamare l'attenzione del mondo della scuola e dei decisori politici sul valore e l'opportunità dell'insegnamento dell'informatica come disciplina scientifica, indicando inoltre delle direzioni concrete per realizzare questa visione.

Le Indicazioni per l'Informatica, scritte seguendo l'impostazione delle ben più note “Indicazioni Nazionali per il Curricolo della Scuola dell'Infanzia e del Primo Ciclo d'Istruzione”⁸, propongono un percorso formativo suddiviso in tre fasi:

1. Scuola primaria: gli allievi sono sensibilizzati alle domande e sono accompagnati alla scoperta e all'esplorazione delle idee che stanno alla base della disciplina, mediante l'uso di dispositivi elettronici o tramite attività *unplugged*⁹.
2. Scuola secondaria di primo grado: l'obiettivo è consentire agli allievi di acquisire una maggiore autonomia, approfondendo le tematiche collega-

⁷ NARDELLI E., FORLIZZI L., LODI M., LONATI V., MIROLO C., MONGA M., MONTRESOR A. e MORPURGO A. (a cura di), *Proposta di Indicazioni Nazionali per l'insegnamento dell'informatica nella Scuola*, <<https://www.consortio-cini.it/index.php/it/component/attachments/download/745>> (ultimo accesso aprile 2020).

⁸ AA.VV., *Indicazioni nazionali per il curricolo della scuola dell'infanzia e del primo ciclo d'istruzione*, Annali della Pubblica Istruzione, Le Monnier, Firenze, 2012.

⁹ BELL T., WITTE I.H. e FELLOWS M., *CS unplugged. Computer science without a computer*, 1998. L'uso del termine *unplugged* nella didattica dell'informatica deriva da questa popolarissima collezione di attività didattiche, che è stata tradotta in oltre 20 lingue; la versione più recente è disponibile in italiano all'indirizzo <<https://classic.csunplugged.org/wp-content/uploads/2016/02/csunplugged-it.2015.1.0.pdf>> (ultimo accesso maggio 2020).

te alla rappresentazione e all'organizzazione dei dati e al concetto di algoritmo. In questa fase viene data particolare enfasi alle attività di programmazione, tipicamente in ambienti visuali.

3. Primo biennio della scuola secondaria di secondo grado: il perfezionamento delle competenze acquisite nelle prime due fasi è volto allo sviluppo della capacità di modellare problemi e progettare algoritmi.

Nel suo complesso, il percorso formativo punta a

sviluppare, oltre alla capacità di formulare algoritmi, la capacità di pensare a più livelli di astrazione, di modellare problemi, di raccogliere, rappresentarne e organizzarne i dati, di individuare schemi comuni, di ridurre la complessità di un problema scomponendolo in sotto-parti più semplici e affrontabili, di riconoscere come alcune soluzioni possano essere riusate e applicate a problemi simili, di usare linguaggi astratti (artificiali) per la descrizione di problemi, soluzioni, dati¹⁰.

In particolare, il documento propone traguardi e obiettivi di apprendimento, in funzione delle diverse età degli allievi, organizzandoli in 5 ambiti: algoritmi; programmazione; dati e informazione; consapevolezza digitale; creatività digitale. Come si vede, i primi tre ambiti fanno in particolare riferimento alle parole chiave discusse precedentemente, andando così a declinare il nucleo scientifico fondamentale della disciplina.

Va da sé che, per essere realizzata, la proposta ha bisogno di tempi curricolari adeguati, nonché (o soprattutto) di docenti opportunamente formati, sia sui fondamenti della disciplina, sia sulle metodologie più efficaci per insegnarla; nel contesto attuale, il raggiungimento di tutti i traguardi indicati è sostanzialmente impensabile. Tuttavia, il documento suggerisce una prospettiva complessiva, entro la quale collocare alcune prime attività e percorsi di apprendimento.

Algomotricità

L'informatica, per sua natura, si sposa molto bene con l'apprendimento esperienziale, l'approccio laboratoriale e collaborativo, l'apprendimento tra pari. Questo è stato lo spunto che ci ha motivato, nell'ambito del laboratorio ALaDDIn dell'Università degli Studi di Milano¹¹, a progettare e realizzare vari percorsi di apprendimento attivo mirati a diffondere una più corretta percezione dell'informatica e a presentarne gli aspetti fondamentali.

Tutti i percorsi di ALaDDIn si fondano su un approccio socio-costruttivi-

¹⁰ AA.VV., *Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola*, op. cit., p. 2.

¹¹ Il laboratorio ALaDDIn, di cui sono co-fondatrice, è attivo dal 2008; le pubblicazioni e i materiali prodotti dal laboratorio sono disponibili all'indirizzo <<https://aladdin.unimi.it>> (ultimo accesso aprile 2020).

sta¹². Il presupposto costruttivista fondamentale è che il sapere non si trasmette, ma si costruisce, facendo e riflettendo su ciò che si fa; tale costruzione avviene per mezzo delle interazioni sociali, cioè dal confronto con altri individui con i quali si devono negoziare significati e convincimenti.

Con un gioco di parole che combina il termine *algoritmo* con il termine *motorio*, abbiamo coniato il termine *algomotricità* per definire la nostra metodologia didattica. I percorsi algomotori prendono solitamente avvio dalla presentazione di un problema concreto, spesso ambientato in una situazione non convenzionale, in cui è richiesta la manipolazione di oggetti tangibili oppure l'utilizzo del proprio corpo per muoversi o interagire con i compagni e l'ambiente. Si procede poi in un processo di concettualizzazione, in cui gli oggetti tangibili vengono via via sostituiti da oggetti astratti, così arrivando alla formulazione di soluzioni "informatiche" (concettuali, non tecniche) al problema di partenza. Infine, si propone un'attività che prevede l'uso del computer: gli strumenti software con cui gli studenti devono confrontarsi sono specificamente progettati affinché questi ultimi possano sfruttare le conoscenze acquisite durante le fasi precedenti. Quest'ultima fase serve dunque a rendere esplicito il collegamento tra il concetto elaborato e la sua applicazione, ed è necessaria anche a soddisfare, almeno in parte, le aspettative degli alunni, in quanto anche loro identificano spesso l'informatica con l'utilizzo di un computer.

Nel laboratorio "Wikipasta", ad esempio, giocando con pasta e altri piccoli oggetti, si parte dal problema di marcare tipograficamente un testo per arrivare a scoprire e comprendere il linguaggio di marcatura con cui è scritta Wikipedia¹³ (Fig. 1).

Nel laboratorio "Sacchi nell'ascensore", l'obiettivo è automatizzare tramite un algoritmo, il trasporto di sacchi in un magazzino; gli alunni imparano a simulare l'esecuzione dell'algoritmo e a tenerne traccia, per capirne comportamento e limiti.

In altri percorsi invece, gli oggetti da manipolare nella prima fase sono oggetti informatici (di solito programmi sviluppati ad hoc). Gli alunni, lavorando

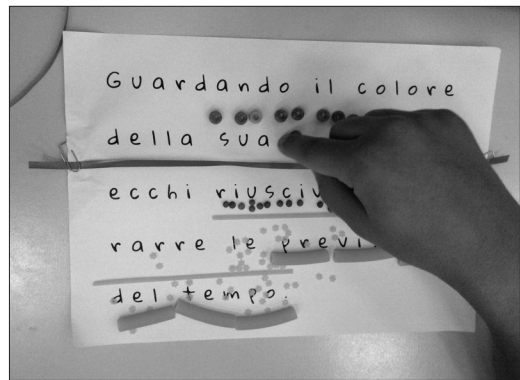


Fig. 1 – Laboratorio "Wikipasta"

¹² CARLETTI A. e VARANI, A., *Didattica costruttivista. Dalle teorie alla pratica in classe*, Erikson, Trento, 2005.

¹³ BELLETTINI C., LONATI V., MALCHIODI D., MONGA M., MORPURGO A. e TORELLI M., *Exploring the processing of formatted texts by a kynesesthetic approach*, in KNOBELSDORF M. e ROMEIKE R., *Proceedings of WIPSC 2012*, ACM, 2013, pp. 143-144.

do in piccoli gruppi, sono accompagnati a ragionare sulle proprietà, le regole e i meccanismi di funzionamento di questi oggetti; partendo da casi specifici e via via astraendo verso il “caso generale”, gli alunni devono formulare ipotesi, progettare piccoli esperimenti, condurre osservazioni e tornare a riflettere sulle ipotesi formulate precedentemente, in un processo iterativo caratteristico del metodo scientifico. Nel laboratorio “Scacchiera”, ad esempio, gli alunni devono scoprire come funziona una codifica che permette di descrivere in maniera inequivocabile ma sintetica la posizione dei pezzi degli scacchi; per farlo possono sfruttare un piccolo programma che consente loro di fare osservazioni ed esperimenti.

Questo approccio metodologico si può naturalmente applicare a molti altri temi informatici, al di là di quelli esemplificati qui sopra, adattandolo ai vari contesti e all'età degli alunni coinvolti.

Conclusioni

Dovrebbe essere chiaro, a questo punto, che l'informatica va riconosciuta come la *scienza che studia i principi e i metodi per l'elaborazione automatica dell'informazione* e non va confusa con la capacità di usare il computer o con le competenze digitali.

Per sua stessa natura, l'informatica si presta ad essere presentata in classe attraverso attività laboratoriali e collaborative, secondo un approccio socio-costruttivista che promuove processi di apprendimento centrati sul discente e che prendono avvio dalla riflessione che queste stesse attività sono in grado di generare.

Questo modo di fare informatica insegna agli alunni a prestare attenzione ai dettagli, a fare piani, a disciplinare l'organizzazione del proprio pensiero; richiede di lavorare con astrazioni per creare e manipolare oggetti concreti e loro rappresentazioni; dimostra l'importanza della precisione descrittiva e della coerenza logica; invita a osservare, formulare ipotesi e argomentazioni, sviluppare esperimenti; favorisce infine lo sviluppo delle capacità di analisi e di pensiero critico.

Si tratta di una disciplina scientifica affascinante, di grande valore formativo: l'informatica merita di essere svelata agli insegnanti e proposta agli alunni, fin dalla scuola del primo ciclo.