

## Coding e Pensiero Computazionale per il potenziamento delle competenze logiche e matematiche

Roberto Capone<sup>1</sup>, MariaRosaria Del Sorbo<sup>2</sup>, Paolo Musmarra<sup>1</sup>,  
Antonietta Esposito<sup>1</sup>, Ilenia Veronesi<sup>3</sup>

Dipartimento di Matematica, University of Salerno, Italy<sup>1</sup>  
Liceo “Leonardo. Da Vinci” Poggiomarino (NA)<sup>2</sup>  
Liceo “Mancini” (AV)<sup>3</sup>

{rcapone, antesposito, pmusmarra}@unisa.it  
marodel@gmail.com, ilaria.veronesi67@gmail.com

**Abstract.** L'articolo descrive le attività proposte nell'ambito di un progetto interdisciplinare, il “Liceo Matematico”, attivato in diverse regioni del Nord e Sud Italia. L'offerta didattica che il progetto propone è frutto di un lavoro congiunto dei Dipartimenti di Matematica, Informatica e Letteratura dell'Università di Salerno. Il paper descriverà nello specifico, le metodologie didattiche innovative utilizzate, le attività dei primi tre anni di sperimentazione del modulo di Informatica e i risultati degli studenti in termini di competenze logico-matematiche acquisite. Le innovazioni educative delle metodologie e le attività di sperimentazione proposte, sembrano confermare che abituare gli studenti al pensiero computazionale migliora anche sul rendimento delle altre discipline.

**Keywords:** Computational Thinking, Coding, Scratch, App Inventor

### Introduzione

Il “Liceo matematico” è un progetto sperimentale partito tre anni fa in Campania ad opera di un gruppo di ricercatori dell'Università di Salerno [1]. Fonda le sue radici dottrinali sulle idee filosofiche postmoderne in didattica della matematica e sulla teoria della complessità di E. Morin. Si articola in corsi aggiuntivi e di approfondimento rispetto ai normali corsi scolastici per ampliare la formazione dell'allievo al fine di svilupparne le capacità critiche e l'attitudine alla ricerca scientifica. Gli studenti che partecipano al progetto non hanno nel loro curriculum ordinario ore di informatica, pertanto, vengono avviati al coding e al pensiero computazionale fin dal primo anno attraverso attività integrative svolte in laboratorio, organizzati in piccoli gruppi in accordo con il fatto che “ogni funzione dello sviluppo culturale compare prima sul piano sociale e poi su quello psicologico, prima come categoria interpsicologica e poi come categoria intrapsicologica” (Vygotskij, 1987, p.11) [2].

## Quadro teorico

Il quadro teorico di riferimento è quello del Costruzionismo di stampo papertiano secondo cui è fondamentale la centralità dell'artefatto cognitivo come facilitatore per imparare e interagire con gli altri. Nell'ottica di Papert, l'apprendimento è un processo che avviene attraverso il ruolo attivo di chi impara. Il termine pensiero computazionale fu da lui usato per la prima volta nel 1980 [3] parlando di LOGO, il linguaggio di programmazione che sviluppò al MIT. Il pensiero computazionale, ovvero l'insieme dei processi mentali coinvolti nella formulazione dei problemi e delle loro soluzioni in modo che queste siano rappresentabili in una forma che possa essere efficacemente eseguita da un agente di elaborazione delle informazioni [4], è oggi ritenuta un'abilità fondamentale per tutti, non solo per gli informatici. Oltre a leggere, scrivere e far di conto, dovremmo aggiungere il pensiero computazionale alla capacità analitica di ogni bambino [5].

In una prospettiva didattica di matrice montessoriana, riteniamo che un approccio al *coding* aiuti lo studente a sperimentare in prima persona, a procedere per tentativi ed errori verso nuove soluzioni, a favorirne l'apprendimento come scoperta permettendogli di lavorare in autonomia senza l'aiuto dell'adulto. Altri aspetti importanti del coding sono la possibilità di individualizzare i percorsi di apprendimento oltre allo sviluppo e potenziamento della creatività, dei processi logici, dello sviluppo della concentrazione, dell'attenzione e della precisione. Infine, l'alfabeto del pensiero computazionale permette di coniare nuove parole, crea una nuova sintassi, forma il linguaggio che parliamo interponendosi così in maniera trasversale tra le diverse discipline favorendo lo sviluppo del pensiero logico e la logica del pensiero, sviluppando la creatività demandata, nella didattica tradizionale, alle materie umanistiche. Le competenze digitali sviluppate attraverso il pensiero logico sono fondamentali alla formazione dello studente come cittadino del mondo e contribuiscono a unificare i tre saperi (saper, saper fare, saper essere) in un unico sapere che è il saper agire [6].

## Organizzazione del modulo di Informatica

Prima di organizzare le attività didattiche per il modulo di Informatica, ci siamo chiesti come avremmo potuto rendere più facile l'accesso ai linguaggi di programmazione e al pensiero computazionale per coloro che non avevano una precedente conoscenza di informatica. Si è da subito pensato ad un approccio interattivo basato su attività di laboratorio attraverso cui gli studenti potessero acquisire e sviluppare competenze trasversali oltre quelle più specificamente informatiche. Si è deciso di utilizzare dei tools basati su linguaggi di Programmazione Visuale, in modo da consentire agli studenti di realizzare un prodotto finito. Nei primi due anni per acquisire familiarità con il coding, viene utilizzato *Scratch* [7] come linguaggio/ambiente di programmazione, suscitando un notevole interesse e contribuendo anche allo sviluppo di abilità di pensiero logico matematico. L'ambiente di Scratch è composto da una area detta di *stage* al cui interno possono muoversi i personaggi detti *sprite*. Uno dei motivi del successo di Scratch è dovuto al fatto che lo

studente non deve utilizzare un editor di testo per scrivere del codice, correndo il rischio di commettere errori di sintassi o di punteggiatura che in genere scoraggiano. Utilizzando Scratch, gli studenti hanno avuto la possibilità di focalizzare concetti di Matematica e Fisica già affrontati in corsi precedenti.

Il terzo anno è previsto un approfondimento delle tecniche di programmazione visuale orientato allo sviluppo di “applicazioni mobile” utilizzando App Inventor (AI) [8]. Ciò è stato fatto per consentire loro di diventare, in maniera consapevole, non solo “consumatori” passivi, ma soprattutto dei “creatori” attivi di tecnologia.

AI è un tool cloud-based, creato dal MIT, che permette di costruire e sviluppare app funzionanti direttamente dal web browser. E’ ormai un punto di riferimento soprattutto per chi desidera programmare la propria app, o realizzare un vero e proprio prototipo dell’interfaccia utente.

L’interfaccia grafica di AI consente di organizzare ogni aspetto della propria app senza la necessità di scrivere nessuna riga di codice testuale e permette agli studenti di vedere e testare ciò che loro sviluppano in modo rapido ed immediato subito dopo aver assemblato correttamente i blocchi di componenti diverse. E’ possibile effettuare il test dell’app utilizzando direttamente il proprio smart phone/tablet o un emulatore nel caso in cui lo studente non sia in possesso di un terminale mobile di tipo Android. Questo processo incrementa negli studenti il desiderio di fare e fare sempre meglio perché evitano di incorrere in possibili errori sintattici, molto frequenti, e spesso causa di scoraggiamento nel caso dei linguaggi di programmazione tradizionale. La parte di programmazione logica avviene attraverso l’utilizzo dei blocchi che sono raggruppati in colori in base al tipo di azione che il programma deve compiere.

Durante il quarto e quinto anno è prevista l’introduzione ai linguaggi di programmazione ad alto livello attraverso l’uso di Python per implementare alcuni algoritmi significativi di carattere matematico e informatico.

## **Come costruire app: steps e metodi**

In questa sezione mostreremo un possibile framework didattico per costruire una semplice app per creare una interfaccia utente che permetta l’accesso a dei dati personali attraverso uno screen di login e di autenticazione. In questo modo i dati personali verranno memorizzati nel proprio dispositivo mobile in modo affidabile e persistente [9]. Di seguito descriveremo i principali steps utilizzati per consentire agli studenti di costruire una propria mobile app con App Inventor.

- Il primo step è quello di aiutare gli studenti a stabilire, prima ancora di cominciare a lavorare con AI, cosa deve fare la app, quale deve essere il suo comportamento, le componenti visibili e non visibili necessarie, e settare le proprietà per ogni componente.
- Il secondo step è quello di programmare il comportamento della app, familiarizzando con i blocchi contenuti nel Blocks Editor, connettendoli in modo semplice, se compatibili tra loro, trascinandoli con la tecnica del drag and drop.
- E’ necessario, infine, testare e verificare il funzionamento della propria app, attraverso l’utilizzo dell’emulatore o con il proprio dispositivo mobile.

## Ricaduta Didattica

L'attività di formazione svolta nel "Liceo Matematico" ha una forte valenza didattica in quanto l'approccio non tradizionale, prevedendo numerosi momenti di interazione tra pari, è fonte di autoformazione degli studenti che diventano a loro volta formatori. Tra le varie attività sono, infatti, previsti seminari peer to peer nelle giornate di orientamento per gli alunni delle scuole medie. In questo contesto gli allievi hanno illustrato ai visitatori, in "speed sessions", i prodotti realizzati con Scratch.

Un altro momento importante di ricaduta didattica si è verificato durante il Primo Convegno Nazionale dei Licei Matematici che si è svolto all'Università di Salerno nel mese di settembre 2017. In questa occasione, gli studenti sono stati "docenti" protagonisti di laboratori di coding e di pensiero computazionale. In un contesto di classe destrutturata del rigore gerarchico che vede il docente avere il ruolo di "mentore" o di "facilitatore" degli alunni, le occasioni esposte hanno avuto la valenza di affinare le competenze trasversali come l'imparare ad esporre e ad argomentare i progetti realizzati e le attività svolte davanti ad un pubblico sempre diverso.

## Conclusioni e sviluppi futuri

Dall'esperienza dei primi tre anni di attività si è potuto osservare come il pensiero sistemico si sia rivelato trasversale alle varie discipline ed abbia, attraverso il paradigma della programmazione, permesso di rafforzare negli studenti il ragionamento logico, la capacità di astrazione, di valutazione oltre al pensiero algoritmico. Al termine di ogni attività si è somministrato un questionario di tipo affect, da cui si è evinto un elevato gradimento sia per i contenuti che per le metodologie didattiche utilizzate, pertanto per il futuro si pensa di continuare a progettare attività simili attraverso cui è evidente la corretta integrazione del saper e del saper fare.

## Riferimenti Bibliografici

1. Capone R., Rogora, E., Tortoriello, F.S.(2017) La matematica come collante culturale nell'insegnamento in *Matematica, Cultura e Società*.
2. Vygotskij, L. (1987). Il processo cognitivo. Raccolta di scritti a cura di Michael Cole, Sylvia Scribner, Vera John-Steiner, Ellen Souberman. Torino: Bollati Boringhieri.
3. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
4. Wing, J. (2011). Research notebook: Computational thinking—what and why? *The Link Magazine*. Pittsburgh
5. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
6. D'Amore, B. (2000). La Didattica della Matematica alla svolta del millennio: radici, collegamenti e interessi. *La matematica e la sua didattica*, 3, 407-422
7. <http://scratch.mit.edu>
8. <http://appinventor.mit.edu>
9. Musmarra, P. The safety on your own app with App Inventor, CISIS 2017, Torino.