Equipe Formazione Digitale:
M. Gabbari
R. Gagliardi
A. Gaetano
D. Sacchi*

Integrare "Coding e Pensiero computazionale" nella didattica

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 tecnologie digitali e la programmazione informatica – *(coding) e il pensiero computazionale*[1] – funzionano come un ponte di collegamento tra l'uomo e la macchina assicurandone la reciproca interazione.

Quando il presidente Obama ha promosso una campagna esortativa[2] per comunicare alle nuove generazioni l'importanza di prendere consapevolezza che

* L'Equipe Formazione Digitale di OPPI da alcuni anni è impegnata nelle scuole e on line per svolgere corsi di formazione sul "Coding e Pensiero computazionale". Questi impegni hanno richiesto un considerevole approfondimento, sugli strumenti e le applicazioni che riguardano la didattica e le metodologie.

[1] Processo mentale per la risoluzione di problemi costituito dalla combinazione di metodi caratteristici e di strumenti intellettuali. Termine introdotto dalla scienziata informatica Jeannette Wing nel 2006, in sintesi definibile come: "*l'applicazione dei principi del pensiero informatico per la risoluzione di attività o problemi logici più o meno complessi. Una capacità che può risultare fondamentale nella vita di tutti i giorni e non soltanto davanti allo schermo di un computer*".

[2] Il presidente americano Barack Obama, con una dichiarazione alla stampa (gennaio 2016), aveva esortato i giovani americani a non spendere le proprie risorse per comprare videogiochi, ma piuttosto a concentrare i propri sforzi per crearli. Il Presidente ha poi lanciato un piano per diffondere il coding in tutte le scuole. E una legge vuole equipararlo a una lingua straniera, per insegnarlo in classe.

il futuro ha bisogno di loro, della loro capacità di programmare e di acquisire competenze di pensiero computazionale, si è scatenato un forte interesse per la programmazione. In tutto il mondo, non solo informatico e scolastico, si è diffusa l'importanza di far conoscere e studiare ai giovani allievi i linguaggi della programmazione, includendo anche i più piccoli, con l'impiego, ovviamente, di linguaggi adatti e semplificati. Tali prerogative sono indispensabili per essere "*creatori*" e attivamente partecipi della propria crescita e non semplici "*consumatori*" di un'economia digitale: assumere insomma il ruolo di cittadini attivi in una società sempre più tecnologica. Nel prossimo futuro, ormai alle nostre porte, la capacità di programmare farà la differenza tra chi dà ordini alle macchine e chi esegue gli ordini delle macchine. Pertanto imparare a programmare consentirà ai nostri alunni di uscire dalla logica di essere solo dei meri utilizzatori potendo diventare loro stessi dei potenziali sviluppatori. La capacità di programmare permette inoltre di rendere concreti i concetti del pensiero computazionale e contribuisce a farlo diventare a sua volta uno strumento di apprendimento. Il coding riconduce alla teoria dell'apprendimento del costruzionismo[3] poiché promuove la costruzione di modelli mentali, lo sviluppo delle capacità critiche e la realizzazione di percorsi concreti e operativi. È in grado di correggere le cattive impostazioni della scuola tradizionale dove lo studente è oggetto di sistemi educativi che lo rendono spesso un soggetto poco partecipativo.

Il coding per tutti

In questi ultimi anni il coding si è pienamente affermato per la sua valenza didattica, ormai universalmente riconosciuto come dinamica di apprendimento nel trasmettere agli alunni competenze di problem solving, creatività digitale e lavoro di gruppo. Il Coding ha un autorevole antenato: il LOGO, il primo linguaggio di programmazione orientato all'educazione dei bambini. Alla fine degli anni '60 il matematico Seymour Papert e Wallace Feurzeig[4] si incontrarono e, insieme alla scienziata ed esperta di Lisp, Cynthia Solomon, crearono la prima versione di LOGO[5] 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

[3] Secondo S. Papert, il costruzionismo è una variante del costruttivismo che condivide con quest'ultimo l'idea di un uomo costruttore di strutture di conoscenze, ma specifica anche che la costruzione di conoscenze è molto più significativa in un contesto dove il soggetto che apprende è impegnato nella costruzione di un qualcosa di concreto e condivisibile. Il costruzionismo introduce quindi il concetto di "*artefatti cognitivi*" o degli oggetti o dispositivi che facilitano l'apprendimento e di cui l'uomo necessita esattamente come un costruttore ha bisogno dei materiali da costruzione.

[4] 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.

[5] 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> (ultimo accesso maggio 2020).

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[6]:

> "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".

Il Coding offre la possibilità di realizzare una straordinaria esperienza sviluppando il ragionamento, la fantasia e le personali abilità per realizzare qualcosa (Fig. 1). 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*[7], 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. Alternandosi con 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 pratica 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 migliorare i propri elaborati. 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.

[6] Docente ordinario, coordinatore della Scuola di Scienze e Tecnologie dell'Informazione dell'Università degli Studi di Urbino Carlo Bo – Dipartimento di Scienze pure e applicate. Autore di corsi e di libri sul Coding.

[7] 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.

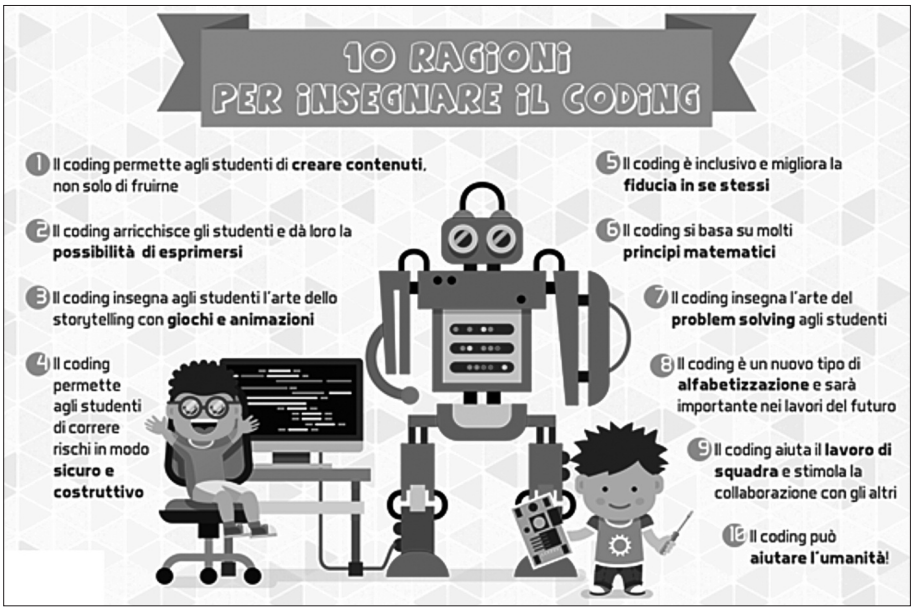

Fig. 1 – Infografica riepilogativa delle ragioni del Coding (DEA Scuola)

Coding e pensiero computazionale in aula

Il Coding è anche un modo divertente e appassionante per introdurre 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 (anche negli adulti), il timore di sbagliare o di non riuscire pone il freno all'accettazione di sé e alla disponibilità al dialogo (*sapersi mettere in gioco*): ciò può diventare un blocco mentale che accompagna l’alunno durante tutto il proprio percorso scolastico. L’errore, invece, 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 sua insita valenza positiva per andare oltre (stimolo al miglioramento).

L’utilizzo di questi software di programmazione (es. Scratch[8]) da parte degli

[8] 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.

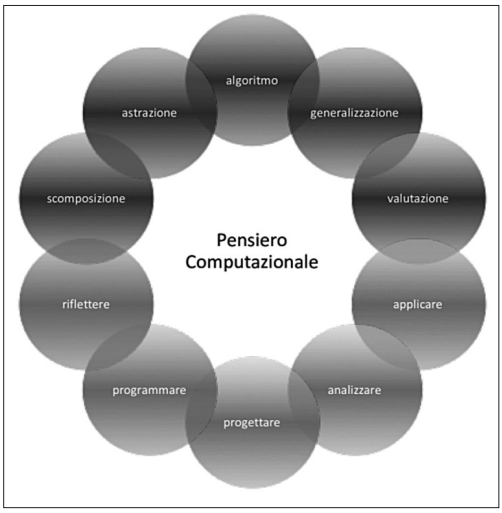

Fig. 2 – Computational thinking - A guide for teachers (CNR)

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*[9] 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 (Fig. 2). 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 sicuramente utile a sviluppare proposte didattiche innovative e inclusive, e a valutare le competenze. L'attuazione di una formazione docente attenta e indirizzata alla dimensione umanistico-espressivo-creativa è da considerarsi indispensabile alla pratica del pensiero computazionale, anche se 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"*[10]. 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. È 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, raf-

[9] L'Equipe svolge la sua attività di formazione on line e in presenza presso l'OPPI di Milano.
[10] MORIN E., *La testa ben fatta. Riforma dell'insegnamento e riforma del pensiero*, op. cit., pp. 89-90 e ss.

figurata 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*[11]. 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[12] 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 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 a disposizione, poiché consentono nuove modalità di approccio alla didattica che possono

[11] 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 Wing J.M., direttrice del Dipartimento di Informatica della Carnegie Mellon University, del marzo 2006 alla pagina <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf> (ultimo accesso maggio 2020).

[12] Vedi *Computational thinking - A guide for teachers* © Copyright 2015 Computing At School - © Copyright 2016, Consiglio Nazionale delle Ricerche in <http://community.computingatschool.org.uk/ resources/ 2324> (ultimo accesso maggio 2020).

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[13] 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[14] hanno predisposto nei propri corsi di formazione l'impiego del software Scratch, 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.

Considerando il rapporto fra apprendimenti scolastici ed extrascolastici, è frequente che, una volta appreso l'utilizzo di Scratch, alcuni alunni 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 (Fig. 3). 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

[13] Software Scratch versione on line: <https://scratch.mit.edu/>, (ultimo accesso maggio 2020).

[14] 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/>, (ultimo accesso maggio 2020).

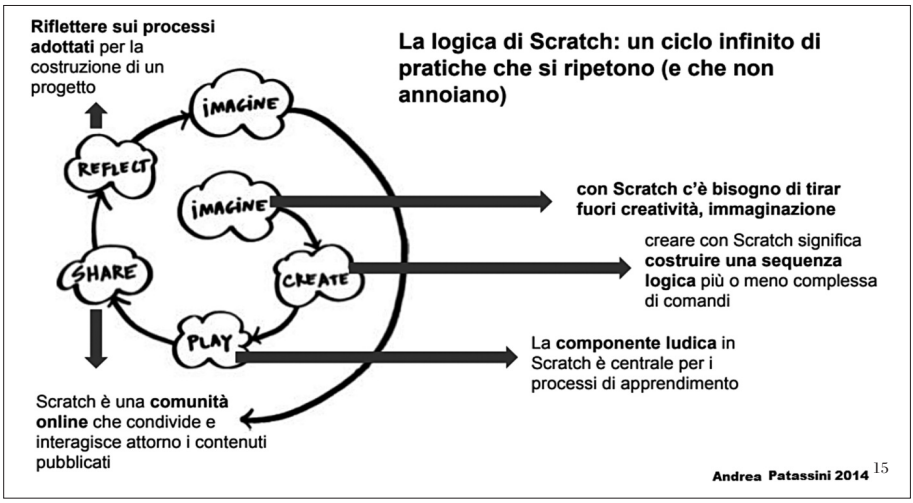

Fig. 3 – Schema che illustra la logica di Scratch

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[16]

Nel metodo dell’apprendimento creativo, di nascita costruzionista[17], 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 compito che oggi appare molto difficile e urgente ma è spesso trascurato. Ogni allievo rivela un

[15] PATASSINI A., <http://www.indire.it/wp-content/uploads/2016/06/Coding_eTwinning_Patassini.pdf>, (ultimo accesso maggio 2020).

[16] Didattica costruttivista, <https://www.edscuola.it/archivio/antologia/recensioni/didattica_costruttivista.htm> (ultimo accesso aprile 2020).

[17] 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.

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, esortando gli studenti a una continua analisi interiore che li formi ad assumere atteggiamenti più riflessivi e imparare, 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. È 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 la finalità di 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 "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[18]. 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[19], verso due direzioni, dove in una l'alunno dovrà comprendere prima come funziona il programma (nel nostro caso "Scratch"), in seguito intendere e mediare, confrontandosi su cosa vuole ottenere il compagno di classe con una procedura definita; mentre, nell'altra direzione, si promuoverà sia una conferma delle conoscenze acquisite in campo che uno sforzo metacognitivo, per riuscire ad arrivare a comunicare e a relazionarsi con il proprio compagno più efficacemente.

In questo modo si migliorano 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[20] e le competenze emotive[21]

L'intelligenza emotiva[22] descrive un aspetto dell'intelligenza legato alla capacità di riconoscere, utilizzare, comprendere e gestire in modo consapevole

[18] Si veda: <https://didatticapersuasiva.com/didattica/che-cose-jigsaw-e-come-si-applica>, (ultimo accesso maggio 2020).

[19] 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.

[20] 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*".

[21] 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".

[22] 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*.

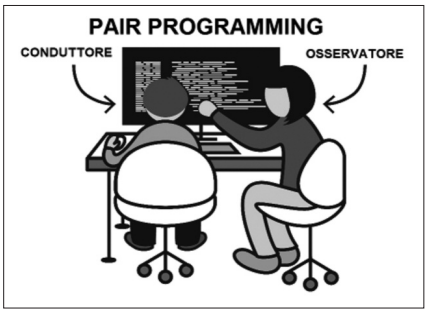

Fig. 4 – Pair programming

Fig. 5 – Diagramma di concettualizzazione

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 (Fig. 4). Tra gli studi trasversali ritroviamo *l'intelligenza emotiva*, poiché è l'intelligenza fondamentale che aiuta lo svolgimento di attività composite nei lavori collegiali (Fig. 5). 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[23]. Secondo diversi studi[24], fin dall'età prescola-

[23] 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, in HARRIS P. L., *The rise of introspection*, 1995, <https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5834.1995.tb00196.x>, (ultimo accesso maggio 2020).

[24] DUNN J., *Changing minds and changing relationships. Children's early understanding of mind: Origins and development*, 1994, pp. 297-310 e DENHAMS S. A., BLAIR K. A., DE MULDER E., LEVITAS J., SAWYER K.,

re, 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 importante in età diverse: se nell'infanzia rappresenta il fondamento del dialogo emotivo preverbale, con l'avanzare 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*[25]" e il "*collaborative learning*[26]". 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.

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

COMPETENZE PERSONALI	
CONSAPEVOLEZZA DI SÈ	***Consapevolezza emotiva: riconoscimento delle proprie emozioni e dei loro effetti***
	Autovalutazione accurata: conoscenza dei propri punti di forza e dei propri limiti
	Fiducia in se stessi: sicurezza nel proprio valore e nelle proprie capacità
PADRONANZA DI SÈ	***Autocontrollo: dominio delle emozioni e degli impulsi distruttivi***
	Fidatezza: mantenimento di standard di onestà ed integrità
	Coscienziosità: assunzione delle proprie responsabilità
	Adattabilità: flessibilità nel gestire il cambiamento
	Innovazione: capacità di sentirsi a proprio agio e di avere un atteggiamento aperto davanti a idee e impostazioni nuove
MOTIVAZIONE	***Spinta alla realizzazione: impulso a migliorare o a raggiungere uno standard d'eccellenza***
	Impegno: adeguamento agli obiettivi dell'organizzazione
	Iniziativa: prontezza nel cogliere occasioni
	Ottimismo: perseguire gli obiettivi nonostante ostacoli e insuccessi

AUERBACHMAJOR S. e QUEENAN P., *Preschool emotional competence: Pathway to social competence? Child Development*, 74 (1), 2003, pp. 238-256.

[25] 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…

[26] Modalità di apprendimento che si basa sulla valorizzazione della collaborazione all'interno di un gruppo di allievi.

COMPETENZE SOCIALI	
EMPATIA	***Comprensione degli altri: percezione dei sentimenti e delle prospettive altrui; interesse attivo per le preoccupazioni degli altri***
	Assistenza: anticipazione, riconoscimento e soddisfazione delle esigenze dell'altro
	Promozione dello sviluppo altrui: percezione delle esigenze di sviluppo degli altri e capacità di mettere in risalto e potenziare le loro abilità
	Sfruttamento delle diversità: saper coltivare le opportunità offerte da persone di tipo diverso
	Consapevolezza politica: saper leggere e interpretare le correnti emotive e i rapporti di potere nei gruppi
ABILITÀ SOCIALI	***Influenza: impiego di tattiche di persuasione efficienti***
	Comunicazione: invio di messaggi chiari e convincenti
	Leadership: capacità di ispirare e guidare i gruppi
	Capacità di iniziare o dirigere il cambiamento
	Gestione del conflitto: capacità di negoziare e risolvere disaccordi
	Costruzione di legami: capacità di favorire e alimentare relazioni utili
	Collaborazione e cooperazione: capacità di lavorare con altri verso obiettivi comuni
	Lavoro in team: capacità di creare una sinergia di gruppo nel perseguire obiettivi

Scratch e la comunità di apprendimento

La classe è un atelier e una comunità di apprendimento, in cui le conoscenze sono costruite, condivise e scambiate. Il programma Scratch permette anche la possibilità di partecipare a una comunità molto più ampia di utenti appassionati (Fig. 6), che non hanno in comune uno stesso spazio fisico, ma interagiscono tramite strumenti ed esperienze didattiche messe a disposizione dal proprio sito Web[27]. 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'inimmaginabile. Tutto questo diventa un'ottima possibilità per dedicarsi a generi e contenuti di "*media education*[28]", sempre più fondamentali in ambito

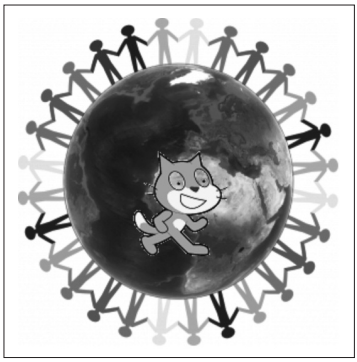

Fig. 6 – La comunità di Scratch

[27] Community Italiana di Scratch con progetti di programmatori italiani <https://scratch.mit.edu/studios/3959299/>, (ultimo accesso maggio 2020).

[28] 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 P., *Modelli, esperienze, profilo disciplinare*, Carocci, Roma, 2001.

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[29], richiamandole esplicitamente quando necessario.

Qualche osservazione critica

Il consenso all'approccio didattico del Coding non è sempre incondizionato e porta spesso con sé anche alcune critiche opportune. Partendo da un'analisi semplice, che chi fa didattica specialistica e individualizzata non può dimenticare che ogni studente ha capacità cognitive e stili cognitivi differenti. Questo implica che, se il coding ha delle ripercussioni positive sulle metodologie di apprendimento per alcuni studenti, potrebbe non essere altrettanto efficace per altri.

Ricordando che il digitale e il coding sono essenzialmente un mezzo e non un fine, qualche osservatore critico si domanda quali possono essere i limiti per la libertà individuale, per la fantasia, per il pensiero critico, all'interno degli schemi certamente meccanicistici imposti dalla logica del linguaggio di programmazione (algoritmi imposti). Si corre pertanto il rischio che 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 tempestività nella ricerca di una soluzione ai problemi. Insegnare a pensare deve esprimere e favorire la libertà d'immaginazione, non orientare o guidare la propria mente su direttrici predeterminate.

Il Coding aiuta a sviluppare il pensiero computazionale quanto altre discipline strettamente legate alla logica. Il pensiero computazionale, pertanto, non deve essere demonizzato, anzi va incoraggiato e insegnato ma, allo stesso tempo, non deve essere idealizzato, né deve essere considerato come la soluzione universale per tutte le discipline e valido per ogni alunno. Bisogna avvalersene con cognizione di causa, affiancandolo con altre metodologie e strategie e valorizzando proposte didattiche che permettano una formazione generale

[29] 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

dello studente, delle sue potenzialità espressive, dei suoi sentimenti, della sua fantasia e della sua libertà di pensiero.

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 progettare, creare percorsi multidisciplinari e a relazionarsi con il prossimo.

Lo stesso ambiente però diventa un po' più disagevole 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 proporne 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[30] 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[31].

[30] 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.

[31] 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.