

Poche frasi, una disciplina.

L'informatica nelle nuove Indicazioni

Nazionali per il primo ciclo

Michael Lodi, Università di Bologna

Violetta Lonati, Università di Milano

Sommario

Le nuove Indicazioni Nazionali per la scuola dell'infanzia e del primo ciclo, pubblicate in Gazzetta Ufficiale nel gennaio 2026, introducono per la prima volta competenze, obiettivi di apprendimento e conoscenze di informatica, collocati nelle discipline matematica e tecnologia. Proponiamo un'analisi delle frasi dedicate all'informatica, organizzata attorno a tre nuclei scientifici fondamentali – dati e informazione, algoritmi, programmazione – e ai due ambiti della consapevolezza e della creatività digitale. Discutiamo infine le condizioni perché il curriculum diventi pratica di classe: formazione degli insegnanti in servizio, riforma dei percorsi universitari per la formazione iniziale dei docenti, materiali di qualità e, in prospettiva, il riconoscimento dell'informatica come disciplina autonoma.

Parole chiave: informatica; Indicazioni Nazionali; primo ciclo; didattica dell'informatica; formazione degli insegnanti.

1 L'informatica nel curriculum

Con il decreto ministeriale del 9 dicembre 2025, pubblicato in Gazzetta Ufficiale il 27 gennaio 2026, le nuove Indicazioni Nazionali per la scuola dell'infanzia e del primo ciclo di istruzione¹ introducono per la prima volta competenze, obiettivi di apprendimento e conoscenze² esplicitamente riferiti all'informatica (MIM – Ministero dell'Istruzione e del Merito 2025). Gli elementi di informatica sono collocati all'interno di

¹ Che include la scuola primaria e la scuola secondaria di primo grado.

² Conoscenze che, a differenza di competenze e obiettivi, non sono prescrittive, ma aiutano a comprendere meglio il pensiero degli autori delle Indicazioni.

matematica e di tecnologia, e arriveranno in classe dall'anno scolastico 2026/2027 a partire dalle classi prime della primaria e della secondaria di primo grado. È una novità che rischia di passare inosservata, perché si concretizza in poche frasi molto compresse: non nasce una nuova disciplina scolastica, eppure quelle frasi contengono un disegno preciso, e vale la pena leggerle con attenzione.

Perché l'informatica a scuola, e fin dalla primaria? L'elaborazione automatica dei dati è il motore invisibile di app, piattaforme, mercati, servizi pubblici: chi non possiede i concetti per capire come i dati rappresentano la realtà, che cosa significa che una decisione è delegata a un algoritmo, come un programma fa ciò che fa, guarda questo mondo da spettatore. Se gli strumenti cambiano, quei concetti sono stabili, e si possono insegnare fin da piccoli come i fondamenti della matematica o delle scienze: il tanto evocato "pensiero computazionale" (Wing 2006) non può svilupparsi senza lo studio e la pratica dell'informatica (Lodi et al. 2017; Lodi and Martini 2021; Lodi 2020).

In questo articolo proponiamo una lettura analitica delle frasi dedicate all'informatica nelle nuove Indicazioni: da dove vengono, che cosa dicono e, soprattutto, quale informatica c'è dietro.

2 Una novità che viene da lontano

L'introduzione dell'informatica nelle Indicazioni Nazionali arriva dopo decenni di rapporti tormentati tra informatica e scuola italiana, che si possono riassumere individuando alcune fasi "storiche": gli anni Ottanta con le sperimentazioni con LOGO e le interessanti proposte avanzate da Giovanni Prodi (Prodi 1990; Barozzi 2010), poi confluite nel Piano Nazionale Informatica (Barozzi and Ciarrapico 2003), centrate sugli aspetti costruttivi della matematica e l'educazione al pensiero algoritmico; la lunga stagione della "patente del computer" in cui fare informatica significava usare applicativi; gli anni di "coding" e "pensiero computazionale" entrati nella normativa senza un chiaro rapporto con la disciplina (Lodi and Martini 2024). Le nuove Indicazioni scelgono finalmente la parola *informatica* e richiamano la Raccomandazione del Consiglio dell'Unione europea del 23 novembre 2023 (Consiglio dell'Unione europea 2023), che invita a introdurre l'insegnamento fin dall'inizio dell'istruzione obbligatoria, come confermato dalle successive linee guida (Commissione europea and Direzione generale dell'Istruzione, della gioventù, dello sport e della cultura 2026).

Nonostante l'orientamento istituzionale prevalente in passato, sono tuttavia sempre esistite iniziative volte a promuovere una diversa visione dell'informatica a scuola, intesa come disciplina scientifica dotata di propri concetti, metodi e fondamenti culturali, non riducibile al semplice uso delle tecnologie o all'acquisizione di competenze di programmazione (Nardelli 2020). Nel dicembre 2017 il CINI, Consorzio Interuniversitario Nazionale per l'Informatica, con i professori universitari di informatica (GRIN) e di ingegneria informatica (GII), pubblicò una "Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola" (CINI, Laboratorio Nazionale Informatica e Scuola 2017), alla cui stesura abbiamo preso parte. La proposta

guardava alle esperienze internazionali, in particolare al curriculum inglese di *Computing*, obbligatorio dalle elementari fin dal 2014 (Department for Education 2013), e ne adattava lo spirito al formato dei documenti curriculari italiani, con traguardi e obiettivi organizzati in cinque ambiti: **dati e informazione, algoritmi, programmazione, consapevolezza digitale, creatività digitale**.

La bozza delle nuove Indicazioni, diffusa nel marzo 2025 per il dibattito pubblico³, riprendeva quasi alla lettera gli obiettivi e i traguardi della proposta CINI; la versione pubblicata nel testo definitivo rielabora quei contenuti, rendendoli più accessibili ai non informatici, ma anche accorpandoli e, in parte, riducendoli. Ne risultano frasi brevi che condensano, a volte in una riga, percorsi concettuali articolati: proviamo a leggerle da vicino, seguendo i cinque ambiti della proposta CINI, i primi tre affidati soprattutto alla matematica, gli ultimi due alla tecnologia.

3 Competenze digitali o competenze informatiche?

Prima degli obiettivi, conviene partire dalle premesse culturali, dove si legge che «esiste differenza tra “competenze digitali” (intese come padronanza di uso efficace, sicuro e consapevole di dispositivi, strumenti e tecnologie digitali) e “competenze informatiche”, riferite, queste ultime, alla comprensione di come i dati (rappresentati digitalmente, cioè in forma numerico/simbolica) possano essere elaborati in maniera automatica, ad opera di “agenti” che eseguono meccanicamente procedimenti ideati da umani» (MIM – Ministero dell’Istruzione e del Merito 2025, 25).

È una distinzione cruciale. Sancisce, in un documento curricolare nazionale, che saper usare le tecnologie e capire l’informatica sono cose diverse: confonderle è stato l’equivoco dominante degli ultimi trent’anni, un po’ come se la biologia fosse la scienza dei microscopi. La novità sta tutta nell’elaborazione automatica, e la premessa di matematica lo ribadisce: «La novità dell’Informatica rispetto alla rappresentazione digitale dei dati è che essa consente di elaborarli in modo completamente automatico mediante l’utilizzo di un dispositivo (informatico), che funge da mero esecutore meccanico di un procedimento di calcolo ideato e progettato dall’uomo» (MIM – Ministero dell’Istruzione e del Merito 2025, 120). L’umanità progetta algoritmi da secoli, e la parola stessa deriva dal nome di al-Khwārizmī, matematico del IX secolo; ciò che l’informatica aggiunge è che quei procedimenti vengano eseguiti da agenti privi di intelligenza, di buon senso e di libero arbitrio (Nardelli 2022, cap. 3). Questo cambia il modo in cui i procedimenti vanno pensati e descritti, e gli effetti che la loro esecuzione, replicabile su scala enorme, produce sulla società.

L’obiettivo dell’informatica nel primo ciclo è che gli allievi acquisiscano la prospettiva culturale della disciplina, concretizzata «modellando tramite dati – opportunamente organizzati e rappresentati – le informazioni rilevanti per gli obiettivi che devono essere raggiunti; progettando algoritmi, ovvero descrizioni precise e non

³ <https://www.mim.gov.it/documents/20182/0/Nuove+indicazioni+2025.pdf/cebce5de-1e1d-12de-8252-79758c00a50b?version=1.0&t=1741684578272>

ambigue (in riferimento a uno specifico esecutore) di procedure (per raggiungere tali obiettivi) che si prestano ad essere automatizzate; utilizzando linguaggi artificiali, direttamente eseguibili, per esprimere gli algoritmi come programmi informatici, in modo che questi possano essere eseguiti automaticamente da “agenti meccanici”» (MIM – Ministero dell’Istruzione e del Merito 2025, 120–21). In una sola frase, tre grandi idee che strutturano gli ambiti scientifici: le informazioni si esprimono mediante dati, codificati digitalmente; gli algoritmi operano sui dati per svolgere compiti e risolvere problemi; i programmi esprimono gli algoritmi in una forma che una macchina può eseguire (Bell et al. 2018).

4 Dati e informazione

Alla fine della primaria gli alunni dovranno saper «esprimere informazioni mediante dati di varia natura e codificare tali dati anche digitalmente»; già in terza, «scegliere ed utilizzare oggetti o simboli per rappresentare informazioni»; in quinta, «scegliere combinazioni di simboli per rappresentare dati strutturati». Tra le conoscenze compare la «rappresentazione di dati e informazioni e loro differenza».

Quella differenza, che a molti suonerà nuova, è un’idea profonda (Lonati and Monga 2025) e insegnabile prestissimo. Un dato, da solo, non dice nulla: “10” può essere un’età, un prezzo, un giorno, una velocità (cfr. Figura 1). Diventa informativo quando chi lo riceve sa interpretarlo in un contesto.

Fig. 1. Un dato, da solo, non dice nulla: “10” può essere un’età, un prezzo, una velocità, ecc.

Viceversa, il passaggio dall’informazione ai dati comporta sempre una scelta. Quale dato scegliamo per esprimere un’informazione dipende dalle finalità e dal contesto: per individuare un poligono tra i due nella parte sinistra della Figura 2 posso usare il colore (o il numero di lati, la posizione sopra/sotto, ...); ma quando i poligoni diventano quattro, il colore da solo non basta (né il numero di lati, né la posizione sopra/sotto).

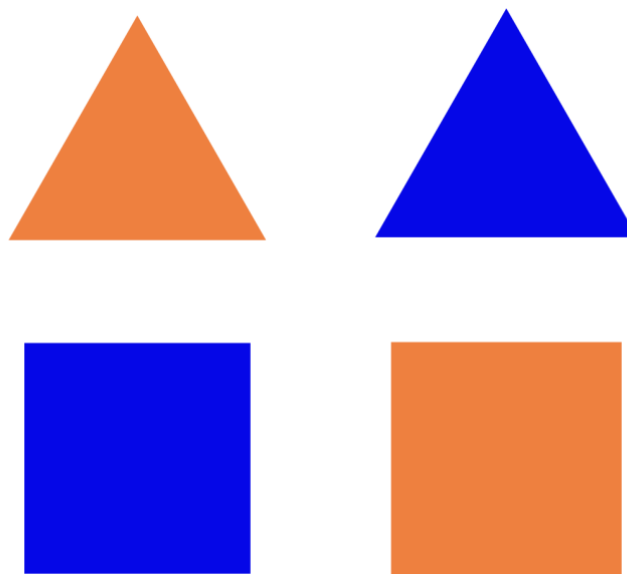


Fig. 2. Quattro poligoni in cui un singolo dato, come colore, forma o posizione, non basta a identificare univocamente un oggetto.

Scegliere i dati significa modellare: ogni modello trattiene alcuni aspetti della realtà e ne esclude altri, in funzione dello scopo. Non a caso la competenza per la secondaria chiede di «rappresentare informazioni in relazione al compito da svolgere»: è il compito a determinare quali dati servono, quali sono superflui, quando due rappresentazioni sono intercambiabili.

Un ulteriore passo è la codifica: qualunque dato «di varia natura» (numeri, ma anche testi, immagini, suoni) può essere rappresentato come una sequenza di simboli scelti da un alfabeto finito, al limite di due soli simboli, purché le regole del sistema permettano di ricostruire il dato senza perdite né ambiguità.

Per questo gli obiettivi insistono sull'esplicitazione dell'interpretazione dei simboli, ossia la legenda. Molto di ciò che serve, peraltro, può essere introdotto a partire da contesti già familiari nelle classi, rileggendoli in chiave informatica: legende delle mappe, sistemi di scrittura, carte da gioco fitte di numeri e simboli (Boldi et al. 2025).

È qui che il «sistema binario», elencato tra le conoscenze della secondaria, trova il suo senso informatico: come caso paradigmatico di sistema di codifica, esplorabile per scoperta (che cosa codificherà questa sequenza? con quali regole?), e non come ennesima procedura di conversione da eseguire meccanicamente. La possibilità di ricondurre dati di diversa natura a rappresentazioni digitali è una delle ragioni della straordinaria versatilità dei dispositivi informatici: numeri, testi, immagini, suoni, video e perfino programmi possono essere conservati, copiati, trasmessi e manipolati dallo stesso tipo di macchina perché ricondotti a una forma comune (Bell et al. 2021). Per questo un computer non è una macchina specializzata per un solo compito, a differenza di molti dispositivi analogici tradizionali, ma un esecutore universale in grado di manipolare simboli attraverso programmi che svolgono compiti specifici ogni volta diversi.

5 Algoritmi

Alla fine della scuola primaria si chiede di saper «descrivere procedure per lo svolgimento di compiti pratici mediante algoritmi». In particolare, come obiettivo alla fine della terza, di «descrivere a parole attività della vita quotidiana tramite sequenze di passi precisi e non ambigui e saperle eseguire»; tra le conoscenze, il «concetto di algoritmo e sua esecuzione rigorosa».

La scuola, e in particolare la matematica a scuola, è piena di algoritmi da sempre: le operazioni in colonna, il calcolo del massimo comune divisore, le costruzioni con riga e compasso. La prospettiva informatica introduce due spostamenti di senso. Il primo riguarda la non ambiguità, che non è un generico invito alla precisione: una descrizione, dicono le Indicazioni Nazionali, è non ambigua solo in riferimento a uno specifico *esecutore*, di cui vanno esplicitate le operazioni elementari di cui è capace: il suo «catalogo di azioni». È il punto debole della metafora della ricetta, onnipresente nei libri di testo (Monga et al. 2025): «attendi qualche secondo e riprova ad assaggiare» funziona per un cuoco dotato di giudizio, mentre l'esecutore automatico non può decidere quanto valga «qualche». Descrivere per iscritto la preparazione di un panino e vederla

eseguita alla lettera da qualcuno che “fa l’automa” mostra ai bambini esattamente questo: chi progetta e chi esegue sono separati, l’esecuzione avverrà dopo, senza possibilità di chiarimenti, e ogni ambiguità presenterà il conto (Bissoli et al. 2026; Lodi et al. 2026). Anche la chiusa dell’obiettivo di terza, quel «saperle eseguire», acquista così spessore: interpretare il ruolo dell’esecutore, compiendo alla lettera, fisicamente o mentalmente, i passi descritti da altri, è il modo più diretto per scoprire dove si annidano le ambiguità.

Il secondo spostamento riguarda il rapporto tra algoritmi e problemi. Nella pratica scolastica l’algoritmo è spesso uno strumento appreso da applicare, tanto che in didattica della matematica “apprendimento algoritmico” indica proprio l’abilità di eseguire metodi risolutivi (Crippa, Branchetti, et al. 2025). In informatica il rapporto si rovescia: l’algoritmo è il punto d’arrivo, la soluzione di un problema generale, dove per problema generale si intende un’intera classe di casi che condividono la struttura e differiscono per i dati (Crippa, Lonati, et al. 2025). Tre quesiti che chiedono se altrettante tavolette di cioccolato costino più o meno della media al chilo sono tre istanze dello stesso problema: si possono risolvere uno per uno, oppure progettare una volta per tutte un procedimento che li risolva tutti, e poi delegarne l’esecuzione. L’ideazione spesso passa dalla scomposizione: individuare sottocompiti semplici e combinarli – come quando, ad esempio, per ordinare una collezione di oggetti, si riparte ogni volta dal saperne trovare il più piccolo. Progettare, analizzare, confrontare algoritmi, capire perché un metodo funziona invece di limitarsi ad applicarlo, sono attività creative e riflessive, l’opposto dell’esecuzione meccanica. In questa direzione va l’obiettivo della secondaria: «esaminare un algoritmo o un programma per capirne il comportamento, identificarne eventuali difetti e correggerli (debug)». L’algoritmo diventa un oggetto da studiare, mettere alla prova, migliorare.

6 Programmazione

Alla fine della primaria gli alunni dovranno saper «scrivere e comprendere semplici programmi, espressi in elementari linguaggi di programmazione a scopo didattico, e valutarne l’adeguatezza rispetto al compito che si vuole automatizzare»; già in terza, «scrivere semplici programmi e verificare, mediante la loro esecuzione, se svolgono il compito previsto ed eventualmente correggerli»; in quinta compaiono le istruzioni «di reazione ad eventi, e selezione (con condizione elementare) e ripetizione», e l’obiettivo di «tradurre un algoritmo in un programma ed esaminarne il comportamento anche al fine di correggerlo». Alla secondaria si aggiungono il «concetto di variabile», i programmi «strutturati in componenti modulari», le «funzioni e procedure», e la capacità di valutare l’esito di un programma «seguendone i passi e tenendo traccia del valore delle variabili». Le Indicazioni, si noti, parlano sempre di programmazione: mai di “coding”, parola dilatata negli anni fino a coprire attività prive di contenuto informatico (Francalanci 2020; Lodi and Martini 2024).

Per essere effettivamente eseguibile, un algoritmo va scritto sotto forma di programma (Lodi and Martini 2025a): la sua espressione in un linguaggio artificiale dotato di regole precise, comprensibile agli umani e, al

contempo, eseguibile⁴ da una macchina (Lonati et al. 2022). I linguaggi progettati «a scopo didattico» citati dal testo esistono e sono maturi, dalla programmazione visuale a blocchi (Resnick et al. 2009) fino a linguaggi testuali semplici come LOGO (Papert 1980); ambienti in cui si può vedere l'esecutore automatico compiere azioni del suo "catalogo" sulla base di istruzioni impartite dal programmatore. I costrutti elencati sono i mattoni di ogni programma: azioni in sequenza, ripetizione ("ripeti mentre/ripeti finché", "ripeti N volte"), selezione al verificarsi di una condizione ("se... allora..."), reazione a un evento ("quando si clicca su..."). Ripetizione e selezione si applicano anche a interi blocchi di istruzioni, come quando si disegna un quadrato ripetendo quattro volte la coppia "traccia un lato, gira di 90 gradi".

Due dettagli meritano attenzione. Il primo: si chiede di scrivere e comprendere, perché leggere un programma altrui, prevederne il comportamento, modificarlo per vedere che cosa succede è palestra potente anche più di scrivere da zero, specialmente oggi che i programmi possono essere scritti con l'aiuto dell'intelligenza artificiale (Lodi and Martini 2025b; Izu et al. 2019). Il secondo: la correttezza è definita non in assoluto ma in relazione «al compito che si vuole automatizzare» e la verifica mediante l'esecuzione introduce una pratica quasi sperimentale fatta di previsioni, prove e revisioni, in cui l'errore – il bug – è materiale di lavoro ordinario e non fallimento.

7 Consapevolezza e creatività: la parte di tecnologia

A tecnologia le Indicazioni affidano i due ambiti restanti, con una premessa netta: le tecnologie informatiche «vanno presentate agli studenti nella loro natura di strumenti che funzionano sulla base dei risultati teorici e metodologici dell'informatica, ossia come mezzo e non come scopo», evitando «errori metodologici basati su competenze digitali declinate come mero utilizzo di strumenti» (MIM – Ministero dell'Istruzione e del Merito 2025, 140–41).

La consapevolezza digitale, negli obiettivi, è fondata sulla comprensione: «riconoscere la presenza dei computer nella vita quotidiana e distinguere Internet come rete di comunicazione dai servizi e contenuti forniti» alla primaria, fino a comprendere alla secondaria «l'architettura di principio (fisica e funzionale) di un sistema informatico» e i «meccanismi fondamentali di Internet». Distinguere la rete dai servizi che vi transitano (ad esempio, contenuti web, messaggistica, streaming), sapere che cosa sono i «"codici segreti" per tutelare la riservatezza», comprendere «il valore dei dati personali e sensibili» e i «rischi sociali connessi alla raccolta sistematica dei dati»: sono le fondamenta concettuali senza le quali l'educazione alla cittadinanza digitale si riduce a un elenco di divieti. Nelle conoscenze della secondaria entra anche la «valutazione critica dei dati reperiti su rete e delle informazioni fornite da sistemi di intelligenza artificiale generativa», con un principio di fondo che il testo enuncia con chiarezza: «che siano gli esseri umani a

⁴ Dopo un processo stratificato di traduzioni.

mantenere il controllo sulle decisioni basate su sistemi informatici che possono avere un impatto significativo sulle persone» (MIM – Ministero dell'Istruzione e del Merito 2025, 141).

La creatività digitale, infine, sposta gli alunni dal consumo alla produzione: «sviluppare un atteggiamento positivo nei confronti delle applicazioni informatiche riconoscendone le potenzialità come strumenti di espressione personale», «creare semplici contenuti e applicazioni informatiche a fini espressivi usando ambienti adatti e modificando o combinando quanto già disponibile». Quel «modificando o combinando» legittima una pratica preziosa, e profondamente informatica: imparare a partire da artefatti esistenti, esplorandoli, riusandoli e ricombinandoli. È la logica del *creative computing* e del *creative learning*, nel solco del costruzionismo di Papert (ideatore di LOGO) e alla base di Scratch, dove programmare significa progettare media interattivi per l'espressione creativa e personale (Papert 1980; Brennan and Resnick 2012; Brennan et al. 2014; Resnick 2017).

8 Con la matematica, ma con la propria identità

La collocazione dell'informatica all'interno di matematica e tecnologia è una scelta pragmatica: con orari e organici invariati, era difficile immaginare altro. La vicinanza con la matematica è comunque reale e non va sprecata: rigore, astrazione, *problem solving*, argomentazione sono terreno comune, e l'algoritmo è un autentico oggetto di confine tra le due discipline (Crippa, Branchetti, et al. 2025). Il punto di vista informatico può arricchire quello matematico, spingendo dalla soluzione del singolo esercizio verso il problema generale e l'automazione della sua soluzione; viceversa, la sensibilità matematica per la giustificazione e la dimostrazione risuona con l'attenzione che l'informatica pone sulla correttezza di algoritmi e programmi.

Ma perché l'incontro sia fecondo, le identità devono restare riconoscibili: la ricerca sull'interdisciplinarietà insegna che il coordinamento e la trasformazione reciproca richiedono una fase preliminare in cui ciascuna disciplina chiarisce la propria specificità (Sbaraglia 2023; Lodi et al. 2025). La convivenza curricolare è un'opportunità che va costruita. L'informatica può certamente fornire strumenti concettuali per potenziare l'insegnamento/apprendimento della matematica; gli obiettivi nelle Indicazioni, tuttavia, delineano una prospettiva più ampia: l'informatica si configura come la disciplina che offre una specifica lente interpretativa sui fenomeni, ormai ubiqui, in cui l'elaborazione automatica dell'informazione ha un ruolo. È questa la dimensione culturale che le stesse Indicazioni richiamano nel paragrafo «Perché si studia la matematica» (MIM – Ministero dell'Istruzione e del Merito 2025, 120).

9 Perché le frasi diventino scuola

Resta la domanda decisiva: chi insegnerà tutto questo? Le Indicazioni introducono l'informatica senza ore dedicate né docenti specialisti, affidandola a insegnanti che, per la maggior parte, non l'hanno mai studiata. Non è una colpa: finora l'informatica non è mai stata una materia scolastica, e il corso di laurea in Scienze

della Formazione Primaria (LM-85 bis) prevede le tecnologie didattiche, ma nessun insegnamento di informatica né della sua didattica (MIUR 2010). L'aggiornamento frettoloso dei libri di testo, inoltre, corre il rischio di far ripetere gli errori fatti nei testi dello scorso decennio: metafore fuorvianti, "coding" come etichetta buona per tutto, algoritmi senza esecutore (Monga et al. 2025; Capecchi et al. 2025).

L'esperienza inglese è istruttiva. Quando nel 2014 il Computing divenne obbligatorio, la difficoltà maggiore non fu scrivere il curriculum, ma sostenerlo: pochi anni dopo, un rapporto della Royal Society fotografava una scuola in affanno proprio nella preparazione dei docenti (The Royal Society 2017), e il governo rispose nel 2018 con un investimento pubblico di 84 milioni di sterline nel National Centre for Computing Education (NCCE), una rete nazionale per fornire formazione e risorse agli insegnanti di informatica in Inghilterra (The Royal Society 2019). La lezione è che un curriculum vive solo se accompagnato da un piano di formazione strutturale, in servizio e soprattutto a livello di formazione iniziale: servono la riforma della LM-85 bis e percorsi abilitanti che includano l'informatica e la sua didattica, servono materiali di qualità, guide, esempi commentati, comunità di pratica in cui insegnanti e ricercatori lavorino insieme. Esperienze pilota, progettate anche da gruppi di ricerca interdisciplinari, stanno già partendo, ma serve un piano strutturale: la riuscita non può essere lasciata alla buona volontà dei singoli.

E serve, in prospettiva, il passo che queste Indicazioni preparano senza compierlo: il riconoscimento dell'informatica come disciplina scolastica autonoma, con ore e docenti propri, per tutto il percorso dell'obbligo, nella direzione indicata dalla Raccomandazione europea e già imboccata da molti Paesi. Le poche frasi entrate nelle Indicazioni sono, per chi lavora da anni su questi temi, un punto di arrivo quasi insperato; per la scuola italiana, un punto di partenza. Dietro ciascuna di esse, come abbiamo provato a mostrare, c'è un'intera disciplina: adesso occorre creare le condizioni perché entri davvero in classe.

Riferimenti bibliografici

Barozzi, Giulio Cesare. 2010. *Giovanni Prodi (1925–2010): Un Ricordo Personale*. <https://matematica.unibocconi.eu/articoli/giovanni-prodi-1925-2010-un-ricordo-personale>.

Barozzi, Giulio Cesare, and Lucia Ciarrapico. 2003. "Il Piano Nazionale Per l'informatica." *Bollettino Dell'Unione Matematica Italiana. La Matematica Nella Società e Nella Cultura*, Serie VIII, 6-A (3): 441–61.

Bell, Tim, Paul Tymann, and Amiram Yehudai. 2018. "The Big Ideas in Computer Science for K-12 Curricula." *Bulletin of EATCS* 124: 36–46. <http://bulletin.eatcs.org/index.php/beatcs/article/view/521>.

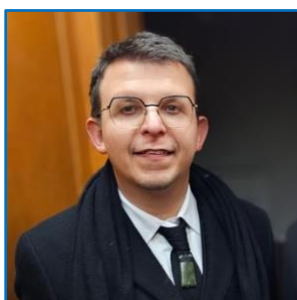
Bell, Tim, Paul Tymann, and Amiram Yehudai. 2021. *The Big Ideas of K-12 Computer Science Education*. <https://www.csse.canterbury.ac.nz/big-ideas/BigIdeas-webdocument-15-Feb-2021.pdf>.

Bissoli, Giorgia, Agnese Del Zozzo, Michael Lodi, and Alberto Montresor. 2026. *Sulle orme di Milù – Guida di Informatica per l'insegnante*. Guida docente, Mondadori Education, Milano, ISBN 9791220419833.

- Boldi, Arianna, Sara Capecchi, Violetta Lonati, and Mattia Monga. 2025. "Playcards, Pasta and Rectangles: Teaching Information Representation and Data Encoding in 4th Grade." *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (New York, NY, USA), ITiCSE 2025, 354–60. <https://doi.org/10.1145/3724363.3729091>.
- Brennan, Karen, Christan Balch, and Michelle Chung. 2014. *Creative Computing Curriculum Guide*. <https://scratched.gse.harvard.edu/guide/curriculum.html>.
- Brennan, Karen, and Mitchel Resnick. 2012. "New Frameworks for Studying and Assessing the Development of Computational Thinking." *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (Vancouver, BC, Canada). <https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>.
- Capecchi, Sara, Michael Lodi, Violetta Lonati, and Anna Morpurgo. 2025. "L'informatica negli attuali libri di testo della scuola secondaria di primo grado, alla luce delle Indicazioni Nazionali 2025." In *La Didattica della Matematica con e per la scuola*, edited by Giorgio Bolondi, Bruno D'Amore, and Federica Ferretti. Bonomo Editore.
- CINI, Laboratorio Nazionale Informatica e Scuola. 2017. *Proposta Di Indicazioni Nazionali Per l'insegnamento Dell'informatica Nella Scuola*. Consorzio Interuniversitario Nazionale per l'Informatica. <https://www.consortio-cini.it/images/Proposta-Indicazioni-Nazionali-Informatica-Scuola-numerata.pdf>.
- Commissione europea, and Direzione generale dell'Istruzione, della gioventù, dello sport e della cultura. 2026. *Linee Guida Per l'insegnamento Dell'informatica: Strategie Pratiche Per Le Classi Europee*. Ufficio delle pubblicazioni dell'Unione europea. <https://doi.org/10.2766/0709808>.
- Consiglio dell'Unione europea. 2023. *Raccomandazione del Consiglio del 23 novembre 2023 sul miglioramento dell'offerta di abilità e competenze digitali nell'istruzione e nella formazione*. https://eur-lex.europa.eu/legal-content/IT/TXT/?uri=OJ:C_202401030.
- Crippa, Marta, Laura Branchetti, Luca Lamanna, and Violetta Lonati. 2025. "Conoscenza procedurale, algoritmi e problemi: spunti di riflessione da una prospettiva interdisciplinare tra matematica e informatica." In *La Didattica della Matematica con e per la scuola*, edited by Giorgio Bolondi, Bruno D'Amore, and Federica Ferretti. Bonomo Editore.
- Crippa, Marta, Violetta Lonati, Luca Lamanna, and Laura Branchetti. 2025. "Educare al pensiero algoritmico: riflessioni e proposte per il primo ciclo." In *ITADINFO 2025. Terzo Convegno Italiano sulla Didattica dell'Informatica*, edited by Maria Angela Pellegrino and Veronica Rossano. Ledizioni.
- Department for Education. 2013. *National Curriculum in England: Computing Programmes of Study*. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>.
- Francalanci, Lucia. 2020. "Coding." *Italiano Digitale* 12 (1): 95–101. <https://doi.org/10.35948/2532-9006/2020.3311>.

- Izu, Cruz, Carsten Schulte, Ashish Aggarwal, et al. 2019. "Fostering Program Comprehension in Novice Programmers: Learning Activities and Learning Trajectories." *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (New York, NY, USA), ITiCSE-WGR '19, 27–52. <https://doi.org/10.1145/3344429.3372501>.
- Lodi, Michael. 2020. "Informatical Thinking." *Olympiads in Informatics* 14: 113–32. <https://doi.org/10.15388/oi.2020.09>.
- Lodi, Michael, Agnese Del Zozzo, Alberto Montresor, and Giorgia Bissoli. 2026. *Insegnare Informatica - Una guida per docenti del primo ciclo*. Vol. 1.
- Lodi, Michael, Violetta Lonati, and Simone Martini. 2025. "Informatica nelle Indicazioni di Matematica: un'opportunità interdisciplinare." In *La Didattica della Matematica con e per la scuola*, edited by Giorgio Bolondi, Bruno D'Amore, and Federica Ferretti. Bonomo Editore.
- Lodi, Michael, and Simone Martini. 2021. "Computational Thinking, Between Papert and Wing." *Science & Education* 30 (4): 883–908. <https://doi.org/10.1007/s11191-021-00202-5>.
- Lodi, Michael, and Simone Martini. 2024. "Coding, pensiero computazionale e informatica: quale evoluzione possibile?" *ITADINFO 2024. Secondo Convegno Italiano sulla Didattica dell'Informatica* (Genova), 20–24.
- Lodi, Michael, and Simone Martini. 2025a. "Algoritmi, programmi e linguaggi." *Nuova Secondaria* 43 (3): 80–83.
- Lodi, Michael, and Simone Martini. 2025b. "Le Notional machine per l'apprendimento della programmazione, anche nell'era dell'AI generativa." In *ITADINFO 2025. Terzo Convegno Italiano sulla Didattica dell'Informatica*, edited by Maria Angela Pellegrino and Veronica Rossano. Ledizioni.
- Lodi, Michael, Simone Martini, and Enrico Nardelli. 2017. "Abbiamo davvero bisogno del pensiero computazionale?" *Mondo Digitale*, no. 72: 1–15. https://mondodigitale.aicanet.net/2017-5/articoli/MD72_02_abbiamodavverobisognodelpensierocomputazionale.pdf.
- Lonati, Violetta, Claudio Mirolo, and Mattia Monga. 2022. "Di cosa parliamo quando parliamo di programmi?" *Mondo Digitale* 20 (97): 1–12.
- Lonati, Violetta, and Mattia Monga. 2025. "Rendere concreta l'informazione per elaborarla." *Nuova Secondaria* 43 (3): 74–79.
- MIM – Ministero dell'Istruzione e del Merito. 2025. *Indicazioni nazionali per il curricolo della scuola dell'infanzia e del primo ciclo d'istruzione*. https://www.mim.gov.it/documents/20182/10554370/curricolo_web.pdf/.
- MIUR. 2010. *Decreto 10 settembre 2010, n. 249 – Regolamento concernente la definizione della disciplina dei requisiti e delle modalità della formazione iniziale degli insegnanti*.
- Monga, Mattia, Renzo Davoli, Luca Lamanna, and Luca Forlizzi. 2025. "Il concetto di algoritmo nei libri di testo: spunti per una riflessione." In *La Didattica della Matematica con e per la scuola*, edited by Giorgio Bolondi, Bruno D'Amore, and Federica Ferretti. Bonomo Editore.
- Nardelli, Enrico, ed. 2020. *Coding e Oltre: L'informatica Nella Scuola*. Lisciani Scuola.

- Nardelli, Enrico. 2022. *La Rivoluzione Informatica: Conoscenza, Consapevolezza e Potere Nella Società Digitale*. Themis.
- Papert, Seymour. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Prodi, Giovanni. 1990. "Quale informatica per la scuola media?" *L'Insegnamento Della Matematica e Delle Scienze Integrate* 13 (4): 355–58.
<https://www.centromorin.it/home/pubblicazioni/riviste/speciale.asp?cs=2>.
- Resnick, Mitchel. 2017. *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*. MIT Press. <https://doi.org/10.7551/mitpress/11017.001.0001>.
- Resnick, Mitchel, John Maloney, Andrés Monroy-Hernández, et al. 2009. "Scratch: Programming for All." *Communications of the ACM* 52 (11): 60–67. <https://doi.org/10.1145/1592761.1592779>.
- Sbaraglia, Marco. 2023. "Teaching Informatics to Novices: Big Ideas and the Necessity of Optimal Guidance." PhD thesis, Alma Mater Studiorum – Università di Bologna. <https://doi.org/10.48676/unibo/amsdottorato/10914>.
- The Royal Society. 2017. *After the Reboot: Computing Education in UK Schools*. Royal Society. <https://royalsociety.org/topics-policy/projects/computing-education/>.
- The Royal Society. 2019. *Policy briefing on teachers of computing*. <https://royalsociety.org/-/media/policy/Publications/2019/21-08-19-policy-briefing-on-teachers-of-computing.pdf>.
- Wing, Jeannette M. 2006. "Computational Thinking." *Communications of the ACM* 49 (3): 33–35. <https://doi.org/10.1145/1118178.1118215>.
-



Michael Lodi

michael.lodi@unibo.it

Michael Lodi è professore a contratto e borsista di ricerca presso il Dipartimento di Informatica – Scienza e Ingegneria dell’Alma Mater Studiorum – Università di Bologna e membro del Laboratorio CINI “Informatica e Scuola”. Studia le difficoltà e le metodologie per l’introduzione alla programmazione, gli aspetti filosofici

dell'Informatica come disciplina scientifica (es. pensiero computazionale, big ideas), e la didattica della crittografia. Insegna a studenti di informatica e di altre discipline, nonché agli insegnanti in abilitazione. Ha fatto parte del nucleo redazionale della proposta CINI di Indicazioni per l'insegnamento dell'informatica nella scuola. Tiene corsi di formazione per docenti e ha tenuto laboratori per bambini e ragazzi. È coautore di materiali didattici e guide per l'informatica nella scuola primaria.



Violetta Lonati

lonati@di.unimi.it

Violetta Lonati è ricercatrice presso il Dipartimento di Informatica dell'Università degli Studi di Milano, dove è tra i fondatori del laboratorio ALaDDIn e del progetto Bebras dell'Informatica, gara di informatica a squadre per le scuole. È membro del Laboratorio CINI "Informatica e Scuola" e ha fatto parte del nucleo redazionale della proposta CINI di Indicazioni per l'insegnamento dell'informatica nella scuola. Svolge attività di ricerca sulla didattica dell'informatica e si occupa di formazione degli insegnanti, temi su cui tiene corsi e collabora con la comunità di ricerca in didattica della matematica.