

TEMA

# Integrare Coding e Pensiero Computazionale nella didattica

**a cura di Equipe Formazione Digitale:**

**Mario Gabbari – Roberto Gagliardi**

**Antonio Gaetano – Daniela Sacchi**

mm.gabbari@tiscali.it – gagliardi\_roberto@tiscali.it –  
antoniopasquale.gaetano@fastwebnet.it – danisacchi94@gmail.com

Docenti formatori che operano presso OPPI Milano - <http://oppi.it/>

**Keywords :** Coding – Pensiero computazionale – Didattica – Scratch – Intelligenza emotiva

*“ insegnando all’elaboratore a pensare, i bambini si lanciano in un’esplorazione del loro stesso modo di pensare”.*

Papert, Mindstorms, 1980

La vita lavorativa di quasi tutti, è ormai permeata dall’utilizzo di software specialistici dedicati: i medici usano appositi software, gli avvocati oltre ad avvalersi di sistemi di gestione informatici fanno anche un grande uso delle banche dati informatiche (infinità d’informazioni necessarie per un esperto del settore), la sanità pubblica e quasi tutte le imprese commerciali utilizzano complessi software per la gestione della clientela, per la contabilità e per la gestione del magazzino. Per ogni attività è pertanto progettato e realizzato un software che ha bisogno di essere programmato e gestito in maniera tale da svolgere i compiti che gli sono richiesti. Siamo circondati dalle



Figura 1 - Interazione Tecnologie digitali

tecnologie digitali e la



Papert e Wallace Feurzeig<sup>4</sup> si incontrarono e, insieme allo scienziato ed esperto di Lisp, Cynthia Solomon, crearono la prima versione di Logo<sup>5</sup> nel 1967. Nacque così il primo linguaggio di programmazione per bambini, che consisteva nel poter spostare una piccola tartaruga con semplici istruzioni. Originariamente la Tartaruga era un robot che si muoveva su una superficie tramite comandi impartiti attraverso un computer. In seguito divenne uno strumento grafico, fu trasferita sul monitor del computer e usata per disegnare e creare immagini.

L'interesse verso il coding ha finalmente convinto molte nazioni a sviluppare un'informatica come creatori di programmi piuttosto che semplici utilizzatori (*lato maker* oltre che *consumer*) e a creare, in particolare, dei programmi didattici dedicati rivolti alle scuole, in modo da aiutare a formare gli esperti di domani.

Secondo il prof. Alessandro Bogliolo<sup>6</sup>: *"Questa capacità merita di essere coltivata e applicata in modo interdisciplinare perché costituisce una sorta di fertilizzante che prepara il terreno per l'uso consapevole della tecnologia e per la comprensione profonda degli aspetti computazionali delle attività che svolgiamo e della realtà che ci circonda. Il coding si presta a eccezionali applicazioni pedagogiche perché presuppone (e quindi induce) una comprensione profonda del procedimento che s'intende affidare a una macchina."*



Figura 3 - Alunna al PC

Il Coding offre la possibilità di realizzare una straordinaria esperienza sviluppando il ragionamento, la fantasia e le personali abilità per realizzare qualcosa. Gli alunni sono introdotti al pensiero computazionale (*capacità di pensare in termini di sequenze e regole*) attraverso l'immedesimazione nel linguaggio universale della *programmazione visuale a blocchi*<sup>7</sup>, utilizzata oggi anche nelle migliori università, tenendo però presente che non deve rappresentare solo la mera programmazione dei computer, ma un modo di affrontare le sfide e i problemi. Con l'alternarsi tra la formazione didattica, secondo le diverse metodologie e le simulazioni pratiche, gli studenti riescono a modulare e variare le tecniche di apprendimento, in modo da assimilare la materia e sedimentare le conoscenze acquisite. Il coding consente di far apprendere le basi della programmazione informatica, insegna a *"colloquiare"* con il computer, a impartire alla macchina comandi in modo semplice e intuitivo con una modalità che implica poca teoria e tanta pratica. Gli allievi tramite la diffusione del coding nella didattica, anche a partire dall'infanzia, apprendono i valori della cooperazione e condivisione, comprendono l'importanza della collaborazione tra loro per risolvere uno stesso problema, mettono a disposizione le conoscenze e le scoperte, accettando i suggerimenti anche dai più bravi per i propri elaborati.

<sup>4</sup> Papert fondatore del laboratorio di intelligenza artificiale del MIT, Wallace Feurzeig direttore di BBN Technologies, una società leader che ha avuto un ruolo di primo piano nella creazione di tecnologie come l'e-mail, VoiceIP

<sup>5</sup> Una versione attualmente in uso del LOGO è "Micromondi", applicata anche alla robotica dei mattoncini LEGO. Per chi vuole approfondire l'argomento LOGO segnaliamo il link: [http://it.wikipedia.org/wiki/Logo\\_%28informatica%29](http://it.wikipedia.org/wiki/Logo_%28informatica%29)

<sup>6</sup> Docente ordinario, coordinatore della Scuola di Scienze e Tecnologie dell'Informazione dell'Università Degli Studi di Urbino Carlo Bo – Dipartimento di Scienza pure e applicate. Autore di corsi e di libri sul Coding.

<sup>7</sup> Nei sistemi di programmazione visuale le singole istruzioni sono rappresentate da blocchi colorati che si possono trascinare in un'area di lavoro (*drag and drop*). I blocchi possono essere combinati tra loro in modo da comporre una sequenza d'istruzioni, che costituisce il programma. Sono preferiti dagli insegnanti perché la programmazione a blocchi permette di concentrarsi sul procedimento e sul ragionamento, non tenendo in considerazione la correttezza del linguaggio.

Il coding rappresenta una palestra per il pensiero computazionale ed è uno strumento utile all'analisi metacognitiva in quanto offre agli studenti una rappresentazione visiva dei processi sottesi ai loro ragionamenti.

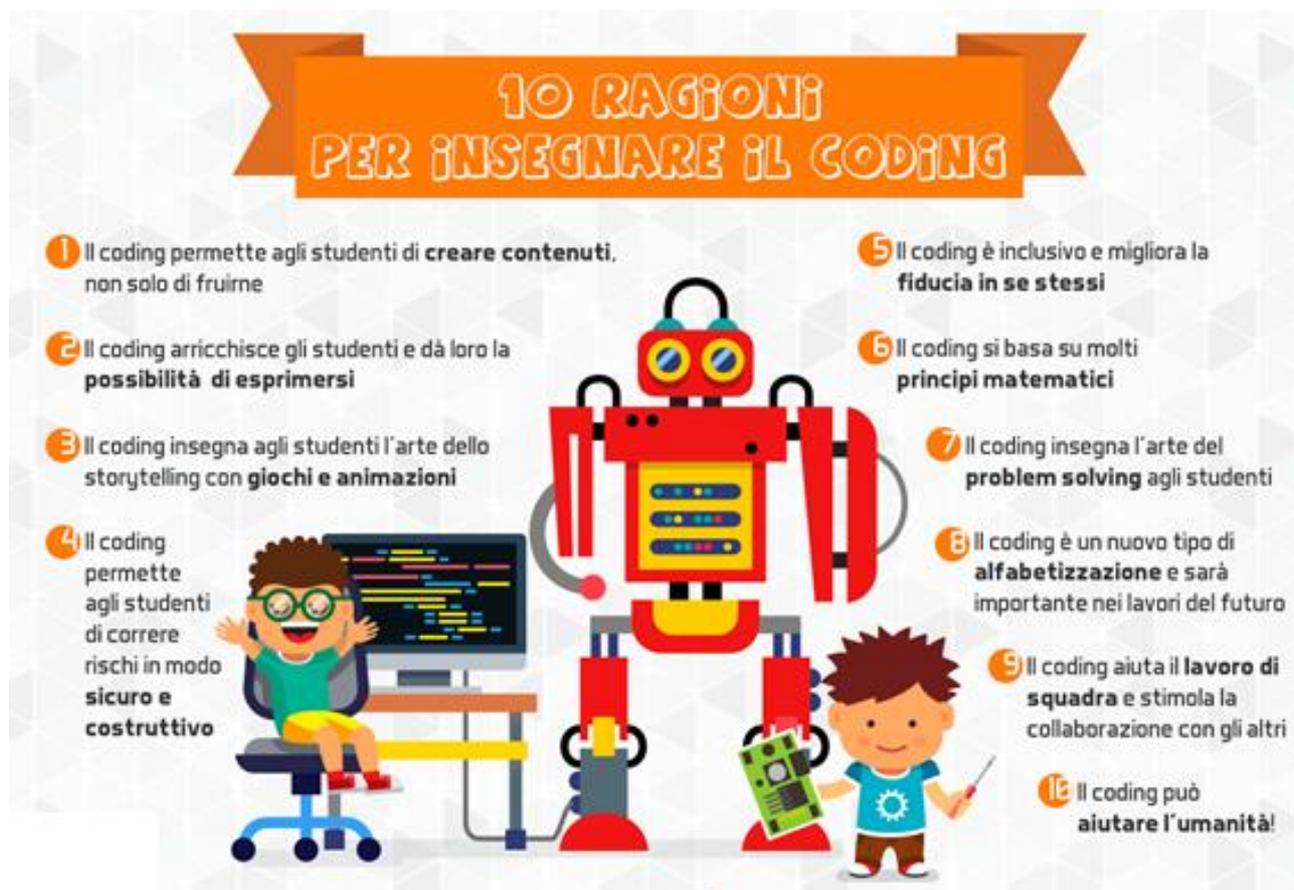


Figura 4 - Infografica riepilogativa delle ragioni del Coding (DEA Scuola)

## Coding e pensiero computazionale in aula

Il Coding è anche un modo divertente e appassionante per portare il pensiero computazionale in classe, poiché consente di esprimere se stessi e permette allo studente di migliorare le proprie capacità di logica e analisi; è anche un potentissimo strumento per sviluppare e realizzare progetti e percorsi creativi. Permette di giocare e sperimentare, *“senza paura di sbagliare”*; l'errore deve essere accettato come parte integrante dell'apprendimento e del processo educativo, inoltre diventa un'opportunità per scoprire nuove idee per arrivare alla soluzione (processo euristico). Spesso negli alunni e non solo (spesso anche negli adulti), il timore di sbagliare o di non riuscire, frenando l'accettarsi e il rendersi disponibili (*sapersi mettere in gioco*), può diventare un blocco mentale che accompagna l'alunno durante tutto il proprio percorso scolastico. L'errore può essere trasformato in un momento costruttivo di valutazione, di apprendimento e di crescita, purché non sia affrontato con pessimismo o, ancora peggio, con senso di colpa. Allora può diventare occasione di avanzamento e così l'errore riconquista una valenza positiva per andare oltre (stimolo al miglioramento).

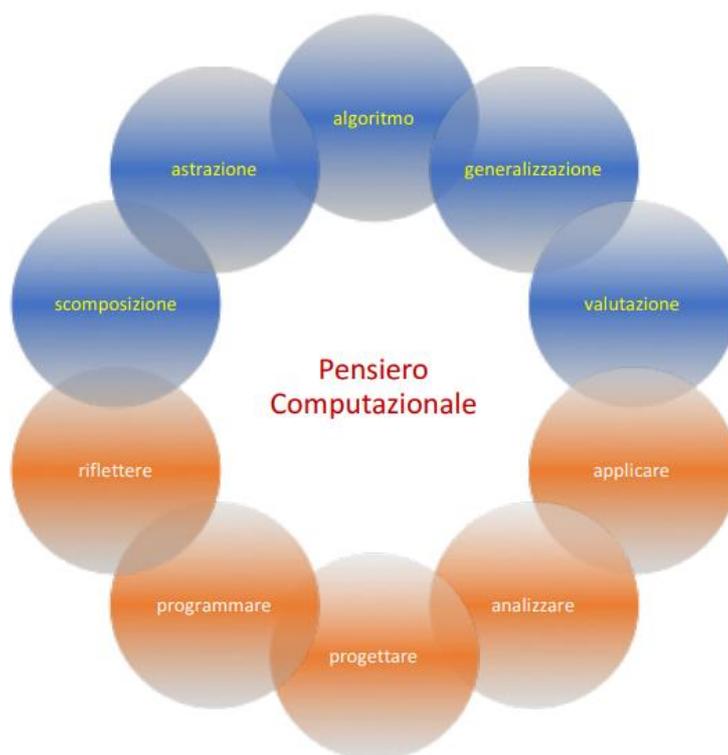


Figura 5 - Computational thinking - A guide for teachers (CNR)

L'utilizzo di questi software di programmazione (es. Scratch<sup>8</sup>) da parte degli studenti richiede la conoscenza dell'uso dei "mattoncini" della programmazione: *le funzioni, le variabili, le iterazioni, le condizioni da rispettare, ecc.* Stimolare gli alunni ad apprendere i meccanismi logici è anche utile ai fini del curriculum scolastico e personale, inoltre questi programmi offrono suggerimenti e opportunità per collaborare a esprimersi attraverso la polivalenza della tecnologia.

L'esperienza raccolta e costruita in questi anni di corsi di formazione rivolti ai docenti dei vari ordini di scuola, dall'infanzia alle superiori, ha consentito e permesso alla nostra *Equipe di formazione digitale*<sup>9</sup> di conoscere e approfondire le dinamiche e le

necessità che accompagnano questo genere di formazione e competenza che diventa fondamentale per integrare le idee computazionali nelle diverse discipline e in attività di programmazione funzionali alla didattica interdisciplinare. Il coding richiede di essere introdotto nella scuola come attività didattica trasversale poiché è trasversale anche la preparazione che consente di espandere le conoscenze. Si può quindi sostenere che il coding è un'attività



Figura 6 - Alunni al PC

sicuramente utile a sviluppare proposte didattiche innovative e inclusive, e a valutare le competenze. L'attuazione di una formazione docente attenta e indirizzata alla

<sup>8</sup> Scratch è stato ideato dal gruppo del Lifelong Kindergarten del MIT, guidato da Mitchel Resnick, e si ispira alla teoria costruzionista dell'apprendimento di Seymour Papert. Software gratuito specifico per la fascia d'età 7-16 anni. Permette di realizzare in modo semplice e veloce testi, suoni, animazioni, storie interattive, quiz, videogiochi, ... il limite è solo nella fantasia dell'utente e nel desiderio di imparare e sperimentare.

<sup>9</sup> L'Equipe svolge la sua attività di formazione on line e in presenza presso l'OPPI di Milano.

dimensione umanistico-espressivo-creativa è da considerarsi indispensabile alla pratica del pensiero computazionale, nella realtà, è spesso sottovalutata e/o non adeguatamente tenuta in conto e avvertita universalmente.

Edgard Morin, nella sua opera *"La testa ben fatta"*, invita gli insegnanti a considerare la complessità, e dice che *"la cultura, non solo è frammentata in parti staccate, ma anche spezzata in due blocchi"*<sup>10</sup>. Da una parte si pone la cultura umanistica, dall'altra la cultura scientifica; la prima riflette sui fondamentali problemi umani, la seconda divide i campi della conoscenza, produce scoperte straordinarie, ma non promuove la riflessione sul destino umano e sul divenire della scienza stessa. E' essenziale pertanto prendere in esame la proposta di considerare e realizzare anche una cultura di base filosofica e logico-matematica, attenta anche agli sviluppi linguistico-semiotici (*linguaggi e sistemi di segni naturali-artificiali*). Tale proposta deve essere esposta, narrata, raffigurata come il naturale cambiamento storico del pensiero umano universale/particolare-locale, resa affascinante e intrigante anche per i livelli di scolarizzazione primaria.

Ancora oggi la diffusione del coding e del pensiero computazionale non utilizzano a sufficienza, rispetto ai fini educativi, la conoscenza, la diffusione e la consapevolezza dell'opera di geni-benefattori dell'umanità che realmente hanno prodotto ciò che oggi si definisce comunemente col binomio *"coding-pensiero computazionale"*<sup>11</sup>. Spesso per cause legate ai vari marketing e agli sviluppi di qualche genere, si è andati nella direzione opposta, assegnando meriti e fama ad altri personaggi, che sicuramente hanno avuto un ruolo nello sviluppo delle nuove tecnologie, ma altrettanto hanno beneficiato di una sovraesposizione presso l'opinione pubblica.

## Coding e docenti

Nelle indicazioni<sup>12</sup> rivolte ai docenti si definisce il pensiero computazionale come un *"processo cognitivo e di pensiero che coinvolge logiche e ragionamenti attraverso i quali i problemi sono risolti e gli artefatti, le procedure e i sistemi compresi."* Le abilità di riflessione computazionale devono diventare parte integrante della struttura della personalità, infatti: *"si riferiscono a capacità di pensare e risolvere problemi in tutto il curriculum e nella vita in generale"*. Oggi è indispensabile aggiornarsi e adeguare le proprie competenze ai cambiamenti in atto nella società: in particolare, una buona conoscenza dell'informatica e un corretto approccio alle nuove tecnologie digitali forniscono nuove opportunità nella vita didattica e professionale.

Il compito dell'istituzione scolastica e dei docenti è quindi quello di far conseguire le competenze necessarie attraverso l'acquisizione di una cultura digitale che ancora fa fatica a diffondersi nella scuola ad eccezione di alcune ottime situazioni sperimentali.

La presenza dei PC si è ormai diffusa in modo omogeneo e generalizzato, pertanto avere familiarità con i concetti di base dell'informatica come materia scientifica è un nuovo elemento critico e vitale del processo di formazione e di cittadinanza per tutti.

Anche la didattica dei docenti deve assolutamente prendere in carico una concezione digitale, poiché le situazioni in cui agiscono prevedono sempre più l'uso di strumenti

<sup>10</sup> Morin E., *La testa ben fatta. Riforma dell'insegnamento e riforma del pensiero*, cit., pp. 89-90 e seguenti.

<sup>11</sup> Il primo a utilizzare la locuzione computational thinking è stato il matematico e informatico sudafricano Seymour Papert, già nel 1980 nel suo celebre libro *Mindstorms*, Vedi anche l'articolo suggestivo di Jeanette M. Wing, direttrice del Dipartimento di Informatica della Carnegie Mellon University, del marzo 2006 alla pagina <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>

<sup>12</sup> Vedi Computational thinking - A guide for teachers © Copyright 2015 Computing At School - © Copyright 2016 Consiglio Nazionale delle Ricerche (<http://community.computingschool.org.uk/resources/2324>)

digitali integrati con quelli tradizionali, consentendo così la creazione di nuovi ambienti didattici più dinamici, coinvolgenti e molto inclusivi. Bisogna sfruttare le opportunità che il mondo digitale mette loro a disposizione, poiché consente nuove modalità di approccio alla didattica che possono facilitare e aiutare l'apprendimento. Di tutto questo ne beneficiano gli stessi docenti e di riflesso anche gli alunni, "nativi digitali". Si rileva che i "digital natives" sono solamente coloro che sono nati in una società digitale e che è errato suggerire una loro capacità di comprensione "innata" di tale mondo. L'utilizzo di Internet, dei dispositivi mobili, ecc da parte loro, è, spesso, non consapevole, ma prevalentemente indirizzato ai "social media" e non all'apprendimento. La professionalità e la volontà dei docenti in questi ambiti devono però essere stimolate e potenziate da una rivalutazione morale ed economica del ruolo docente e del valore aggiunto che un docente "informatizzato" può dare al proprio insegnamento, alla propria disciplina e alla società.

## Il software per il Coding

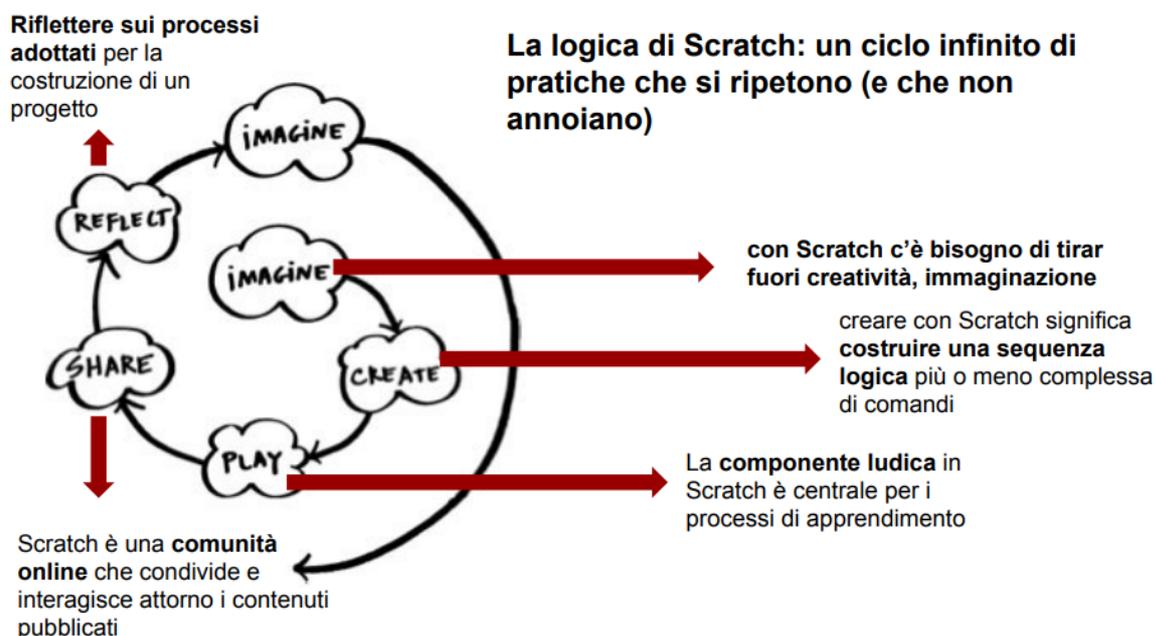
Tra i diversi software didattici disponibili e gratuiti per insegnare il Coding, sicuramente il programma Scratch<sup>13</sup>, rappresenta uno tra i software più interessanti, poiché semplice, intuitivo, accattivante ed è inoltre tra i più diffusi e utilizzati nel mondo scolastico. Si basa su un linguaggio di programmazione a blocchi e consente a chiunque di realizzare dei contenuti interattivi digitali. Le caratteristiche di Scratch rendono questo software particolarmente adatto a promuovere un apprendimento attivo e costruttivo, sia a livello individuale sia all'interno di un gruppo o di un ambiente collaborativo. Con Scratch sono certamente agevolati l'apprendimento di concetti, le pratiche e le prospettive del pensiero computazionale. Tuttavia Scratch rappresenta anche uno strumento di mediazione didattica per gli apprendimenti disciplinari. L'OPPI di Milano e l'Equipe Formazione digitale<sup>14</sup> hanno predisposto nei propri corsi di formazione l'impiego del software Scratch che è adottato come strumento di supporto alla progettazione didattica multidisciplinare e interdisciplinare. Il punto, naturalmente, è che l'utilizzo delle tecnologie digitali, come di qualunque altro strumento di mediazione didattica, non si giustifica di per sé, ma solo se è inserito nella visione di una didattica consapevole cioè pensata per questo scopo.



Figura 7 - Docenti informatizzati

<sup>13</sup> Software Scratch versione on line : <https://scratch.mit.edu/projects/318997608/editor>

<sup>14</sup> L'Equipe svolge i suoi corsi di formazione presso l'OPPI (Organizzazione per la Preparazione Professionale Insegnanti) di Milano vedi <http://oppi.it/corsi/corso-coding-pensiero-computazionale/>



Andrea Patassini 2014 15

Considerando il rapporto fra apprendimenti scolastici ed extrascolastici, è frequente che, una volta appreso l'utilizzo di Scratch, alcuni alunni spesso si sentano coinvolti al punto di adoperarlo regolarmente anche presso la propria abitazione, intraprendendo un percorso di approfondimento personale che spesso può portare al conseguimento di competenze anche molto approfondite in una logica di apprendimento permanente. Tali esperienze possono poi essere trasferite come un "divertimento" anche all'interno della propria classe, sviluppando così nuovi accorgimenti per interagire all'interno della comunità di apprendimento e perseguendo, allo stesso tempo, metodi didattici (strategie) legati all'individualizzazione e alla personalizzazione dei percorsi.

## Attività didattiche collaborative progettate con Scratch

- **Operare in modalità didattica sociocostruttivista<sup>16</sup>**

Nel metodo dell'apprendimento creativo, di nascita costruzionista<sup>17</sup>, la collaborazione, la cooperazione e la condivisione sono tutti fondamenti vitali che concorrono ad agevolare la costruzione della conoscenza e sono a loro volta il risultato delle interazioni sociali che si compiono all'interno di una comunità scolastica di apprendimento. La progettazione didattica, la ricchezza degli scambi comunicativi e affettivi, la costruzione collettiva del sapere, il rendere gli studenti protagonisti di un apprendimento di tipo individualizzato, sono tutte condizioni che concorrono alla ricerca del conseguimento del risultato ottimale e accompagnano l'alunno verso un utilizzo più cosciente e consapevole delle tecnologie digitali. Gli episodi di cronaca dimostrano che numerosi alunni hanno difficoltà a distinguere il bene dal male, con conseguenze comportamentali e psicologiche che, nei casi peggiori, conducono a comportamenti di violenza, bullismo e cyberbullismo. Educare alle emozioni è un

<sup>15</sup> Andrea Patassini [http://www.indire.it/wp-content/uploads/2016/06/Coding\\_eTwinning\\_Patassini.pdf](http://www.indire.it/wp-content/uploads/2016/06/Coding_eTwinning_Patassini.pdf)

<sup>16</sup> [https://www.edscuola.it/archivio/antologia/recensioni/didattica\\_costruttivista.htm](https://www.edscuola.it/archivio/antologia/recensioni/didattica_costruttivista.htm)

<sup>17</sup> Questa corrente di pensiero ritiene che la conoscenza sia una costruzione del soggetto che agisce in una situazione concreta e in collaborazione con altri soggetti con i quali negozia i significati, anche grazie alla comunicazione interpersonale e alla collaborazione sociale.

compito che oggi appare molto difficile e urgente ma è spesso trascurato. Ogni allievo rivela un diverso stato d'animo sulla base delle proprie emozioni e sensazioni e si pone in modo differente nei confronti della classe e dei docenti. Conseguentemente ogni insegnante ha il dovere di apprendere e interpretare le diverse sfumature, positive o negative, che contraddistinguono la soggettività degli alunni, per aiutarli ad assumersi la responsabilità e comprendere cosa è giusto e cosa invece è sbagliato. In genere la soluzione migliore per raggiungere quest'obiettivo è di educare alle emozioni attraverso le emozioni stesse, consigliando gli studenti a una continua analisi interiore che li formi ad assumere atteggiamenti più riflessivi e imparando, tramite le strategie didattiche, a relazionarsi correttamente gli uni con gli altri. L'insegnante, sulla base dell'esame del contesto-classe e degli obiettivi da perseguire (inclusi quelli trasversali e relazionali), deciderà se sia più appropriato far realizzare un determinato progetto didattico stabilendo la priorità più efficace tra le diverse modalità: *a coppie omogenee/eterogenee per competenze*, *a coppie omogenee/eterogenee per gusti e interessi e/o individualmente*. E' pertanto responsabilità del docente valutare le diverse situazioni e stabilire i suggerimenti e gli aiuti nel modo più vantaggioso. Tenendo sempre presente, in alternativa o parallelamente, anche la possibilità di un percorso differente ad esempio lasciando che sia un compagno di classe ad aiutare l'alunno in difficoltà. Quando si programma l'attività didattica con Scratch, con lo scopo di stimolare un'interazione creativa e relazionale tra il visivo, il digitale e il manuale, tutto questo, in genere, avviene all'interno di un "setting operativo" che presuppone delle caratteristiche dedicate e pertanto anche dei paletti vincolanti (ad esempio: un vincolo fondamentale è interagire tramite un PC). Con la metodologia collaborativa e cooperativa (*cooperative learning*), generalmente si prevedono gruppi costituiti da 3 a 5 studenti, si assegnano dei ruoli specifici e dei compiti con una finalizzazione verso un obiettivo condiviso. La gestione di un tale percorso con Scratch

diventa un po' più complicata poiché possono scaturire delle difficoltà tra i partecipanti (es. *demoralizzazione, contrasti e litigi per l'utilizzo condiviso della tastiera e/o del mouse*). Pertanto, per una corretta impostazione bisogna adattare la necessaria organizzazione dell'attività didattica in modo da prevedere una strategia basata sulla scoperta e sul tentativo di apprendere tramite prove ed errori. In queste situazioni è meglio pertanto approntare metodi e tecniche di lavoro collaborativo e di co-costruzione collettiva delle conoscenze che utilizzino proficuamente i dispositivi digitali, ma che permettano all'alunno

di entrare nel sistema da protagonista in una "logica partecipativa e relazionale". Gli studenti coinvolti nel percorso si dovranno occupare a tutto tondo del progetto, senza definire un'esplicita suddivisione di compiti oppure a ciascuno dovrà essere assegnato un ruolo preciso e circoscritto. La modalità "collaborativa" è certamente la più "aperta" in quanto non prevede una separazione formale del lavoro. Tuttavia a volte, per gli alunni, può rivelarsi difficile condurre l'attività in maniera efficace, governare l'interazione in modo da sentirsi ugualmente coinvolti nel progetto, essere in grado di gestire eventuali conflitti, evitare la confusione o la frustrazione. Perciò può essere interessante sperimentare una strategia operativa che utilizzi la "programmazione in coppia" (*pair programming*) in modo strutturato. Ad esempio uno dei due alunni programmatori, prendendo spunto dal linguaggio dei rally, svolge il ruolo di

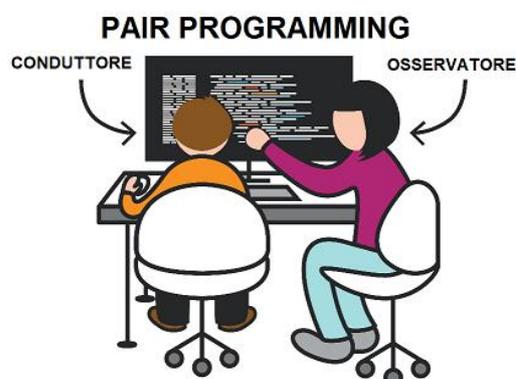


Figura 8 - Pair programming

“conduttore” (*driver*) e scrive il codice, mentre l’altro “osservatore” (*navigator*) svolge un ruolo di supervisione e di revisione del codice, ricercando eventuali errori e riproponendo, eventualmente metodi alternativi. I ruoli dovranno essere poi invertiti nel corso delle successive sessioni di lavoro (almeno in ambito didattico) secondo la strategia del *jigsaw*<sup>18</sup>. Il dinamismo e l’attività di programmazione svolta dagli alunni in coppia permetterà di sviluppare aiuti reciproci e una più intensa capacità di decentramento cognitivo<sup>19</sup>, verso due direzioni, dove in una, l’alunno dovrà comprendere “come funziona il programma” (nel nostro caso “Scratch”), intendere e mediare confrontandosi su cosa vuole ottenere il compagno di classe con una procedura definita, mentre, nell’altra direzione, permetterà di incoraggiare una maggiore conferma delle conoscenze acquisite in campo e un impegno metacognitivo per riuscire ad arrivare a comunicare e a relazionarsi con il proprio compagno più efficacemente.



Figura 9 - Compagni Tutor

In questo modo sono migliorate la capacità di decentramento e l’elaborazione metacognitiva, sia da parte del “compagno aiutante” sia da chi ha richiesto l’intromissione, che deve essere in grado di spiegare in modo chiaro i termini del problema. Se la collaborazione è utile, il “compagno tutor” può ricavare a sua volta benefici dal punto di vista della percezione, dell’autoefficacia e di sviluppo dell’autostima. Lavorare insieme per uno stesso obiettivo è pertanto un’esperienza costruttiva che rafforza la partnership all’interno della coppia, favorisce sia la risoluzione dei problemi sia l’arricchimento personale degli alunni coinvolti.

- **Stimolare l’intelligenza emotiva<sup>20</sup> e le competenze emotive<sup>21</sup>**

L’intelligenza emotiva<sup>22</sup> descrive un aspetto dell’intelligenza legato alla capacità di riconoscere, utilizzare, comprendere e gestire in modo consapevole le proprie e le altrui emozioni. L’intelligenza emotiva è dunque la capacità di riconoscere i nostri sentimenti e quelli degli altri, di riuscire a motivare noi stessi e di governare positivamente le emozioni tanto interiormente quanto nelle relazioni sociali. Progettare un’attività didattica collaborativa con “Scratch” favorisce l’apprendimento di gruppo e collaborativo. Tra gli studi trasversali ritroviamo *l’intelligenza emotiva*, poiché è l’intelligenza fondamentale che aiuta lo svolgimento di attività composite nei lavori collegiali.

<sup>18</sup> Si veda: <https://didatticapersuasiva.com/didattica/che-cose-jigsaw-e-come-si-applica>

<sup>19</sup> Dall’egocentrismo al decentramento cognitivo Secondo Piaget (1923) il bambino usa un linguaggio egocentrico, per se stesso, che scompare con la socializzazione (critica di Vygotskij 1934). Nel cap.II, aggiunto nel 1947, Piaget sostiene che il bambino è egocentrico perché non è capace di assumere mentalmente la prospettiva dell’altro, che occupa ad esempio una posizione nello spazio diversa dalla propria (*perspective taking*). In questo caso occorre superare la prospettiva egocentrica (epistemicamente) per effettuare un decentramento cognitivo.

<sup>20</sup> Intelligenza emotiva intesa come: *La capacità di controllare i sentimenti ed emozioni proprie ed altrui, distinguere tra di esse e di utilizzare queste informazioni per guidare i propri pensieri e le proprie azioni*

<sup>21</sup> La competenza emotiva intesa: come la capacità di gestire e regolare le proprie emozioni per affrontare le diverse situazioni che si propongono. Attraverso questi elementi, l’individuo è in grado di intraprendere relazioni positive con gli altri e di favorire comportamenti socializzanti.

<sup>22</sup> L’intelligenza emotiva è stata trattata la prima volta nel 1990 dai professori Peter Salovey e John D. Mayer nel loro articolo “Emotional Intelligence”. Il tema dell’intelligenza emotiva è stato successivamente trattato nel 1995 da [Daniel Goleman](#) nel libro “Emotional Intelligence” tradotto in italiano nel 1997 “Intelligenza emotiva che cos’è perché può renderci felici”.

Esercitare l'intelligenza emotiva fin dalla scuola primaria, permette di far crescere individui predisposti ad affrontare le sfide del futuro, reciprocamente connessi, rispettosi verso se stessi e gli altri. Inoltre comporta un miglioramento rilevante delle competenze socio-emotive nell'ambiente classe fornendo agli insegnanti gli strumenti essenziali per una migliore gestione e sinergia. Mettere in collegamento le attività informatiche svolte con Scratch e la competenza emotiva sembra incoerente, ma nella pratica operativa, se si analizza l'attività didattica nei gruppi di lavoro, sono

strettamente legate. La competenza emotiva rappresenta la capacità di un soggetto di manifestare le proprie emozioni in totale libertà e deriva dall'intelligenza emotiva, o meglio dalla capacità di riconoscere le emozioni stesse. La competenza emotiva è un requisito che si apprende e che indica la capacità di una persona di interagire in modo costruttivo anche con gli altri. **Se non conosciamo le nostre emozioni, è arduo provare empatia o sostenere un altro individuo.** Tutti i soggetti devono comprendere che è possibile esprimersi emotivamente tramite l'utilizzo dei diversi canali comunicativi e ampliare la propria conoscenza delle emozioni di cui hanno fatto esperienza<sup>23</sup>. Secondo diversi studi<sup>24</sup>, fin dall'età prescolare, molti alunni sono già preparati in differenti abilità fondanti la **competenza emotiva** (per esempio sono in grado di riconoscere i propri e gli altrui stati emotivi, di descrivere queste situazioni in maniera fluente e di gestire le proprie emozioni in base al fine che vogliono ottenere). La **competenza emotiva** a livello espressivo costituisce pertanto un elemento



Figura 10 - Alunni in collaborazione

importante a età diverse: se nell'infanzia rappresenta il fondamento del dialogo emotivo preverbale, con il crescere dell'età deve assicurare il buon andamento degli scambi sociali, consentendo così di affrontare le relazioni interpersonali tramite il controllo delle proprie emozioni. Nel contesto scolastico la competenza emotiva assume dunque un ruolo rilevante e deve essere potenziato sia negli alunni sia negli insegnanti e deve quindi essere introdotto anche nella formazione dei docenti.

Progettare e realizzare percorsi didattici di matrice costruttivista con Scratch, favorisce la "peer education"<sup>25</sup> e il "collaborative learning"<sup>26</sup>. Gli alunni, creando e condividendo l'esperienza con gli altri compagni di classe, possono pertanto apprendere concetti interdisciplinari e computazionali rilevanti e "insieme" iniziare a pensare con modalità creative, ragionare e operare in modo sistematico e collaborativo.



Figura 11 - Diagramma di

La **competenza emotiva** a livello espressivo costituisce pertanto un elemento importante a età diverse: se nell'infanzia rappresenta il fondamento del dialogo emotivo preverbale, con il crescere dell'età deve assicurare il buon andamento degli scambi sociali, consentendo così di affrontare le relazioni interpersonali tramite il controllo delle proprie emozioni. Nel contesto scolastico la competenza emotiva assume dunque un ruolo rilevante e deve essere potenziato sia negli alunni sia negli insegnanti e deve quindi essere introdotto anche nella formazione dei docenti.

<sup>23</sup> Comprendere le emozioni proprie e altrui vuol dire dare significato a eventi interni, o stati mentali di natura emotivo-affettiva e sviluppare una concezione della "mente emotiva" che ha la funzione di orientare le azioni dell'individuo durante gli scambi sociali (Harris P. L., 1995)

<sup>24</sup> Dunn, 1994 e S. Denham e coll., 2003

<sup>25</sup> Proposta educativa attraverso la quale, in un gruppo, alcuni soggetti (*peer educators*) vengono scelti (e formati) per svolgere il ruolo di educatore nei confronti degli altri membri dai quali, però, sono percepiti come loro simili per età, condizione lavorativa, provenienza culturale, esperienze, etc..

<sup>26</sup> Modalità di apprendimento che si basa sulla valorizzazione della collaborazione all'interno di un gruppo di allievi.

## Declinazione delle competenze personali e sociali con riferimento all'intelligenza emotiva

COMPETENZE PERSONALI	
CONSAPEVOLEZZA DI SE'	<i>Consapevolezza emotiva: riconoscimento delle proprie emozioni e dei loro effetti</i>
	<i>Autovalutazione accurata: conoscenza dei propri punti di forza e dei propri limiti</i>
	<i>Fiducia in se stessi: sicurezza nel proprio valore e nelle proprie capacità</i>
PADRONANZA DI SE'	<i>Autocontrollo: dominio delle emozioni e degli impulsi distruttivi</i>
	<i>Fidatezza: mantenimento di standard di onestà ed integrità</i>
	<i>Coscienziosità: assunzione delle proprie responsabilità</i>
	<i>Adattabilità: flessibilità nel gestire il cambiamento</i>
	<i>Innovazione: capacità di sentirsi a proprio agio e di avere un atteggiamento aperto davanti a idee e impostazioni nuove</i>
MOTIVAZIONE	<i>Spinta alla realizzazione: impulso a migliorare o a raggiungere uno standard d'eccellenza</i>
	<i>Impegno: adeguamento agli obiettivi dell'organizzazione</i>
	<i>Iniziativa: prontezza nel cogliere occasioni</i>
	<i>Ottimismo: perseguire gli obiettivi nonostante ostacoli e insuccessi</i>
COMPETENZE SOCIALI	
EMPATIA	<i>Comprensione degli altri: percezione dei sentimenti e delle prospettive altrui; interesse attivo per le preoccupazioni degli altri</i>
	<i>Assistenza: anticipazione, riconoscimento e soddisfazione delle esigenze dell'altro</i>
	<i>Promozione dello sviluppo altrui: percezione delle esigenze di sviluppo degli altri e capacità di mettere in risalto e potenziare le loro abilità</i>
	<i>Sfruttamento delle diversità: saper coltivare le opportunità offerte da persone di tipo diverso</i>
	<i>Consapevolezza politica: saper leggere e interpretare le correnti emotive e i rapporti di potere nei gruppi</i>
ABILITA' SOCIALI	<i>Influenza: impiego di tattiche di persuasione efficienti</i>
	<i>Comunicazione: invio di messaggi chiari e convincenti</i>
	<i>Leadership: capacità di ispirare e guidare i gruppi</i>
	<i>Capacità di iniziare o dirigere il cambiamento</i>
	<i>Gestione del conflitto: capacità di negoziare e risolvere disaccordi</i>
	<i>Costruzione di legami: capacità di favorire e alimentare relazioni utili</i>
	<i>Collaborazione e cooperazione: capacità di lavorare con altri verso obiettivi comuni</i>
	<i>Lavoro in team: capacità di creare una sinergia di gruppo nel perseguire obiettivi</i>

## Scratch e la comunità di apprendimento



Figura 12 - La comunità di Scratch

La classe è un atelier e una comunità di apprendimento, in cui le conoscenze sono costruite, condivise e scambiate. Ma il programma Scratch permette anche la possibilità di partecipare a una comunità molto più ampia di utenti appassionati, che non hanno in comune uno stesso spazio fisico, ma interagiscono tramite strumenti ed esperienze didattiche messe a disposizione dal proprio sito Web<sup>27</sup>. Sul sito ufficiale di Scratch ogni alunno o docente può registrare e caricare un proprio progetto da condividere con gli altri. Tutti i progetti consentono di utilizzare liberamente strumenti interattivi di musica, giochi, applicazioni didattiche, simulazioni, labirinti, storie animate, ... fino all'immaginabile. Tutto questo diventa un'ottima possibilità per dedicarsi a generi e contenuti di

"*media education*"<sup>28</sup>, sempre più fondamentali in ambito educativo man mano che i social network accrescono la loro ingerenza e il potenziale impatto sulla sfera emotiva. Il fatto che il programma Scratch metta alla prova motivazione, passione, curiosità e collaborazione, cioè che sia un ambiente focalizzato sui progetti scolastici e sul piacere di imparare a fare cose nuove e appassionanti, può essere un riferimento molto positivo per far accedere in modo corretto gli alunni al mondo dei social media. Perciò è importante invitare gli allievi, accanto all'esplorazione e all'analisi dei progetti, anche a interagire con altri utenti, commentare i loro percorsi, scambiare opinioni. Oltre alle ricadute sugli apprendimenti, questo programma ha una valenza educativa anche per lo sviluppo delle competenze trasversali, comunicative e di cittadinanza già indicate. Il docente dovrà però fare attenzione alla correttezza delle interazioni e al far rispettare le norme di netiquette<sup>29</sup>, richiamandole esplicitamente quando necessario.

## Qualche osservazione critica

L'introduzione nella didattica del Coding non ha avuto un consenso incondizionato e ha suscitato alcune critiche opportune. Partendo da una semplice considerazione, ben presente a chi fa didattica, che "*ogni studente ha capacità cognitive e stili di apprendimento differenti*", Ne deriva che, se il coding ha delle ripercussioni positive sulle modalità di apprendimento di alcuni studenti, non è detto sia altrettanto efficace per gli altri. "Dato che il *digitale e il coding sono essenzialmente un mezzo e non un fine*, qualche osservatore critico si potrebbe domandare quali possano essere i limiti per la libertà individuale, per la fantasia, per il pensiero critico, all'interno di schemi meccanicistici imposti dalla logica del linguaggio di programmazione (algoritmi imposti). Si corre pertanto il rischio che ne siano esaltati gli schematismi e si riduca il mondo dell'apprendimento solo a paradigmi logico - matematici. Il rischio insito in un credo incondizionato del Coding è di perdere l'abitudine ad applicare criteri di

<sup>27</sup> Community Italiana di Scratch con progetti di programmatori italiani <https://scratch.mit.edu/studios/3959299/>

<sup>28</sup> La Media Education è un'attività, educativa e didattica, finalizzata a sviluppare nei giovani un'informazione e comprensione critica circa la natura e le categorie dei media, le tecniche da loro impiegate per costruire messaggi e produrre senso, i generi e i linguaggi specifici- Rivoltella 2001

<sup>29</sup> Insieme di regole informali che disciplinano il buon comportamento di un utente sul web di Internet, specie nel rapportarsi agli altri utenti attraverso risorse come newsgroup, mailing list, forum, blog, reti sociali o email in genere

tempestività nella ricerca di dare soluzione ai problemi. Insegnare a pensare deve esprimere e favorire la libertà d'immaginazione, non orientare la propria mente su guide predeterminate"<sup>30</sup>. Non tutti sono d'accordo sull'utilità e validità del Coding. Stefano Penge (ricercatore e docente presso diverse Università italiane) non si esprime né a favore, né contro, ma solleva dubbi e pone domande. Altri si schierano decisamente contro. Quali sono i punti controversi?

## Il nome Coding

Il nome Coding già solleva molte critiche e perplessità. Cosa s'intende con Coding? Tradurre il termine Coding in italiano è difficile, perché questo termine coinvolge più concetti e azioni. La traduzione potrebbe essere: *codifica, cifratura, programmazione*, ma "*tradurre è un po' tradire*", diceva Voltaire.

Coding<sup>31</sup> significa letteralmente "l'azione di scrivere codice sorgente", che è un'attività della programmazione informatica. "*La programmazione, in informatica, è l'insieme delle attività e delle tecniche che una o più persone specializzate, programmatori o sviluppatori (developer), svolgono per creare un programma, ossia un software da far eseguire a un computer, scrivendo il relativo codice sorgente in un certo linguaggio di programmazione*"<sup>32</sup>.

## Programmare

Programmare, però, significa, in particolar modo, analizzare un problema, progettarne la soluzione, scrivere una serie d'istruzioni, di dati, cioè un programma, verificarne la validità e la correttezza, mentre coding fa riferimento solo alla scrittura del codice. Parlare, quindi, solo di "insegnare il *coding*" è **un approccio riduttivo**, perché si focalizza solo sulla parte finale e operativa di un processo progettuale molto più complesso. Una discrepanza, un contrasto di termini, ma anche di concetti, di propositi, tra scrivere un codice e programmare, attività che richiede conoscenze e competenze specifiche. Inoltre, in prospettiva, quanto potrebbe servire insegnare il coding in questo modo?

Nel *coding*, così inteso, non c'è quasi traccia di scrittura del codice sorgente facilitando da un lato la creazione di contenuto programmabile, ma limitando sicuramente, dall'altro, la portata delle conoscenze acquisite in campo informatico. Per colmare questo deficit di competenze tecniche nei prossimi anni assisteremo verosimilmente a un ridimensionamento della febbre da *Scratch*: il *coding* a blocchi rappresenterà solamente il primo gradino di un curriculum didattico che, almeno nelle intenzioni del MIUR, esplorerà tutti gli aspetti e le connessioni interdisciplinari dell'informatica, assomigliando sempre di più ai programmi d'informatica sperimentati nei decenni passati<sup>33</sup>. È opportuno, quindi, che, se si vuole che i nostri giovani acquisiscano competenze utili per tutto l'arco della loro vita, siano formati ad acquisire competenze da progettisti e non solo da esecutori, da creatori e non solo da utilizzatori.

## Scratch

In questo momento, Scratch, essendo molto diffuso, è diventato sinonimo di coding e, in particolar modo, come avviamento dei bambini all'informatica, attraverso ambienti di programmazione visuale. Non serve scrivere il codice. Si fa uso di un ambiente

<sup>30</sup> Per approfondimenti, si veda: <http://www.archilabo.org/content/il-coding-non-e-il-pensiero-computazione-ma-la-sua-rappresentazione/>

<sup>31</sup> Per approfondimenti, si veda il sito [https://it.wikibooks.org/wiki/Dietro\\_il\\_coding](https://it.wikibooks.org/wiki/Dietro_il_coding), di S.Penge

<sup>32</sup> Si veda <http://minimalprocedure.pragmas.org/writings/Coding/coding.html> di M.M. Ghisalberti

<sup>33</sup> <http://www.archilabo.org/content/il-coding-non-e-il-pensiero-computazione-ma-la-sua-rappresentazione/>

grafico di programmazione, che utilizza dei blocchetti, dei mattoncini tipo Lego. È sufficiente "posizionare" opportunamente questi mattoncini, "oggetti simbolici", che stanno al posto di operatori, variabili e condizioni. Sono di tipo diverso e di colore differente, ognuno con una propria funzione e il proprio compito. *"La forma dei blocchi suggerisce come questi vadano collegati e il sistema rifiuta di attaccare tra loro blocchi la cui unione non ha significato. Quindi in Scratch, la grammatica visiva della forma dei blocchi e le regole per la loro unione hanno il ruolo della sintassi in un linguaggio di tipo testuale"*.<sup>34</sup> Se si rispettano i vincoli della sintassi, i mattoncini s'incastano facilmente, in caso contrario, no. Non si tratta, quindi, proprio di "linguaggi visuali", ma d'interfacce grafiche che consentono di disporre linee di codice sorgente una sotto l'altra, nell'ordine giusto. Una specie di puzzle informatico. Scratch è un ambiente di programmazione gratuito, di tipo grafico, orientato agli oggetti chiamati: "sprites". È stato progettato, specificamente, per ragazzi dagli 8 ai 16 anni. È stato ideato con l'obiettivo di insegnare a pensare in modo creativo, a lavorare in collaborazione e a ragionare in modo sistemico. Si è pensato, inoltre, che imparare un "linguaggio di programmazione", possa essere un valido aiuto all'elaborazione di strategie, riguardanti: la modalità di risoluzione dei problemi, di creazione di progetti e di comunicazione delle proprie idee. Consente, infatti, di elaborare storie interattive, giochi, animazioni, arte e musica.

Scratch è uno strumento molto valido, intuitivo e accattivante, quindi relativamente facile per chi si avvicina alla programmazione a livello "coding" e per l'approccio iniziale alla programmazione dei computer.

Quando le cose si complicheranno e si dovrà parlare veramente di programmazione, come si metteranno le cose?

### **Un progetto USA esportato così com'è**

Scratch è stato progettato per studenti di lingua inglese, statunitensi, utilizzando un linguaggio imperativo, secondo le strutture di controllo tipiche di questa programmazione. Queste strutture di controllo della programmazione in inglese non corrispondono sempre, come significato, a quelle di lingua italiana. La differenza linguistica comporta una diversità nella definizione e, inevitabilmente, nell'apprendimento dei nostri studenti, in particolare, se di giovane età. Un *linguaggio imperativo*, inoltre, contribuisce a formare una mente imperativa; cosa che è in contrasto con i moderni modi di pensare al software e quindi non sempre adatto nella preparazione a un possibile futuro lavorativo.

Sempre nell'articolo citato di M.M. Ghisalberti, ritroviamo questa affermazione: *"L'approccio meramente visuale e compositivo, porta a distorsioni della comprensione del problema. Questo è causato anche da un linguaggio limitato, che obbliga, per risolvere anche semplici problemi, a strutture che sono, di fatto, più complesse del dovuto"*.

### **La pratica del coding sviluppa il pensiero computazionale**

Gli obiettivi di Scratch sono prevalentemente due, quelli di preparare futuri programmatori e iniziare gli alunni al pensiero computazionale (Computational Thinking). Il coding non è, quindi, semplicemente la programmazione, ma è ritenuto il modo migliore per sviluppare il Pensiero computazionale.

Questo tipo di pensiero permette di trovare la soluzione di un problema, di qualunque genere sia, indipendentemente dalle sue dimensioni (scala), si può riferire a pochi casi

<sup>34</sup> Si veda <http://tesi.cab.unipd.it/40383/1/stivalfr592920.pdf>, di F. Stival e M. Moro

o a intere popolazioni di dati. Non riguarda la tecnologia, ma, piuttosto, la creatività, la capacità di espressione e di autorealizzazione. È un invito a guardare il mondo, la complessità della realtà e i problemi con occhi nuovi.

La programmazione, infatti, dà delle istruzioni a un esecutore, che non ha un'intelligenza propria, né fantasia, e quindi, costringe a essere rigorosi e a fornirgli istruzioni, un poco per volta, passo a passo, in modo semplice ed esaustivo, senza tralasciare nessun dettaglio. I computer, eseguono velocemente un set d'istruzioni semplici ed elementari, una dietro l'altra, scandite dal registro puntatore. Sono, però, solo dei semplici esecutori, che non hanno né immaginazione, né inventiva. Siamo noi a dir loro cosa fare e a fornire le debite indicazioni. Di conseguenza, il Computational Thinking è riferito solo ai computer o agli "uomini macchina" e non a degli "uomini umani".

Il Coding serve a sviluppare il pensiero computazionale quanto altre discipline strettamente legate alla logica (*lo studio del latino, della filosofia, della letteratura o leggere una poesia tramite un paradigma costruttivista*). Se l'insegnante riesce a coinvolgere la classe, a renderla partecipe in modo vivo, anche la lettura dei classici come l'Odissea o la Divina Commedia può stimolare la curiosità e la creatività. Per certi giovani, la maga Circe o la bellissima Nausica possono essere più stimolanti di uno sprite.

### **Necessità di formare programmatori**

"Secondo uno studio dell'Unione Europa, entro il 2020 serviranno 900 mila sviluppatori in più rispetto a quanti le scuole europee ne riusciranno a formare. Secondo Code.org, addirittura negli Stati Uniti, entro il 2020, ci sarà bisogno di 1.4 milioni d'informatici esperti in programmazione, ma ci sono oggi solo 400 mila studenti iscritti in corsi di Scienze Informatiche"<sup>35</sup>!

Il proposito d'iniziare alla programmazione gli studenti, per preparare i futuri sviluppatori, di cui avremo bisogno nei prossimi dieci anni, proponendo il coding/Scratch, andrebbe considerato con più attenzione. Ci serviranno, infatti, figure professionali diverse, non solo degli sviluppatori. La programmazione, inoltre, richiede tanta creatività nell'inventare, quanta abilità nel portare a termine l'idea.

Collegata al tema del lavoro e ai futuri programmatori, è la scelta dei linguaggi di programmazione che si dimostrino adatti per essere insegnati proficuamente a scuola. Infatti, insegnare precocemente uno specifico linguaggio pone due ordini di problemi:

- il primo, legato all'effettiva difficoltà, in quanto tale linguaggio non è pensato come strumento didattico;
- il secondo, legato al fatto, che tra una decina di anni, questo linguaggio sarà quasi sicuramente superato.

Di conseguenza, almeno per la scuola primaria e sicuramente per l'infanzia, ordini coinvolti in questo progetto, affrontare questo tipo di problema sembrerebbe poco utile e fruttuoso. L'introduzione giocosa alla programmazione richiede, poi, una riflessione. Le attività ludiche sono importanti se finalizzate all'apprendimento, non semplicemente a far passare il tempo. Spesso, i giovani programmatori, abituati a giocare ai videogiochi, attratti dalla dotazione di strumenti di disegno integrati e dalla notevole libreria di elementi grafici di Scratch, sono più tentati a provare, a giocare, piuttosto che a costruire e a creare programmi. Se lasciati a se stessi, con Scratch, impiegano la maggior parte del tempo a far muovere gli sprite, spostando le figurine da una parte all'altra dello schermo, in percorsi preordinati, rinunciando a progettare e

<sup>35</sup> Si veda [https://www.wecanjob.it/archivio21\\_coding-programming-future-works\\_0\\_46.html](https://www.wecanjob.it/archivio21_coding-programming-future-works_0_46.html)

a realizzare qualcosa. Pertanto è fondamentale l'azione dell'insegnante, che con una solida programmazione didattica e, tanta pazienza, deve orientare, guidare, sostenere i giovani programmatori, affinché concretizzino i loro sforzi in un prodotto, che non sia la replica di qualche cosa già visto o già sentito.

Scratch fa parte di quegli ambienti preconfezionati e omnicomprensivi che richiedono, da parte degli insegnanti tutte le attenzioni per stimolare nell'alunno la creatività e la volontà di esprimersi.

Scratch, inoltre, stimola il *copia ed incolla*, secondo la logica del *programming by example*. La programmazione per esempi, l'assemblaggio del codice scritto da altri nei vari e diffusi *tutorial*, è un'attività più che utile, se condotta in modo appropriato. Imparare dagli altri e con gli altri è una modalità valida e connaturata alla natura umana, ma poi, deve esserci il salto di qualità. Il vero esercizio di creatività: pensare e realizzare qualcosa di diverso, di nuovo, che corrisponda agli interessi, alla curiosità dello studente programmatore e che stimoli a riflettere e a esprimere la sua personalità, la sua individualità.

### Coding nell'infanzia

Il primo livello di apprendimento è quello che **nasce dall'interazione dell'individuo con l'ambiente che lo circonda**. In particolare, secondo Piaget, il modo migliore per ottenere questo tipo di apprendimento è attraverso la manipolazione: mediante la propria azione sugli oggetti, il bambino acquisisce nuove competenze e nuove informazioni. Nei bambini, uno degli *stili di apprendimento prevalenti* è il *sistema tattile*. I bambini imparano attraverso esperienze tattili<sup>36</sup>, specialmente quando sono molto piccoli (infanzia). Questo è il motivo per cui i bambini e le bambine della scuola dell'infanzia sono stimolati a sviluppare questa competenza attraverso l'uso della pasta di sale, la pittura, il disegno, il movimento, per poi poter finalmente trasferire l'apprendimento in quella parte della neocorteccia cerebrale, deputata a migliorare le abilità linguistiche, il pensiero critico, la risoluzione dei problemi, la lettura e la scrittura.

I bambini, inoltre, apprezzano anche le seguenti azioni:

- *Attività pratiche che coinvolgono progetti artistici, naturali, recitazione,*
- *Giocherellare spesso con la matita, muovere i piedi, tenere un oggetto mentre studiano,*
- *Usare le dita per tracciare lettere, numeri e forme per l'ortografia e la lettura,*
- *Avere bisogno di muoversi e fare pause per allontanarsi dal banco.*

Ricordiamo che un uso eccessivo dei dispositivi digitali nella prima infanzia può interferire con il normale sviluppo cognitivo, motorio e sociale<sup>37</sup>. L'utilizzo di oggetti, tecnologicamente sofisticati, ma di aspetto semplice ed invitante, tipo cubetto, Ape Bee Bot, Makey Makey<sup>38</sup> ecc, possono provocare, in certi elementi giovani, una *distorsione percettiva*. La percezione<sup>39</sup> che si ha di un oggetto, infatti, è il processo attraverso cui un individuo seleziona, organizza e interpreta gli stimoli ricevuti e le informazioni acquisite, in modo da ottenere una visione del mondo che sia per lui credibile e dotata di senso. Persone diverse possono avere dello stesso oggetto

<sup>36</sup> <http://www.familing.it/limportanza-manipolazione-per-sviluppo-dellapprendimento-non/>

<sup>37</sup> <http://www.healthdesk.it/prevenzione/bambini-troppe-ore-davanti-uno-schermo-compromettono-sviluppo-cognitivo>

<sup>38</sup> *Makey Makey è un kit in grado di trasformare qualunque cosa in una tastiera del computer. Basta collegare le pinze a coccodrillo contenute nella sua scatola a qualsiasi oggetto per trasformarlo in un controller touch. Si può giocare ai videogame con dei tasti di plastilina o suonare dei vegetali come se fossero i tasti di un pianoforte....". Per approfondimenti, si veda: <http://www.maestramarta.it/makey-makey-classe-5a/>*

<sup>39</sup> Si veda: [https://amslaurea.unibo.it/2129/1/nuozzi\\_silvio\\_tesi.pdf](https://amslaurea.unibo.it/2129/1/nuozzi_silvio_tesi.pdf); 4. Fattori psicologici (motivazione, percezione, apprendimento, opinioni e atteggiamento).

percezioni diverse, rielaborando le informazioni secondo un proprio punto di vista (distorsione selettiva). Gli stimoli ricevuti non sempre, poi, sono interpretati nel modo desiderato, infatti, si ha la tendenza a ritenere un'informazione che concorda con quanto si è rielaborato, e che ci è diventato familiare, dimenticando molte delle altre cose che si sono imparate (ritenzione selettiva). Così può accadere che il bambino "programmatore unplugged", se non debitamente seguito e avvertito, conferisca a degli oggetti comuni, caratteristiche e proprietà derivanti dai gadget elettronici: come banane che cantano, oggetti che parlano, ecc.

### Riflessione finale

Infine, è così importante imparare a programmare? La società attuale, orientata all'uso sempre più diffuso delle tecnologie digitali, ha più bisogno, per sopravvivere di consumatori, operatori, piuttosto che di programmatori. I nostri studenti, dall'infanzia alla secondaria di secondo grado, diventeranno tutti programmatori? Ne hanno tutti le capacità, la voglia l'interesse, la possibilità economica? Secondo le ultime stime dell'OCSE, in Italia le competenze degli studenti del Sud sono inferiori a quelle del Nord, in gran parte, per motivi economici. E si parla solamente di comprensione del testo, di matematica e di scienze. Immaginiamo la programmazione con dispositivi sicuramente non sempre gratuiti e attrezzature per scuole spesso fatiscenti. Utilizzare il software Scratch e simili, muovere robot programmabili, api, cubetti è sicuramente divertente, intrigante. Sono attività accattivanti e svaganti, preferibili a lezioni tradizionali.

Il pensiero computazionale<sup>40</sup> insegna ad analizzare e a risolvere i problemi, fornendo le istruzioni necessarie in modo tale che un umano o una macchina possa effettivamente eseguire. La vita reale, però, non è una simulazione, un'astrazione, le variabili in gioco sono, pressoché, infinite e la relazione tra causa ed effetto è solo probabilistica. Il pensiero computazionale, pertanto, non deve essere demonizzato, anzi va incoraggiato e insegnato. Allo stesso tempo, però, non deve essere idealizzato, né considerato come la soluzione universale per tutte le discipline e valido per ogni alunno. Bisogna avvalersene con cognizione di causa, affiancandolo ad altre metodologie e strategie e valorizzando proposte didattiche, che permettano una formazione generale dello studente, arricchiscano le sue potenzialità espressive, gli permettano di esprimere i suoi sentimenti, la sua fantasia e la sua libertà di pensiero<sup>41</sup>.

### Conclusioni

Quanto si è osservato finora ci porta a considerare che il Coding e Scratch abbiano un valore non solo come ambiente di programmazione, su cui si può sostenere anche qualche critica, ma soprattutto come ambiente didattico propedeutico poiché aiuta gli alunni a



Figura 13 – Alunni in collaborazione

<sup>40</sup> Si veda: [https://it.wikipedia.org/wiki/Pensiero\\_computazionale](https://it.wikipedia.org/wiki/Pensiero_computazionale)

"Il pensiero computazionale è l'insieme dei processi mentali coinvolti nella formulazione di un problema e della sua soluzione, quindi di individuare un procedimento costruttivo, fatto di passi semplici non ambigui, che ci porta alla soluzione di un problema complesso<sup>40</sup>. Il pensiero computazionale è un processo iterativo basato su tre fasi:

1. Formulazione del problema (astrazione);
2. Espressione della soluzione (automazione);
3. Esecuzione della soluzione e valutazione della stessa (analisi)"

<sup>41</sup> Si veda: <http://www.archilabo.org/content/il-coding-non-e-il-pensiero-computazione-ma-la-sua-rappresentazione/>

progettare, creare percorsi multidisciplinari e a relazionarsi con il prossimo. Lo stesso ambiente però diventa un po' più disagiata quando i programmi diventano troppo lunghi e quando si rende indispensabile avere strutture dati oltre le variabili e le liste, ecc. Diventa insufficiente e inadeguato quando gli alunni hanno superato la fase iniziale propedeutica della programmazione di avvio, della realizzazione e del proporre l'utilizzo. Gli studenti che desiderano continuare ad approfondire: algoritmi, programmazione e conoscenze degli ambienti digitali in genere, dovranno necessariamente passare ad altri strumenti e software un po' più avanzati. Per concepire gli usi creativi dell'IT<sup>42</sup> è diventato ormai indispensabile possedere i concetti basilari della computer science, inoltre, la conoscenza dei concetti base, consentirà agli studenti di essere creativi anche in altre discipline e ambienti. Il coding per la sua particolare azione rivelativa, è anche uno strumento di sostegno molto efficace per chi s'interessa di metodologie di studio per gli alunni con disturbi dell'apprendimento e permette di verificare immediatamente i rapporti razionali di causa ed effetto e costatare la legittimità di una categorizzazione o di un ragionamento logico. La missione della scuola è di fornire a ciascun alunno le giuste competenze per "saper" affrontare il proprio futuro.

---

**Gli autori:** molti materiali e indicazioni presenti in questo articolo sono stati ripresi da letture e stimoli ritrovati in rete e non sempre è stato possibile citare tutti gli autori. Desideriamo comunque ringraziare tutti quelli che in qualche modo ci hanno dato idee e proposte che hanno permesso la realizzazione di quest'articolo.

## Indicazioni utili in rete

- Corso di formazione on line gratuito della UniUrb <https://mooc.uniurb.it/wp/welcomemooc/>
- Corso di formazione dedicato OPPI <http://oppi.it/corsi/corso-coding-pensiero-computazionale/>
- Il MIUR, in collaborazione con il CINI – Consorzio Interuniversitario Nazionale per l'Informatica, ha avviato questa iniziativa con l'obiettivo di formare gli studenti ai concetti di base <https://programmmailfuturo.it/>
- Storia del LOGO <http://www.comune.jesi.an.it/jesicentro/TDC/DISPENSE/LOGO/storia.htm>
- Software Scratch versione on line : <https://scratch.mit.edu/projects/318997608/editor>
- Google corso base programmazione con Scratch. <https://csfirst.withgoogle.com/signin-front>
- CNR – Pensiero computazionale - Guida per insegnanti [http://pensierocomputazionale.itd.cnr.it/pluginfile.php/957/mod\\_page/content/7/Guida%20al%20Pensiero%20Computazionale.pdf](http://pensierocomputazionale.itd.cnr.it/pluginfile.php/957/mod_page/content/7/Guida%20al%20Pensiero%20Computazionale.pdf)
- Progetti Scratch [https://aretepiattaforma.it/progetti\\_scratch/progetti.php](https://aretepiattaforma.it/progetti_scratch/progetti.php)
- RAI-scuola Cos'è il pensiero computazionale di V. Sellitto <http://www.raiscuola.rai.it/lezione/cos%20C3%A9-il-pensiero-computazionale/10184/default.aspx>
- Tesi di laurea della prof.ssa Elena Pacetti, correlatore prof. Michael Lodi, anno 2016 [https://michaellodi.files.wordpress.com/2014/11/tesi-di-laurea\\_olivari.pdf](https://michaellodi.files.wordpress.com/2014/11/tesi-di-laurea_olivari.pdf)
- Goleman D., Intelligenza emotiva, Bur 1996
- Risorse per aiutare la classe e i docenti: <http://scratched.media.mit.edu/>  
<http://scratched.gse.harvard.edu/>
- Raccolta di articoli [http://www.rivistabricks.it/wp-content/uploads/2019/03/BRICKS\\_1\\_2019.pdf](http://www.rivistabricks.it/wp-content/uploads/2019/03/BRICKS_1_2019.pdf)
- Critica sul Coding: <https://www.techeconomy2030.it/2016/03/16/contro-il-coding/>
- Carletti – Varani e contributi Equipe IAD "Didattica costruttivista" Erikson 2005 [https://www.edscuola.it/archivio/antologia/recensioni/didattica\\_costruttivista.htm](https://www.edscuola.it/archivio/antologia/recensioni/didattica_costruttivista.htm)

---

<sup>42</sup> L'acronimo **IT** (*information technology*), indica l'utilizzo di elaboratori e attrezzature di telecomunicazione per memorizzare, recuperare, trasmettere e manipolare dati, spesso nel contesto di un'attività commerciale o di un'altra attività economica.

— **Gli autori: Equipe Formazione Digitale** —



**Mario Gabbari:** Docente-formatore presso OPPI-Milano. Ha collaborato per lungo tempo con l'Ufficio Scolastico della Lombardia partecipando e/o organizzando attività seminari e di formazione e svolgendo numerosi incarichi per conto del MIUR e Indire. Ha scritto e pubblicato diversi articoli sulle tecnologie informatiche e insieme ad altri, un libro guida sulla LIM (Lavagna Interattiva Multimediale), come ambiente di apprendimento costruttivista e dedicato a tutti gli operatori della comunicazione.



LIM.

**Antonio Gaetano:** Docente in scuola statale, formatore per conto del MIUR, del nucleo Regionale Lombardia e dell'ex Indire. Collabora dal 1999 come esperto con l'USR Lombardia e con il CSA di Milano per l'introduzione delle ICT e delle LIM nella scuola. Autore, con altri, di un libro guida sull'utilizzo della



**Roberto Gagliardi:** Docente-formatore, coordinatore della Equipe dell'OPPI di Milano, per Tecnologie didattiche e di Comunicazione. Si è occupato della formazione LIM per conto di INDIRE/MIUR e degli Aspetti strumentali e metodologico - didattici. Ha collaborato con l'USR Lombardia in corsi e attività seminari come tutor, relatore e progettista. Autore, con altri, di diverse pubblicazioni: libri e articoli sugli aspetti metodologici didattici e sull'utilizzo della LIM come ambiente di apprendimento costruttivista.



**Daniela Sacchi:** Docente-formatore presso OPPI-Milano, ha collaborato per lungo tempo con l'Ufficio Scolastico della Lombardia partecipando ad attività seminari e di formazione e ha svolto diversi incarichi per il MIUR. Ha condotto per alcuni anni il laboratorio ADA (Progettazione di percorsi didattici multimediali per i diversi tipi di disabilità) all'Università Bicocca di Milano. Ha scritto articoli e libri nell'ambito delle ICT, personalmente o insieme ad altri.