

Dal codice al robot: un'esperienza alla scuola primaria

DIDAMATICA 2016

Tamburini Federica Carla
Scuola primaria "Vera Vassalle" (IC Marco Polo Viani)
Via Pistoia 66-68 55049 Viareggio (Lu)

Il presente contributo descrive uno scenario di apprendimento che ha visto l'introduzione del coding, prima, e della robotica in una fase successiva, all'interno di un progetto più ampio che ha previsto l'uso della tecnologia e della multimedialità nella didattica. Vengono descritte le motivazioni didattiche che hanno portato a questa scelta, la metodologia adottata, le fasi di lavoro ed alcune considerazioni sull'esperienza vissuta.

1. Introduzione

L'importanza del pensiero computazionale, come attitudine di pensiero verso la risoluzione di problemi, di strategie risolutive e di approccio alle fasi processuali, è ormai riconosciuta ed ha radici lontane.

Già alla fine degli anni '60 Seymour Papert introdusse la prima versione di LOGO, il software di programmazione della tartarughina pensato proprio come strumento per l'apprendimento e nel 1971 scrisse un articolo chiamato "Venti cose da fare con un computer" [Papert, 1972], dove il computer viene presentato soprattutto come una macchina per simulare e come ambiente di apprendimento che aiuta a costruirsi nuove idee. Il punto fondamentale di Papert è che devono essere gli studenti a "comandare" il computer e non viceversa ed è per questo che ha sviluppato il suo famoso linguaggio, usando una tartaruga giocattolo virtuale che poteva esser programmata. Purtroppo in Italia la "buona pratica" della programmazione a scuola non è mai riuscita a diventare sistema, rimanendo confinata a isolate (quanto interessantissime e significative) sperimentazioni e dobbiamo attendere il 2012, anzi, soprattutto il 2014, perché con la circolare MIUR dedicata, si sollecitino gli insegnanti a sperimentare il coding a scuola (Nota prot. 2937 del 23/09/2014): grazie all'iniziativa "Programma il Futuro", che riprende l'esperienza di successo avviata negli Usa nel 2013 e di cui utilizza anche i percorsi inseriti nel sito Code.org, la scuola italiana si è mossa decisamente verso l'attivazione e la valorizzazione del pensiero computazionale come arricchimento della persona

e dello stile cognitivo. Anche nella Buona Scuola se ne afferma la validità e anzi se ne incentiva la diffusione sin dalle prime classi della scuola dell'infanzia.

Questa visione è senz'altro in linea anche con la nostra personale idea di scuola ed è per questo che abbiamo pensato di introdurre questa attività in classe.

Lo scopo non è tanto imparare a programmare, anche perché così dicendo si limiterebbe la sua pratica alla disciplina informatica (che tra l'altro non è più presente nelle discipline della scuola primaria), quanto piuttosto sviluppare capacità logiche e cognitive utili in ogni area disciplinare.

Lo sviluppo successivo, cioè l'introduzione della robotica, è stata la conseguenza naturale: potendo validare il codice creato con una macchina che "si muove" i bambini hanno avuto la percezione reale di quanto progettato ed hanno potuto osservare e valutare immediatamente la correttezza della procedura. L'uso del robot è stato utile anche per attivare una riflessione più generale sul "linguaggio delle cose", cioè di cosa voglia dire imparare una modalità per poter interagire con macchine *smart* (anche il nostro *smartphone*) e di come il processore sia il vero fulcro di questo "dialogo".

2. Scenario di apprendimento generale

La classe in cui è stata inserita questa attività è una classe "tecnologica". Sin dalla prima classe i bambini hanno sperimentato a scuola attività laboratoriali, soprattutto di tipo manipolativo ma hanno anche potuto gradatamente familiarizzare con i computer. Oltre a questo, i bambini hanno familiarizzato con i programmi contenuti nella loro chiavetta. Tra il materiale iniziale, è stata infatti richiesta una chiavetta con almeno 2 gb liberi. In essa l'insegnante ha inserito una serie di programmi freeware e direttamente eseguibili senza bisogno di installazione. L'idea, il blocco iniziale del launcher e dei primi software è stata presa dal progetto *Luposuite* e successivamente la suite dei programmi è stata ampliata fino a raggiungere i 134 attuali.

Tale strumento è stato utilizzato sia nelle esercitazioni in classe alla LIM sia nei compiti a casa, in quasi tutte le discipline. Nel lavoro al laboratorio informatico l'uso della chiavetta ha ampliato le potenzialità di lavoro sia perché i bambini si sono trovati a disposizione molte più risorse di quelle installate sui pc, sia perché è stato possibile salvare i loro lavori direttamente sul supporto.

Dalla classe terza, quando i bambini hanno potuto essere più autonomi, è iniziato un progetto denominato "**Verso la classe 3.0**" che ha introdotto l'uso della tecnologia come integrazione della didattica.

All'interno di questo progetto i bambini hanno potuto conoscere ed utilizzare varie risorse (sia software sia online) con cui hanno realizzato prodotti in tutte le aree disciplinari. Per documentare le varie attività è stato creato anche un blog di classe (<http://classemultimediale.blogspot.it>) in cui è possibile visionare le attività man mano realizzate.

2.1 Metodologia utilizzata

Questo progetto si è basato in modo peculiare sulla didattica per competenze, riprendendo quindi le caratteristiche del costruttivismo e, in modo particolare, del *cooperative learning*. Ai bambini è stato proposto in modo quasi sistematico il lavoro a coppie o a gruppi cooperativi (con o senza l'uso della tecnologia), sia per incentivare la loro autonomia personale sia per promuovere lo scambio cognitivo tra gli alunni. Ad oggi le insegnanti possono affermare la bontà di questa proposta: la classe si mostra nel suo complesso come un buon gruppo di lavoro, affiatato e autonomo sia nel lavoro individuale sia in quello di gruppo. Questa buona progressione nella metodologia di lavoro fa sì che adesso i bambini possano creare i gruppi per interesse o affinità personali, ruotando autonomamente, per raggiungere comunque gli obiettivi prefissati.

Ai gruppi viene solitamente assegnato una parte di un lavoro più ampio, in modo che il prodotto finale sia la somma del lavoro dei vari gruppi. Al loro interno, i gruppi si dividono da soli gli incarichi in modo da coinvolgere tutti i componenti. Naturalmente, per arrivare a tale autonomia i bambini sono stati guidati dall'insegnante fin dalle prime classi, sia nella definizione degli incarichi sia nella consapevolezza di come gestire un lavoro di gruppo in un definito

3. Introduzione del coding: come e perché

Il coding è stato introdotto in classe terza proprio come esperienza di “diverso” approccio alle situazioni problematiche e di come le procedure informatiche siano in realtà applicabili anche a diversi contesti.

La scelta è stata fatta sia per la sua valenza generale sia per l'analisi della situazione di contesto.

A livello generale, siamo profondamente convinti che molti degli spunti cognitivi offerti dalla Computer Science permettano di raggiungere traguardi logici e di attivare aree di ragionamento che saranno elementi chiave per la cittadinanza digitale di questo millennio. Lo scopo non è certamente quello di creare piccoli “informatici” o, peggio ancora, quello di limitare e indirizzare il modo di pensare in un modo standardizzato e rigido, quanto al contrario quello di permettere di sviluppare la creatività del loro pensiero facendogli percorrere però una strada logica e processuale che possa arrivare alla creazione di prodotti strutturalmente ben costruiti, convinti che questa “competenza” possa comunque essere loro utile in futuro in ogni settore che affronteranno.

Per la situazione di contesto, le insegnanti avevano notato la tendenza diffusa in molti alunni a ragionare in modo frettoloso in ogni area disciplinare, ad arrivare in fretta a conclusioni che la maggior parte delle volte si rivelavano inesatte, senza soffermarsi a pensare anche a quali passaggi intermedi sarebbero stati necessari. Con l'introduzione di questa attività si è quindi inteso rafforzare la capacità di riuscire a seguire un percorso lineare che permettesse di raggiungere l'obiettivo (qualunque esso sia) percorrendo una strada fatta di gradini in successione.

Gli obiettivi nell'ambito delle Raccomandazioni del Consiglio d' Europa del 2006 sulle competenze chiave e inseriti all'interno della progettazione di classe, che hanno fatto da sfondo pedagogico al percorso sono stati:

- Imparare ad imparare, acquisire un proprio metodo di studio e di lavoro
- Risolvere problemi, costruire e verificare ipotesi
- Spirito di iniziativa e imprenditorialità, utilizzare e organizzare le conoscenze in modo creativo per ipotizzare e realizzare un progetto
- Individuare collegamenti e relazioni, collegare fatti e idee e organizzarle in modo logico utilizzando anche schemi, diagrammi
- Competenze sociali e civiche, instaurare rapporti di collaborazione con i pari, controllando il conflitto di idee nelle discussioni comuni e intervenendo con proposte produttive.

Con gli alunni è stato seguito un percorso deduttivo a ritroso, partendo dall'esperienza (presentando la programmazione a blocchi) e arrivando gradualmente al concetto di algoritmo.

Ai bambini è stato inizialmente proposto un percorso tra quelli offerti dal sito Code.org, in particolare il percorso "Flappy" ritenuto più immediato e adatto all'età degli alunni. Già con questa prima attività, realizzata a coppie, i bambini si sono entusiasmati ma hanno anche potuto osservare cosa vuol dire descrivere nel dettaglio una procedura, rispettandone i vari step processuali.

Al termine di questo percorso è stata fatta una riflessione di verifica insieme ai bambini in cui abbiamo potuto riflettere sul percorso fatto, facendo emergere quale è stata la cosa che più li ha colpiti: come forse era prevedibile, la cosa che ha colpito di più è stata la necessità di "scomporre" ogni passaggio, stando attenti ad ogni minima variazione.

3.1 L'esperienza con Scratch

Dopo aver sperimentato i percorsi sul sito di Code.org, la classe ha affrontato la programmazione con Scratch, il software ideato da Michael Resnick del MIT di Boston (Massachusetts Institute of Technology) che può essere utilizzato sia online sia offline. Nel nostro caso, anche per la connessione un po' incerta che avevamo a scuola, abbiamo scelto di installarlo sui notebook della classe.

Scratch riprende la visione educativa del "vecchio" Logo: i suoi punti di forza erano infatti la modularità, l'estensibilità, l'interattività e l'estrema flessibilità. La programmazione con il Logo non era mai fine a se stessa, ma sempre pensata in relazione ad un progetto e adattabile alle discipline più svariate, anche se la sua applicazione più frequente è sempre stata quella della geometria. A differenza di Logo però, Scratch è gratuito, non prevede un linguaggio specifico perché ogni istruzione è già organizzata in "blocchi" (suddivisi in colori a seconda della funzione che svolgono) con cui interagire tramite *drag'n'drop* e consente anche di poter condividere con una grande comunità le proprie creazioni (cosa che all'epoca di Logo non era possibile): questo vuol dire che dal sito è anche possibile visionare progetti già pronti, sia

per confrontarli con i propri, sia per osservare ed imparare la realizzazione di alcuni passaggi.

Scratch, rispetto ai percorsi strutturalmente più “chiusi” del sito di Code, ha permesso ai bambini di esprimere la loro creatività: già dalla primissima lezione, in cui l'insegnante ha dato i rudimenti di base e chiesto ai bambini di realizzare la semplice animazione di uno *sprite*, moltissimi di loro sono andati oltre, intuendo anche alcune possibili varianti e dando da subito un'impronta personale a quanto richiesto.

Gradatamente i ragazzi si sono cimentati con animazione più complesse, anche se ancora improntate ad algoritmi lineari (senza variabili né funzioni), con cui hanno realizzato brevi storie, piccoli cartoni animati o racconti.

Nel corso dei tre anni, attraverso Scratch sono state anche fatte esperienze particolari; ad esempio, la classe ha partecipato ad una sorta di corso online (“*Una giornata a Scratchtown*” proposto da Aulavirtuale.it) con cui ha potuto confrontarsi anche con altre classi e sperimentare la comunicazione mediata dalla tecnologia (come nel forum); ha partecipato (in classe quarta) ad un concorso bandito dalla regione Toscana (“*Wikid 2.0*”) in cui, attraverso un percorso specifico in step, è stato chiesto ai bambini di pensare e progettare uno scenario di scuola nel futuro (i bambini hanno voluto realizzare in ogni step proprio un'animazione con Scratch); e anche in classe quinta ha partecipato a due concorsi di programmazione. “*Alicejam*”, in cui era richiesta la creazione di un semplice videogioco e, ultimamente, “*Codi-Amo*” (organizzato dal MIUR tramite “Programma il Futuro”), dove la consegna è stata la creazione di una breve storia attraverso la programmazione di alcuni elementi.

Il coding e Scratch sono stati anche l'occasione di scambi e attività in continuità. Proprio in quest'anno scolastico, in concomitanza con l'European Week Code e della Settimana del PNSD, è stata organizzato nella nostra scuola un Open Day all'insegna del digitale: la nostra classe ha organizzato vari laboratori di programmazione a gruppi misti (coinvolgendo anche le due classi 2.0 della scuola media dell'Istituto comprensivo) realizzando prodotti inseriti in varie aree disciplinari: ad esempio, un racconto giallo (italiano), una tastiera colorata che suonasse (musica), un labirinto (geografia), ecc. prendendo spunto anche dal libro “*Coding*” [Ferraresso et al, 2015], da cui abbiamo attinto alcune idee di realizzazione e che è servito ai bambini per strutturare la progressione di alcune richieste più complesse.

Scratch è diventato adesso per i ragazzi un mezzo familiare con cui esprimersi e con cui poter realizzare molti tipi di “animazione” diversi.

4. Dallo schermo....al robot

Arrivati in classe quinta e dopo aver avuto negli anni la possibilità di esplorare il coding sotto varie sfaccettature, si è dunque pensato di far fare un passo ulteriore in questo percorso di riflessione cognitiva e di pensiero computazionale presentando anche un piccolo robot che consentisse di applicare quanto progettato e programmato sullo schermo.

Nonostante la varietà di proposte che ormai si trovano in commercio anche per essere utilizzati con gli studenti più piccoli (come l'apina Bee-Bot e altre), per lo scopo che volevamo raggiungere la scelta è ricaduta su un kit robotico che possedesse due caratteristiche:

- avere un software di controllo (le apine Bee-Bot, ad esempio, non necessitano di nessun software specifico)
- e che questo software non si discostasse troppo con quanto fatto finora con Scratch

È stato scelto quindi il kit *mBot*, (assemblato su piattaforma Arduino) che coniuga il costo contenuto (rispetto ad altre proposte del mercato) con un software a blocchi creato sulla stessa architettura di Scratch. Scaricando il software *mBlock* direttamente dal sito del produttore, è possibile realizzare algoritmi anche complessi e farli eseguire alla macchina, che può essere connessa al pc tramite bluetooth o tramite wifi (al momento dell'ordine del robot si può scegliere l'una o l'altra delle versioni).

Il passaggio dalla proposta del coding a quella della robotica è stato fatto a seguito di alcune considerazioni. In primo luogo il fatto che il robot, o comunque l'oggetto tecnologico e/o programmabile in generale, fa ormai parte della vita dei bambini: molti giocattoli hanno questa caratteristica di animazione o di interazione e i bambini maneggiano comunque ormai moltissimi oggetti smart con cui interagiscono, spesso senza rendersene conto. Questa familiarità, e forse anche questa inconsapevolezza, in questo caso ci aiutano in quanto i bambini imparano praticamente senza accorgersene (almeno fino a quando, aiutati dall'insegnante, non attivano un processo di metacognizione dell'attività svolta). Lo scopo dell'attività è stato quindi quello di trovare un "oggetto" tecnologicamente raffinato ma che permettesse un'interazione abbastanza semplice che potesse rendere i bambini soggetti attivi nella costruzione della loro conoscenza.

Inoltre, la robotica stimola sia la sfera dell'intelligenza cognitiva sia la dimensione affettiva, producendo quindi alta motivazione negli alunni.

Essa pertanto, attraverso anche l'importante mediazione del docente, si rivela un contesto importante in cui il sapere e il saper fare si possano coniugare per raggiungere concreti obiettivi formativi consentendo anche di sperimentare concretamente un "vero" linguaggio di programmazione.

4.1. L'interazione con i robot

L'introduzione al mondo dei robot è stata fatta prendendo spunto da un testo inserito nel libro di lettura, all'interno del testo fantascientifico, che trattava proprio della nascita di un robot; come attività di espansione veniva proposta la realizzazione di un piccolo robot utilizzando materiale di recupero (vedi Fig.1).



Figura 1 - i robot costruiti con materiale di recupero

Questa attività ha entusiasmato i bambini, che a casa hanno realizzato ciascuno il proprio manufatto. Al rientro in classe questo ha permesso di attivare una riflessione più ampia sulle caratteristiche proprie di un robot, su cosa potesse voler dire “intelligenza artificiale” e su quali potessero essere i “robot” che conosciamo intorno a noi.

L'insegnante ha successivamente introdotto in classe *mBot*, illustrandone le caratteristiche salienti (sensori, processore, ecc.) e lasciando che i bambini potessero osservarlo e manipolarlo. Le riflessioni successive, sulla differenza tra questo robot e quelli costruiti da loro, hanno portato alla naturale conseguenza di interrogarci su come poter fare per “dialogare” con questo oggetto e di cosa sia necessario perché questo “dialogo” possa essere efficace. Ci ha aiutato in questo anche la visione di un breve filmato del prof. Bagliolo dell'Università di Urbino dedicato proprio al “linguaggio delle cose” e di cosa sia un oggetto “smart” (<https://www.youtube.com/watch?v=AEXF33EgH0w>).

Insieme alla LIM abbiamo quindi osservato il software con cui poter interagire con la macchina, notando immediatamente la grande affinità con Scratch, a cui i bambini erano già abituati (vedi Fig.2).

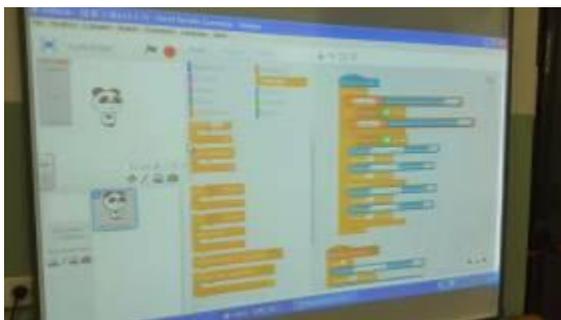


Figura 2 - Un esempio di codice realizzato con mBlock

Questo ha permesso di poter creare facilmente insieme una piccola successione algoritmica lineare (accensione e spegnimento dei sensori luminosi in sequenza, movimento avanti e indietro, attivazione del sensore del suono), che abbiamo poi verificato e validato immediatamente attraverso il collegamento del robot al computer.

Dal punto di vista educativo in generale, questa attività, pur semplice di per sé, ha permesso di tastare concretamente quanto detto nel paragrafo precedente su come la robotica possa attivare anche la componente affettiva e relazionale nei bambini.

Il robot che avevano prima toccato e manipolato come “oggetto”, non appena collegato al computer è diventato...,”vivo”, suscitando una reazione istintiva di relazione con “lui”, che è stato trattato quasi come fosse un piccolo animaletto. Sfruttando la connessione bluetooth, e quindi scollegandolo fisicamente dalla macchina, ciò è divenuto ancora più evidente: i bambini “chiamavano” a sé il robot e lo salutavano come a cercare un contatto. Alla fine, la naturale conseguenza è stata quella di dargli un nome e *mBot* è diventato “Multy”, la mascotte della classe.

4.2. Programmazione del robot e avvio della metacognizione

In una fase successiva, divisi in coppie di lavoro, ai bambini è stato chiesto di provare a realizzare in autonomia una piccola stringa di istruzioni per poter interagire con il robot utilizzando i notebook di cui è dotata la classe (vedi Fig.3).



Figura 3 - creazione del codice tramite lavoro collaborativo

Forti della loro pregressa esperienza con Scratch, non è stato difficile per loro poter realizzare qualcosa di efficace anche se in qualche caso, per la foga di voler vedere subito l’effetto finale, la tendenza è stata quella di “saltare” qualche passaggio logico.

La validazione del codice inserito, fatta attraverso il collegamento del robot (vedi Fig.4), ha permesso ai bambini di essere immediatamente consapevoli dell’efficacia dell’istruzione inserita e della bontà del percorso logico seguito, creando evidenti e comprensibili manifestazioni di soddisfazione ed entusiasmo.



Figura 4 - validazione del codice collegando il robot

Tutti gli alunni, compreso i ragazzi con più difficoltà, sono riusciti a completare la consegna e questo ha ancora di più dimostrato la reale validità di questo tipo di approccio cognitivo, che permette di arrivare – gradualmente – alla scomposizione di un problema (*“come fare per farlo muovere o suonare?”*) in una serie di step processuali in cui poi ogni alunno possa trovare una strada logica da seguire passo passo.

A seguito di questa attività sperimentale, è stata avviata successivamente una riflessione induttiva che ha portato idealmente a compimento anche il percorso sul coding iniziato negli anni precedenti e arrivando a comprendere cosa sia un algoritmo e di come esso possa essere “scombinato” e ma anche “ricombinato” con altri. Ai bambini è stata chiesta anche la stesura di un testo in cui essi potessero raccontare l’esperienza, facendo emergere in modo libero le loro considerazioni e la loro percezione di quanto affrontato. Nei testi è emersa, a volte anche stupendo la stessa insegnante, una grande consapevolezza del percorso fatto, di cosa voglia dire “programmare” e di come spesso sia difficile poter parcellizzare un problema in un percorso, segno che anche l’aspetto più astratto e metacognitivo è stato attivato.

5. Conclusioni

A livello generale dopo tre anni in cui i bambini hanno dunque lavorato, a vari livelli, anche con un approccio computazionale possiamo dire che questo tipo di attività ha portato sicuramente benefici, sia in termini di modo di ragionare sia come stile di apprendimento.

I bambini sono sicuramente più riflessivi e più logici rispetto al loro iniziale modo di lavorare; pur con le naturali differenze individuali, mostrano generalmente di aver appreso uno stile di apprendimento in cui il processo viene rispettato e compiuto in modo coerente e questo in tutte le aree disciplinari (dalla risoluzione dei problemi in matematica alla pianificazione e stesura di un testo correttamente strutturato in italiano, dall’esposizione logica di una narrazione storica alla spiegazione di un percorso coerente e ragionato tra le regioni italiane, e così via).

Certamente tutto ciò non è solo merito del coding (o della robotica) ma molto più probabilmente dell'averlo inserito all'interno di una metodologia di lavoro che ha incentivato l'autonomia dei bambini e ha cercato sempre di valorizzare e far uscire le potenzialità di ciascuno. Ma è indubbio che questo tipo di attività abbiano aiutato molto a fare questo perché i bambini hanno appreso uno stile di lavoro più rigoroso in modo quasi inconsapevole, perché per loro lo scopo era quello di far animare uno "sprite" di Scratch o far accendere le luci del robot e tutto ciò era possibile solo conoscendo "le regole" del loro linguaggio.

Oltre agli aspetti di costruzione logica fin qui sottolineati, un'altra delle caratteristiche fondamentali di un'attività di coding, che aiuta moltissimo lo sviluppo del pensiero, e che ha contribuito certamente nel rendere efficace questo percorso, è quella della gestione e del superamento dell'errore.

Molto spesso, infatti, le procedure inserite non funzionano, ciò che dovrebbe funzionare non funziona e anche se a prima vista tutto sembrerebbe corretto la validazione del codice non riesce e tutto rimane fermo oppure si comporta in un modo inaspettato o scorretto. Andare alla ricerca dell'errore, compiendo quello che in termine tecnico si chiama *debugging*, è un'attività non facile ma fondamentale nel pensiero computazionale e per i ragazzi è un "allenamento" importantissimo, non solo, come è evidente, a livello logico ma anche permette di affinare e sviluppare in modo diretto e immediato il processo di revisione, autocorrezione e autovalutazione, molto difficile da acquisire per i bambini che spesso infatti non riescono a trovare gli errori nei propri lavori più tradizionali (ad es. in un testo). Già lo stesso Papert sottolineava la grande importanza della gestione dell'errore: egli affermava infatti che l'unico modo per imparare in modo significativo è quello di prendere coscienza dei propri errori.

Possiamo dunque affermare che gli obiettivi generali raggiunti da questa attività siano stati:

- la stimolazione della creatività e dell'espressione personale
- le ricadute cognitive ed interdisciplinari
- il riconoscimento delle abilità individuali e la crescita dell'autostima
- la comprensione del percorso seguito
- l'accettazione e la comprensione dell'errore come stimolo per il superamento di una difficoltà

Recentemente è stato fornito ai genitori un sondaggio conclusivo relativo al progetto triennale portato avanti dalla classe; tra le varie domande, una ha chiesto, sentendo o vedendo anche la reazione dei loro figli, quale fosse stata l'attività che gli aveva colpiti di più: moltissimi hanno risposto proprio l'uso di Scratch e la robotica, segno che molto probabilmente gli stessi bambini hanno vissuto questa come un'attività coinvolgente e stimolante e che anche i genitori hanno potuto osservare che ciò che magari all'inizio poteva sembrare come una cosa che esulava dal normale programma didattico, si è invece rivelata una scelta didattica precisa ed efficace.

Bibliografia e Sitografia

Ferraresso, Colombini, Bonanome, Coding Programmare è un gioco 4/5, DeA Scuola. De Agostini, 2015

Papert S., Salomon C. "Twenty things to do with a computer", in Educational Technology Magazine, Englewood Cliffs, New York, 1972 (reperibile direttamente al sito <http://www.stager.org/articles/twentythings.pdf>)

Code.org, <https://studio.code.org/>

Programma il futuro, <http://www.programmailfuturo.it/>

Scratch, <https://scratch.mit.edu/>

Kit robotico Mbot, <http://www.makeblock.cc/mbot/>

Software di controllo Mblock, <http://www.mblock.cc/>