



*Ministero dell'Istruzione e del Merito
Ufficio Scolastico Regionale per l'Emilia-Romagna*

STUDI E DOCUMENTI

Dicembre 2022

n.36

La programmazione e la robotica nella NGC

Fabio Bertarelli, Giovanni Govoni, Emilio Zilli

Introduzione

In un mondo caratterizzato da straordinarie innovazioni tecnologiche, programmazione e robotica educativa sono fortemente intrecciate e interconnesse tra loro, ricoprendo un campo affascinante dell'informatica e non solo.

La robotica è di fatto considerata un ottimo modo per introdurre i ragazzi all'apprendimento basato sulle STEM (scienze, tecnologia, ingegneria e matematica) sin dalla più tenera età.

La programmazione oltre a sviluppare il pensiero computazionale, sviluppa tutte quelle abilità legate all'ideazione e allo sviluppo del software. Gli elementi presenti in un codice possono essere letti e interpretati da persone secondo un processo logico o possono svilupparsi in un insieme di istruzioni che un robot può interpretare e seguire.

La programmazione e la robotica hanno poi subito nel tempo un "esodo" che, partendo da grandi centri del sapere, si è sempre più diffuso fino ad arrivare nelle stanze delle case degli studenti. I primi esempi di robotica e di programmazione per il Primo ciclo, con l'intento di diffondere la disciplina e non solo come elementi di ricerca universitaria, li vediamo sperimentare al MIT nel 1980 col diffondersi dei primi Personal Computer (in questo caso il sistema era basato su un Apple II)¹. Ad oggi, con qualche decina di euro, si possono acquistare semplici bracci robotici o sistemi a percorso programmabile. Questo ha facilitato non poco il diffondersi

¹ Vedi MIT LOGO for the Apple II (1981), disponibile sul sito Archive.org (<https://bit.ly/3VKlxfp>).

della robotica e della programmazione rendendoli alla portata di tutti. Questo effetto si sta progressivamente diffondendo anche nelle scuole.

Le Next Generation Classroom sono forse tra i primi ambienti di apprendimento che prevedono un uso comune, sistematico e "smart" in piena sintonia con le attività solitamente svolte in aula, a differenza di quanto avviene nei laboratori.

"Fare robotica in classe", ma come?

Pensare a un percorso per i docenti con un taglio pratico, efficace e adatto al contesto in cui si opera è utile e doveroso: se il docente si muove con competenza nella propria zona *comfort*, anche lo studente risentirà di questo clima positivo.

Oltre a questa importante premessa, il "come fare" va progettato e pensato in funzione dell'ordine di scuola in cui ci si trova.

La prima considerazione è di carattere logistico: le attrezzature medio-grandi richiedono spazi che spesso in aula non ci sono, mentre i dispositivi definiti "durevoli" (cioè non inventariabili e non consumabili) trovano una sempre più facile collocazione in aula. Questo però non deve intendersi come uno *standard* fisso. Nulla vieta di creare in classe una zona con un plotter 2D (Robot Cartesiano) o una stampante 3D se il docente pensa di farne un uso continuativo con gli studenti perché il suo stile di insegnamento risulta essere più efficace in queste condizioni.

La seconda considerazione è legata al curriculum: in funzione di un diverso grado di scuola si possono utilizzare dispositivi differenti e più adatti alle diverse esigenze dei discenti.

Esiste poi anche un altro modo di fare robotica ed è quella legata all'uso di simulatori.

Il processo ipotetico-deduttivo che sta alla base della robotica è invariato e, se la scuola ritiene utile utilizzare attrezzature più costose e performanti (si pensi in questo caso alle scuole secondarie di II grado, dove sempre più di frequente si vedono laboratori di automazione con bracci robotici ad uso industriale) è utile dapprima pensare al progetto simulandolo. Solo in un secondo momento lo si potrà trasferire sulla macchina per vedere ciò che realmente compie. Questa seconda fase è utile per potersi confrontare con le proprie ipotesi e rendere reale quanto già sperimentato al simulatore.

In estrema sintesi, sempre più corsi di specializzazione di alto profilo e sempre più dispositivi di alta e media gamma prevedono l'uso di simulatori per abbassare i costi di fermo-macchina tra il cambio di un progetto con quello successivo.

Destinatari

La programmazione è essenziale ora più che mai per preparare gli studenti al futuro. Quando gli studenti imparano le basi della programmazione, si apre loro un mondo completamente nuovo pieno di opportunità entusiasmanti. L'apprendimento non ha limiti di età, ma le giovani menti sono più attive e possono imparare cose nuove facilmente. Il periodo in cui i bambini iniziano a imparare a leggere e scrivere è l'età ideale per iniziare a programmare.

Il Primo ciclo, dove l'autoproduzione di contenuti e il pensiero creativo devono essere sempre stimolati in linea con i tempi, offre i destinatari ideali.

Successivamente, andando avanti con gli studi, si trovano studenti sempre più in possesso di competenze specifiche e questo permette un uso di strumenti più performanti con flessibilità e ambienti di programmazione che permettono non solo di pensare e programmare, ma anche di creare un proprio stile personale di programmazione.

Discipline e competenze disciplinari

Programmazione e robotica contribuiscono a favorire lo sviluppo di elementi di progettualità in ambiti interdisciplinari. Tuttavia, suddividendo per ordine di scuola, possiamo asserire che per la scuola dell'infanzia si riescono a coinvolgere tutti i campi di esperienza con attività *unplugged* e analogiche, per quanto riguarda la scuola primaria si possono impostare percorsi interdisciplinari con italiano, scienze, matematica, arte e immagine, tecnologia, storia, geografia, inglese, mentre per la scuola secondaria di I e II grado i percorsi tecnologici ed interdisciplinari verranno progettati in base alle esigenze specifiche degli alunni e delle alunne e dei rispettivi corsi di studio curriculari.

Attraverso lo studio dei principali costrutti della programmazione (algoritmo, istruzioni condizionali, funzioni) si sviluppa negli studenti di ogni ordine e grado il pensiero riflessivo e procedurale, stimolando la riflessione sull'errore come spunto di lavoro, incrementando capacità linguistiche sia orali che scritte.

Competenze trasversali

Tramite le attività di robotica e programmazione i bambini e le bambine acquisiranno forti capacità di pensiero critico e abilità di *problem solving*, imparando così a scegliere la soluzione ottimale per ogni problema che man mano si presenta.

Attraverso le attività di apprendimento pratico aumenteranno i livelli di concentrazione e attenzione degli studenti.

Altri aspetti salienti che vogliamo citare:

- flessibilità cognitiva: la tecnologia cambia ogni giorno e gli studenti devono essere più preparati ai cambiamenti tecnologici;
- lavoro di squadra: gli studenti lavorano in squadra per adattare i loro robot per competere in diverse sfide dinamiche;
- abilità di programmazione: oltre ad imparare a programmare righe di codice, scopriranno nuovi modi per risolvere i problemi;
- pensiero analitico: per affrontare grandi problemi di robotica, gli studenti devono suddividere il problema in segmenti e trovare la soluzione più adatta;
- capacità di comunicazione scritta e verbale: la comunicazione efficace è un'abilità assolutamente essenziale per tutta la vita dentro e fuori dalla scuola;
- pensiero ipotetico-deduttivo: programmare è come raccontare storie al futuro anziché al passato o al presente;
- pensiero divergente e creativo: le soluzioni possono essere molteplici e permettono l'affinare di stili risolutivi diversi.

Strumenti necessari

Hardware e Software

- Schermi Interattivi
- Rete cablata e Wifi in tutta la scuola
- Notebook, tablet e chromebook
- Robot didattici (ad esempio, Bee Bot, Blue Bot, Lego Spike, Lego Mindstorms, ecc.)
- Robot componibili e programmabili (ad esempio, dispositivi della Makeblock come Mbot, Mbot Ranger, Codey Rocky, oppure Lego WeDo 2.0, Lego Mindstorms, Dash and Dot, droni di terra e d'aria)
- microcontrollori e schede programmabili come Raspberry Pi e Arduino
- applicazioni software per la codifica basata su blocchi come Scratch, Blockly e VEXcode Vr
- applicazioni software per la codifica basata su testo: C/C++, Python, Java
- simulatori di processo
- ambienti di programmazione dedicati ad apparecchiature industriali

Metodologie

La scelta metodologica deve avere un approccio di tipo costruttivista e ludico: attraverso la programmazione i ragazzi sviluppano il pensiero logico computazionale, cimentandosi in attività nuove, come può essere la programmazione a blocchi, imparando un linguaggio di programmazione divertendosi. È quello che gli americani chiamano *learning by doing*: i ragazzi e le

ragazze mettono in pratica le nozioni apprese, imparano facendo, costruendo attività piacevoli e divertenti che motivano l'apprendimento.

La programmazione può avere nella scuola un approccio metodologico trasversale ed interdisciplinare, poiché è trasversale la competenza che consente di sviluppare. Attraverso applicazioni per il *coding* e strumenti di robotica educativa si può facilmente apprendere per scoperta, utilizzando strumenti concreti per acquisire nuove competenze e conoscenze.

Utilizzando il *problem solving* ed aiutando gli alunni ad eliminare l'ansia per l'errore, l'insegnante svolge un ruolo da regista/facilitatore pianificando percorsi e attività multidisciplinari e interdisciplinari facilitate dall'utilizzo all'interno del gruppo classe di una metodologia laboratoriale e del *cooperative learning*.

Valutazione

La riflessione sull'algoritmo utilizzato in fase di programmazione, condivisa con gli studenti, e la narrazione dei propri progetti consente al docente, attraverso l'utilizzo di strumenti come le *rubrics* per la valutazione e l'autovalutazione, il diario di bordo e la valutazione tra pari di tendere sempre di più ad una valutazione autentica.

Con l'uso dei simulatori e con l'idea delle Next Generation Classroom si possono creare in modo continuo, ripetuto o a *spot* veri e propri episodi di apprendimento situato, tesi alla preparazione degli studenti al mondo del lavoro, portando in classe le esperienze svolte in laboratorio o in contesti con attrezzature non utilizzabili in classe per motivi legati alla logistica e alla sicurezza.

Esempi e riflessioni

Segnaliamo alcuni articoli redazionali pubblicati nel sito del Servizio Marconi TSI dell'Ufficio Scolastico Regionale per l'Emilia-Romagna:

- *Sul coding a scuola... siamo a un bivio?* (<https://bit.ly/3VEikyU>)
- *Il Coding a scuola: lo sviluppo del pensiero computazionale, per un approccio attivo al mondo digitale* (<https://bit.ly/3FhBTHS>)
- *SteamOnBoard, piattaforma Makeblock per le scuole dell'Emilia-Romagna* (<https://bit.ly/3UsSwoh>)
- *Due giorni con Sam Aaron e il suo Sonic Pi a Bologna* (testo sul coding musicale) (<https://bit.ly/3XL7uJj>)
- *Educare con la robotica nella scuola dell'infanzia* (<https://bit.ly/3Ff6whh>)
- *Giornata avvio sperimentazione MakeBlock* (<https://bit.ly/3H1F8o5>).

Spunti ulteriori

Per approfondimenti è possibile consultare il webinar *Il Coding a scuola: lo sviluppo del pensiero computazionale, per un approccio attivo al mondo digitale*, a cura di Servizio Marconi TSI dell'USR E-R, Assessorato Scuola, Università, Ricerca, e Agenda digitale della Regione Emilia-Romagna, tenutosi nel novembre 2020 con interventi di Paola Salomoni, Stefano Versari, Enrico Nardelli, Alessandra Serra, Stefano Rini, Rita Trombini *"Il Coding a scuola: lo sviluppo del pensiero computazionale, per un approccio attivo al mondo digitale"*.

Per ulteriori spunti, riflessioni ed approfondimenti, è utile leggere anche *"Che cos'è il Pensiero Computazionale?"* (<https://bit.ly/3VKLzjt>) pubblicato nel sito *Programma il futuro*.

La piattaforma *Programma il futuro*² è fonte di numerose informazioni e consigli, anche per chi è alle prime armi. Si tratta di un portale promosso dal Ministero dell'Istruzione in collaborazione con il CINI (Consorzio Interuniversitario Nazionale per l'Informatica) che offre alle scuole italiane le risorse della piattaforma *code.org*³. Ogni anno sono qui disponibili nuovi contenuti relativi a temi quali *"Cittadinanza digitale consapevole"* (<https://bit.ly/3Urjw7I>) o *"Giocolieri del codice"* (<http://bit.ly/3XoM7MV>). Le attività sopra descritte si possono trovare nel sito italiano dedicato all'iniziativa *"L'ora del codice"* (<https://bit.ly/3Fio781>).

Chi invece ha già esperienza ed è interessato a promuovere e sviluppare un approccio creativo all'insegnamento dell'informatica, può esplorare risorse come le seguenti:

- *Scratch* (<https://scratch.mit.edu/>), piattaforma di programmazione attraverso la quale si possono creare storie interattive, animazioni e giochi. Per gli insegnanti è inoltre possibile richiedere un *account* docente con il quale poter creare *account* per gruppi di studenti e gestire facilmente progetti e commenti dei ragazzi. Registrarsi come docenti è molto semplice ed esiste una pagina dove la procedura è guidata (<https://bit.ly/3VL1yOu>).
- *Learning Creative Learning* (<https://lcl.media.mit.edu/>), corso *on line* del MIT di Boston per la comunità mondiale di insegnanti e di educatori, *designers* e sperimentatori che esplorano l'apprendimento creativo attraverso un corso di sei settimane, gratuito e sempre disponibile.
- *Computer Science First* (<https://bit.ly/3gOAWO3>), piattaforma di Google che ha l'obiettivo di rendere la programmazione semplice da insegnare e divertente da imparare.

² *Programma il futuro* (<https://bit.ly/3UINB8L>).

³ *Code.org* (<https://code.org/>).

- *Code with Google* (<https://bit.ly/3UoVzy2>), comunità in cui sono disponibili diversi percorsi di *coding*.
- *Computer Science Unplugged* (<https://bit.ly/3VojTRU>), sito con attività analogiche per imparare l'Informatica divertendosi.

Infine, altri siti da segnalare sono i seguenti:

- Portale del *Coding* e della *Robotica Educativa* INDIRE (<https://bit.ly/3Vgd9W9>);
- Articolo "Il coding a scuola per lo sviluppo del pensiero computazionale" dal sito delle prove INVALSI (<https://bit.ly/3B2EJOV>);
- Progetto L2TOR (Second Language Tutoring using Social Robots) (<https://bit.ly/3ueDNCT>).