



Il Coding nella scuola Italiana. L'esperienza pugliese e il confronto con le ricerche internazionali / Coding in italian school. The apulian experience and a comparison with the international researches

Giuseppina Rita Mangione

Indire – Istituto Nazionale di Documentazione, Innovazione e Ricerca Educativa
g.mangione@indire.it

Maria Guida

Indire – Istituto Nazionale di Documentazione, Innovazione e Ricerca Educativa
m.guida@indire.it

ABSTRACT

“Docenti InFormAzione” is the teacher training experience within EDOC@Work3.0, a project aimed at apulian schools in which the Coding methodology was tested. The interest in this teaching method, with particular reference to the primary school, is consistent with the changes in European educational programs that are realizing the value of introducing topics such as programming, computer science and computational thinking in the curricula. These changes stimulate scientific debate and lead us to make a critical reading of the results emerged from this experimentation in apulia classroom, with regard to the impact on the curriculum, but also on learning outcomes and teaching practices.

This reasoning is conducted based on stimuli and research lines promoted by scientific literature. These findings represent the query criteria for qualitative data, acquired by the experience, in order to confirm certain conditions whether or not and to understand their variation with respect to the disciplines involved in the trial, to technology integration and to class dynamics triggered in the italian context.

“Docenti InFormAzione”¹ è il percorso formativo all’interno del progetto EDOC@Work3.0² rivolto alle scuole pugliesi in cui è stata sperimentata la metodologia del Coding. L’interesse per questo metodo didattico, con particolare riferimento alla scuola primaria, è in linea con i cambiamenti dei programmi scolastici europei che stanno comprendendo il valore di introdurre temi come la programmazione, la computer science e il pensiero computazionale nei curricula. Tali cambiamenti stimolano il dibattito scientifico e ci guidano in una lettura critica dei risultati emersi dall’esperienza nelle aule pugliesi rispetto non solo ad un primo impatto sul curricolo, ma anche sugli apprendimenti e sulle pratiche didattiche messe a sistema dai docenti.

Questo ragionamento viene condotto sulla base degli stimoli e delle linee di pensiero promosse dalla letteratura scientifica sul tema. Tali studi rappresentano le piste di interrogazione dei dati qualitativi acquisiti dalla esperienza nelle classi pugliesi al fine di confermare o meno alcune condizioni e comprenderne la declinazione rispetto alle discipline coinvolte nella sperimentazione, all’integrazione tecnologica e alle dinamiche di classe attivate nel contesto italiano.

KEYWORDS

Coding, Computational Thinking, Design, Organizational and didactical evidence, Apulian classroom.

Coding, Pensiero Computazionale, Progettazione, Evidenze organizzative e didattiche, Classi pugliesi.

1 Giuseppina Rita Mangione è autrice dei paragrafi: “Il Coding a scuola. Le evidenze della ricerca empirica”, “Analisi delle pratiche pugliesi e confronto con le evidenze”, “Le Osservazioni non partecipate nei cantieri sperimentali”. È co-autrice del paragrafo “Analisi delle progettazioni e confronto con le evidenze”.

Maria Guida è autrice dei paragrafi “Il progetto EDOC@WORK 3.0 l’innovazione didattica”, “Conclusioni”. È coautrice del paragrafo “Analisi delle progettazioni e confronto con le evidenze”.

2 Programma Operativo Nazionale Ricerca e Competitività 2007-2013 (SMART EDUCATION).

1. Il progetto EDOC@WORK 3.0 e l'innovazione didattica

Nel quadro dell'innovazione didattica promossa dal progetto EDOC@WORK 3.0, INDIRE ha promosso un intervento di formazione-azione nella scuola primaria e secondaria di I e II grado nella regione Puglia. IL percorso "Docenti InFormAzione" volto a migliorare la lezione d'aula ha visto la sperimentazione di differenti pratiche innovative e il coinvolgimento nello specifico per "Coding" di 151 insegnanti (fig. 1a) afferenti all'area scientifica, umanistica e al sostegno (fig. 1b.).

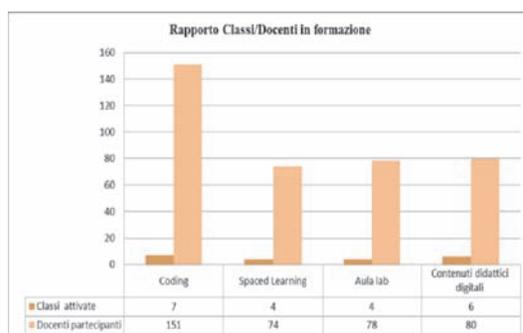


Fig. 1a. Docenti in formazione

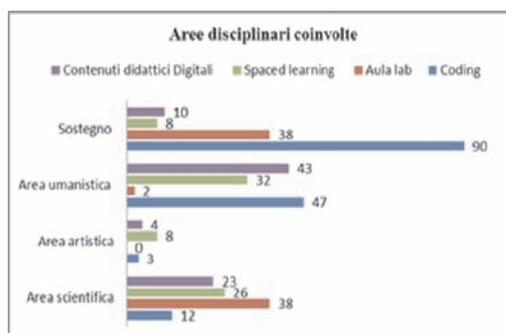


Fig. 1b. Ambiti disciplinari coinvolti

L'interesse per il Coding, con particolare riferimento alla scuola primaria, è in linea con le sostanziali modifiche dei programmi scolastici europei che stanno comprendendo il valore di introdurre e riprogettare i curricula integrando temi come la programmazione, la *computer science* e il pensiero computazionale. Lo sviluppo del pensiero computazionale (Wing, 2006) e delle abilità che fanno capo alla programmazione anche nei più piccoli è parte integrante della *maker education* (Dougherty, 2012) e risponde alla necessità di sostenere quelle competenze ritenute "chiave" nel 21° secolo, come la creatività, il pensiero logico e la capacità di soluzione dei problemi (Binkley et al., 2012).

Nel 2014, proprio quando la progettazione del percorso "Docenti InFormAzione" era in fase di avvio, il report della European Schoolnet (Balanskat e Engelhardt, 2014) diffondeva la situazione di 20 paesi europei mettendo in evidenza come il Coding, introdotto già 13 di questi paesi nei curricula della primaria e

della secondaria, fosse diventato in 7 paesi disciplina obbligatoria (tra cui Grecia, Inghilterra ed Estonia).

Si fa strada l'idea che avvicinando i bambini piccoli ai concetti e processi alla base della *computer science* attraverso un *playful learning process* (Lillard, 2013) venga favorito uno sviluppo logico-cognitivo con effetti sulla definizione di specifiche competenze (autoregolazione, peer seeking, problem posing e solving, peer reviewing) così come su quelle più proprie della *digital literacy* (Pennazio et al., 2013).

Tali cambiamenti guidano una lettura critica dei risultati emersi dalla sperimentazione condotta con le scuole pugliesi rispetto agli impatti sul curricolo, sul tipo di apprendimento, sulle pratiche didattiche e di reinforcement messe in atto dai docenti. Questo ragionamento viene fatto sulla base di un confronto con gli stimoli e le linee di pensiero offerte dalla letteratura scientifica sul tema del pensiero computazionale a scuola. I risultati emersi, con particolare riferimento al primo ciclo di istruzione, rappresentano le piste di interrogazione dei dati qualitativi acquisiti dalla esperienza EDOC@WORK3.0 al fine di comprenderne la declinazione rispetto alla disciplina di sperimentazione, all'integrazione tecnologica e alle dinamiche di classe attivate nel contesto italiano.

2. Il Coding a scuola. Le evidenze della ricerca empirica

La ricerca di (Lye e Koh, 2014) basata su studi empirici pubblicati negli ultimi anni sulla didattica della programmazione nei curricula scolastici fornisce una prospettiva rispetto alla capacità di differenti strategie didattiche hanno di sostenere le dimensioni del pensiero computazionale in particolari domini disciplinari. Le risposte alle domande di ricerca, con riferimento alla fascia K-12, sono di seguito riportate.

Domanda 1. Quali sono gli ambienti che permettono di integrare l'attività di programmazione nei curricula?

Lo studio mostra come dei 27 lavori scientifici presi in considerazione, 9 hanno il loro focus su campione di studenti dai 3 ai 12 anni. I linguaggi e gli ambienti di programmazione (es: Scratch e Logo), vengono adottati per i più piccoli nella didattica di varie discipline, e si distinguono in "lowfloor" (facili da usare) e "highceiling" (che permettono agli studenti di creare da sé). Per quanto concerne le discipline scientifiche in particolare emerge che gli studenti esternalizzano i concetti di base della matematica attraverso soluzioni come TOONTALK (Kahn et al., 2011) e LOGO (Fessakis et al., 2013) ma l'introduzione del Coding è possibile anche in ambiti umanistici e artistici dove progetti specifici sono previsti usando SCRATCH (Lee, 2010).

Domanda 2. Quali dimensioni del pensiero computazionale vengono valorizzate e sostenute?

Con riferimento alla classificazione delle dimensioni del pensiero computazionale di (Brennan e Resnick, 2012) gli studi che si concentrano sui *computational concepts*, hanno esaminato come gli studenti apprendono concetti base quali ad esempio quelli di calcolo o le variabili. Nel complesso rientrano in questa tipologia di analisi 23 studi tra quelli presenti nello studio empirico di (Lye e Koh, 2014): 16 quantitativi e 7 qualitativi. Dei 16 studi quantitativi solo 2 hanno avuto come target gli studenti K-12 e riportano risultati positivi del gruppo sperimentale con trattamenti come *on screen blocks* (schermo a blocchi) e *game-play* utiliz-

zati per insegnare concetti di calcolo (Kazakoff e Bers, 2012; Wang e Chen, 2010). Strategie come la *part-complete* (Garner, 2009) o l'*annotation* e il commento sui prodotti di Coding (Hsiao e Brusilovsky, 2011) sono da ritenersi particolarmente utili per gli alunni più deboli, e l'uso di animazioni ne migliora la motivazione allo studio (Urquiza-Fuentes e Velazquez-Iturbide, 2013). Dei 7 studi qualitativi, quelli condotti su studenti K-12 (Burke, 2012; Denner et al, 2012; Lee, 2010; Lin e Liu, 2012) tramite un approccio di tipo non sperimentale, riportano l'acquisizione di concetti di calcolo e computazione applicati alle materie curriculari.

Per quanto concerne la dimensione relativa alle *computational practices* gli studi che esaminano i processi di problem solving durante le attività di Coding (Fessakis et al., 2013; Esteves et al., 2011) attestano che la visualizzazione a video del linguaggio o codice migliora i processi di testing e debugging anche tra pari. Lo studio (Lin e Liu, 2012) mostra effetti molto positivi sulle pratiche computazionali e il loro sviluppo in modo incrementale attraverso momenti iterativi basati su un processo di revisione e miglioramento continuo.

Infine per quanto concerne la terza dimensione quella delle *computational perspectives* gli studi approfondiscono l'effetto che le attività di Coding hanno sulla comprensione di se stessi e delle relazioni che ciascuno studente ha con gli altri e con il mondo delle tecnologie. In 2 studi in particolare emerge la capacità dei ragazzi di esprimere se stessi creando oggetti interattivi tramite i linguaggi di programmazione: ad esempio alcuni di essi sono capaci di esprimere il concetto di "infinito" (Kahn et al., 2011) con TOONTALK e creare delle vere e proprie storie animate *in situazione* tramite SCRATCH (Burke, 2012).

Domanda 3. Quali sono gli approcci didattici che possono sostenere il pensiero computazionale?

Le ricerche riportano come l'acquisizione di *computational concepts* trovi nei feedback forniti dal docente e nelle modalità di gioco sperimentale un valido ausilio (Wang & Chen, 2010; Kose et al., 2013). Le prestazioni attese dagli studenti vengono guidate tramite momenti di rinforzo positivo o negativo (reinforcement). Quando gli studenti vengono invitati a riflettere sulla loro esperienza di Coding vengono sollecitate le dimensioni di *computational practices* e *computational perspectives* alla cui base vi è la capacità di rivedere e di ripensare il proprio elaborato anche grazie al coinvolgimento dei pari. La riflessione incoraggia la revisione del proprio lavoro intervenendo in modo critico sull'apprendimento (Søndergaard e Mulder, 2012; Yang, 2010) stimolando gli studenti al "pensare facendo". Impegnandosi nelle auto-spiegazioni (Kyungbin & Jonassen, 2011) e nella revisione dell'artefatto realizzato (Wang et al., 2012) tramite commenti e annotazioni (Hsiao e Brusilovsky, 2011), gli studenti mettono in atto processi di revisione, miglioramento e finalizzazione.

Un approccio didattico che punta sulla elaborazione delle informazioni sembra aiutare gli studenti nell'acquisizione di nozioni di calcolo, fornendo strutture cognitive per una efficace elaborazione. Si rileva inoltre come l'uso delle metafore possa favorire l'apprendimento dei concetti base della programmazione (Hui e Umar, 2011) mentre approcci più specifici quali appunto quello dell'analogia sono stati usati per rispondere ai diversi stili di apprendimento degli studenti (Hung, 2012; Ma et al., 2011). Chiedendo agli studenti di riorganizzare attraverso una rappresentazione visiva il loro processo di pensiero si contribuisce a minimizzare gli errori di comprensione e la costruzione dei propri giochi e programmi (Hui&Umar, 2011; Ismail et al, 2010; Ma et al, 2011; Urquiza-Fuentes e Velazquez-Iturbide, 2013) stimolando la riflessione lo sviluppo di una dimensione pratica del pensiero computazionale.

Infine secondo una visione più costruttivista gli studenti possono essere coa-

diuvati tramite appositi e differenti *scaffolds*: *computer scaffolding* (Jiau et al., 2009; Ma et al, 2011.; Moreno, 2012) *teachers scaffolding* (Burke, 2012.; Esteves et al, 2011.; Fessakis et al, 2013.; Kahn et al, 2011; Kazakoff e Bers, 2012; Lee, 2010) *parental guidance* (Lin e Liu, 2012) o *peer scaffolds* (Denner et al, 2012; Hui e Umar, 2011; Ismail et al., 2010).

3. Analisi delle pratiche pugliesi e confronto con i risultati internazionali

Al fine di dare spazio alla ricerca interpretativa che permette, anche per numeri piccoli, di ricostruire l'intenzionalità alla base delle azioni, sono stati selezionati 2 "cantieri sperimentali", ossia scuole pugliesi dove docenti e classi sono stati chiamati, in differenti periodi didattici, a far parte di una osservazione non partecipata sul campo tramite. Attraverso griglie di osservazione e *video recording* della situazione didattica è stato possibile fotografare in modo fedele come il metodo Coding ha trovato una sua integrazione nel contesto d'aula. Il confronto tra l'osservazione puntuale nei due cantieri con le progettazioni didattiche di un campione più vasto costituito da 56 docenti sperimentatori della metodologia Coding in altrettante classi pugliesi, ha permesso poi, tramite un meccanismo di *pattern-matching*, di confermare i dati di livello micro con quelli derivanti dallo studio a livello macro. Tale metodologia di analisi se da un lato ci permette di mettere a confronto i risultati dell'esperienza italiana con l'evidenza derivante dalla ricerca internazionale (Lye e Koh, 2014) recuperando alcune ricorrenze rispetto alle dimensioni investigate, dall'altro si configura come rispettosa dell'unità di analisi specifica, e delle condizioni che influenzano i risultati e le scelte.

4. Le osservazioni non partecipate nei cantieri sperimentali

L'osservazione non partecipata ha previsto la presenza di uno o più osservatori, chiamati a registrare le situazioni didattiche compilando una griglia a bassa strutturazione. Le ipotesi di natura implicita hanno permesso di individuare i fattori su cui costruire le domande della griglia: Coding e curriculum, dimensioni del pensiero computazionale sostenute dall'attività, tipo di supporto da parte del docente, ruolo della tecnologia e collaborazione tra pari. L'osservazione è stata accompagnata dalla registrazione video di momenti significativi per la messa in atto dell'innovazione a scuola.

4.1. Osservazione 1. Classe II. Scuola Primaria

La classe era costituita da 11 alunni (6 femmine e 5 maschi) di età compresa tra i 6 e i 7 anni. L'esperienza, costruita con riferimento alla programmazione annuale dei piani disciplinari e trasversali, è stata inserita nel curriculum di matematica con l'obiettivo di sostenere lo sviluppo di competenze logiche e di risoluzione di problemi. I bambini avevano a disposizione numeri da 1 a 40, segni + - / * <> di cui dovevano servirsi (con riferimento a concetti, fatti e regole dell'aritmetica) per inventare tre quiz che poi avrebbero *gamificato* attraverso l'uso di Scratch. L'attività era caratterizzata da due momenti: il primo di invenzione di quiz matematici, restituiti seguendo un form cartaceo distribuito dalle maestre, il secondo di "Coding" del testo scritto in gioco digitale e verifica con i pari.

La didattica laboratoriale (già sperimentata nei percorsi promossi dal progetto Cl@ssi 2.0) è stata favorita da una riorganizzazione dei banchi idonea al lavoro grupale (tre gruppi da 3 bambini e uno da 2) e la presenza dei Tablet consenti-

va di passare dal linguaggio di tipo testuale a quello multimediale. Si osserva una partecipazione equa di tutti i componenti, in tutti i gruppi con un alto livello di interazione: in particolare il coinvolgimento fisico ed emotivo aumentava durante l'uso del dispositivo mobile permettendo di attivare maggiori flussi comunicativi intra gruppo. Tutto il processo di codifica è stato declinato in termini collaborativi (fig. 2): gli alunni dialogavano durante la realizzazione dei quiz matematici e definiscono i ruoli in modo dinamico: due componenti prendevano il controllo dello strumento per portare avanti il compito, mentre, contestualmente, un terzo bambino compilava il form cartaceo registrando le operazioni effettuate.



Fig. 2. Momenti di lavoro intra gruppo

Le docenti ruotavano tra i banchi fornendo supporto specifico e incoraggiando gli alunni a svolgere il compito in autonomia, stimolandoli alla riflessione ed evidenziando i progressi e la soddisfazione per le prestazioni (fig. 3). La comunicazione inter gruppo ha avuto inizio nella fase di testing tra pari che richiedeva la socializzazione dei lavori e un percorso di riflessione volto a condividere le sulle cose apprese e a indicare le migliorie da apportare ai prodotti. In questa fase i gruppi lavoravano scambiandosi i Tablet per testare i quiz inventati dai compagni e programmati tramite SCRATCH (fig. 4).



Fig. 3. Supporto del docente



Fig. 4. Socializzazione del lavoro e peer evaluation

Le tecnologie mobili favoriscono una maggiore flessibilità nella gestione di tempi e spazi permettendo ai docenti di accompagnare i bambini in un processo di riflessione su errori e progressi facilitando lo sviluppo di capacità critiche.

4.2. Osservazione 2. Classe V. Scuola Primaria

La classe era composta da 28 bambini (11 maschi e 17 femmine) tra i 9 e 10 anni. L'esperienza il cui obiettivo principale era quello di favorire lo sviluppo di un profilo "autore" attraverso l'acquisizione dei concetti computazionali, si inseriva in una attività parallela al curriculum ordinario ed afferente ad un progetto Erasmus +.

Dopo una visita al frantoio del paese (da qui emerge il concetto di "aula estesa" al territorio) i bambini erano chiamati a fare delle ipotesi sulla produzione dell'olio, programmarle e verificarle (con un costante stimolo al pensiero logico) attraverso, anche in questo caso, l'uso di SCRATCH.

L'attività didattica ha beneficiato del tempo prolungato e delle co-presenze (grazie alle quali la docente prevalente ha potuto lavorare per piccoli gruppi). In un'aula tradizionale, divisi in gruppi da 2 o 3, i bambini della stessa fila collaboravano tramite l'uso condiviso di un computer fisso alla costruzione di ipotesi e alla realizzazione del gioco (fig. 5).



Fig. 5. Organizzazione intra gruppo

Il docente, durante il *cooperative learning* e il *pair programming*, valorizzava le proposte degli allievi attraverso azioni di reinforcement, stimolando specifiche aree di competenza e indirizzando i momenti maggiormente critici (fig. 6).



Fig. 6. Supporto del docente

Il confronto e la socializzazione del prodotto con i pari tramite LIM ha favorito le azioni di rivisitazione e miglioramento. L'ambiente cooperativo e non giudicante potenziava l'autostima e la creatività incentivando la messa in discussione degli elaborati (fig. 7).



Fig. 7. Socializzazione del lavoro e peer evaluation

La dimensione tecnologica si caratterizza per essere l'elemento su cui poggia l'inclusione di allievi speciali³ con alti livelli di partecipazione e motivazione.

5. Analisi delle progettazioni e confronto con le evidenze

Con riferimento allo studio di Lye e Koh (2014) e alle tre principali domande di ricerca in esso identificate (e precedentemente esposte e discusse) si è proceduto a comparare i risultati di tale ricerca empirica con un campione più vasto di docenti sperimentatori. Sono state analizzate 56 griglie di progettazione didattica e di rilevazioni delle esperienze. Tali griglie sono state compilate da 28 docenti di scuola primaria, 27 docenti di scuola secondaria di primo grado, 1 docente di scuola dell'infanzia. Le discipline oggetto di sperimentazione didattica sono riconducibili all'ambito storico-letterario, logico-matematico, scientifico, linguistico, con una minore presenza del sostegno (fig. 8).

Da una prima analisi testuale volta a comprendere meglio il profilo dei docenti sperimentatori è stato possibile risalire alle loro competenze pregresse in ambito tecnologico e ai loro atteggiamenti nei confronti della tecnologia. La maggioranza dei docenti, alla richiesta di riferire sulle proprie pratiche di insegnamento assistite dalle tecnologie, fa riferimento ad un uso abituale dando conto di una diffusa presenza della tecnologia nelle didattiche disciplinari (fig. 9).

3 Nel cantiere in oggetto si registravano quattro studenti BES provenienti dalla città vecchia e da quartieri a rischio e un'alunna con DSA riconosciuto.

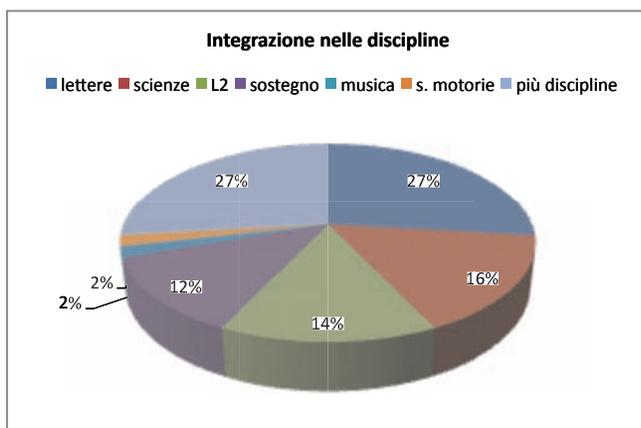


Fig. 8. Integrazione del Coding nelle discipline

Un numero minore di docenti riferisce una carente integrazione tecnologica dettata da una mancanza di risorse nella scuola. Una minoranza di docenti, tutti insegnanti di scuola primaria su area scientifica, dichiara una precedente esperienza sul Coding (maturata attraverso l'Orchestra del Codice di "Programma il Futuro") e l'uso didattico di software come Scribbler o di oggetti programmabili come BeeBot o MakeyMakey.

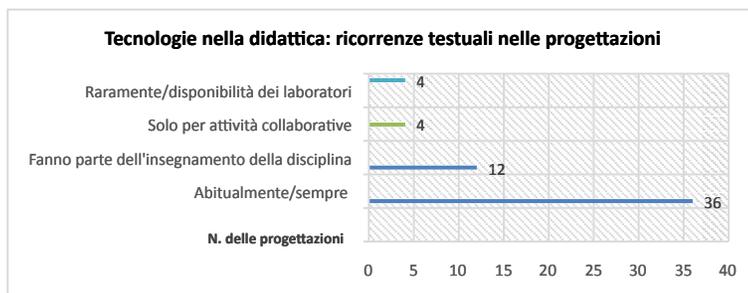


Fig. 9. Uso delle tecnologie: ricorrenze testuali nelle progettazioni

Dalla analisi delle ricorrenze di lemmi intesi come significativi nei testi delle progettazioni emerge che i docenti che fanno un uso abituale delle tecnologie nella disciplina (posizione emersa in 36 progettazioni su 56 analizzate) (fig. 9) sono anche quelli in possesso di certificazioni o attestati di frequenza a corsi di perfezionamento quali CertLim AICA, Epict Silver, Nuova ECDL e corsi di formazione INDIRE come Didatec e InnovaMusica. Solo in 12 progettazioni emerge una integrazione nella didattica disciplinare.

I docenti hanno poi espresso nelle loro progettazioni un atteggiamento verso la tecnologia globalmente positivo e motivato dalla sua efficacia in aula (fig. 10). Sempre attraverso l'individuazione di lemmi caratterizzanti l'atteggiamento, si prende nota di come tra i docenti più esperti sia possibile cogliere una capacità di riflessione e una maggiore consapevolezza nell'uso delle tecnologie nelle pratiche d'aula quotidiane. Le tecnologie incontrano le modalità del conoscere

dei ragazzi, e in particolare nelle progettazioni del sostegno le tecnologie vengono ritenute un utile ausilio per motivare allo studio gli studenti BES migliorando il loro tempo di concentrazione e permettendo la costruzione di percorsi individualizzati e inclusivi.

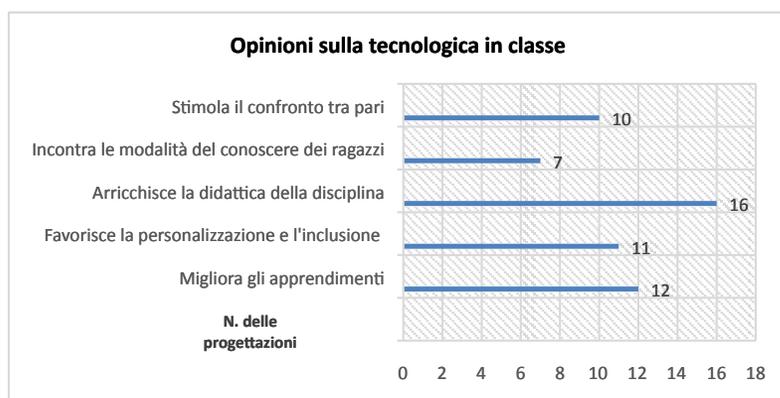


Fig. 10. Opinione sulle tecnologie: ricorrenze testuali nelle progettazioni

Proviamo ora rispondere, attraverso i dati dell'esperienza pugliese, alle tre domande di indagine presenti nel lavoro di (Lye e Koh, 2014)

Domanda 1. Quali sono gli ambienti che permettono di integrare l'attività di programmazione nei curricula?

Dall'analisi delle progettazioni dei docenti sperimentatori e dal confronto con le griglie di rilevazione ex-post dell'esperienza si rileva che, nella quasi totalità dei casi, l'*Initial Learning Environment* (Duncan et al, 2014) scelto per l'introduzione al Coding nella primaria e secondaria di primo grado è il programma SCRATCH, nelle sue diverse configurazioni (online e offline) e versioni (SCRATCH 4, SCRATCH 2.0, SCRATCH JR), in base all'età e all'expertise pregressa di docenti e degli stessi studenti. SCRATCH è stato integrato, in circa la metà dei casi, sulla base della tipologia di attività progettata e di prodotto finale previsto, con altri software come quelli per la creazione di mappe (es: SuperMappe) o per la creazione di ebook (es: BookCreator, Didapages).

Relativamente all'integrazione all'interno di specifici ambiti disciplinari docenti sottolineano l'efficacia del Coding per un uso veicolare della lingua straniera, permettendo di acquisire cultura digitale, abilità tecnologiche operative e contenuti linguistici allo stesso tempo. Ad esempio per i più piccoli è stato utilizzato il metodo del Coding per approfondire i concetti spaziali in lingua inglese. I docenti di matematica sottolineano che la sperimentazione è stata fortemente correlata agli obiettivi di *imparare a pensare in modo creativo* e *potenziare l'uso del linguaggio specifico disciplinare*. Con riferimento alle materie letterarie i docenti segnalano che l'attività di Coding sembra migliorare la capacità dei bambini di distinguere ed utilizzare diverse tipologie di forme narrative (es: la favola o semplici storie o racconti) costruite dai bambini seguendo una successione temporale dei fatti e rafforzando capacità creativa e comunicativa. La dimensione interdisciplinare dell'apprendimento emerge soprattutto nella scuola primaria dove lo stesso docente insegna un gruppo di materie di base e dove quindi c'è una minore parcellizzazione di saperi e una maggiore possibilità di programmare per competenze anche trasversali.

Domanda 2. Quali sono i risultati rispetto alle dimensioni del pensiero computazionale?

Con riferimento ai *computational concepts*, l'uso di ambienti basati su una interfaccia a blocchi sembra aver facilitato l'apprendimento di concetti di calcolo o di programmazione di base (con riferimento alle variabili e ai cicli). L'integrazione con soluzione di Mindmapping ha favorito lo sviluppo del pensiero logico ed analitico (nel caso ad esempio della costruzione di un algoritmo).

Per quanto concerne la dimensione relativa alle *computational practices* si nota che un approccio metodologico incentrato sul problem solving e sulla "didattica per dilemmi" ha, secondo i docenti, permesso di tradurre le conoscenze e le abilità acquisite in competenze, intese come *comportamenti osservabili e trasferibili in altri contesti* di conoscenza e di vita. Gli studenti durante la sperimentazione sembrano aver migliorato la capacità di interazione con i dispositivi. L'interfaccia amichevole basata sul concetto di game-play proposta dai software usati nelle classi stimola la produzione di inferenze e l'applicazione di specifiche proprietà in modo naturale. La visualizzazione del codice e la possibilità di ricevere feedback immediati sono ritenuti veri punti di forza in grado di coadiuvare pratiche riflessive alla base del testing e del debugging.

Per quanto attiene la terza dimensione, quella delle *computational perspectives*, si ritiene che le attività di Coding in aula abbiano avuto effetto sulla comprensione di se stessi e delle relazioni che ciascuno studente ha con gli altri, assumendo come indicatore rilevabile l'aumentata capacità di esprimersi e di comunicare non solo nell'uso e per l'uso delle tecnologie ma a livello di socializzazione e relazione interpersonale. Dalle osservazioni dei docenti emerge inoltre che il valore e l'efficacia della sperimentazione risiede anche nella dimensione collaborativa dell'apprendere poiché la scelta molto diffusa di organizzare il lavoro per piccoli gruppi è risultata inclusiva in quanto ha supportato gli alunni più deboli e coinvolto i meno motivati in un percorso attivo e stimolante.

Domanda 3. Quali approcci didattici possono sostenere il pensiero computazionale?

I docenti sperimentatori hanno ritenuto, in linea generale, di poter utilizzare i propri approcci e le proprie abituali strategie didattiche anche nell'ambito della sperimentazione del Coding.

Nella maggior parte dei casi la lezione frontale è stata sostituita da sessioni che privilegiano un tipo di approccio laboratoriale e collaborativo fra pari (lezione dialogata, classe capovolta, tutoring tra pari) con progettazione a classi aperte o in gruppi. Inoltre il tutoraggio tra pari risulta efficace e motivante per i bambini con disturbi dell'apprendimento o con bisogni educativi speciali. In questi casi le strategie più frequentemente utilizzate sono state quelle cooperative che risultano inclusive e che mirano al coinvolgimento di tutti gli alunni, nei lavori con gruppi eterogenei e autogestiti. In alcuni casi si è previsto l'utilizzo di materiale semplificato e di misure compensative e dispensative (ad esempio sintesi vocale) secondo le necessità. D'altra parte la conversazione guidata e la correzione reciproca sono risultate preziose, a giudizio dei docenti, per guidare gli studenti all'elaborazione attiva e costruttiva dei contenuti di apprendimento e allo sviluppo del pensiero logico e procedurale.

Conclusioni

La sperimentazione del Coding in Puglia ha guidato i docenti a selezionare software e programmi in grado di incoraggiare nuove esperienze di apprendimento nei più piccoli riuscendo da un lato a lavorare sui *computational concepts* (facendo apprendere ai bambini i concetti di calcolo o le variabili) e dall'altro, nel caso dei più grandi, a mettere in campo processi più specifici relativi alle *computational practices* (problem solving) e *computational perspectives* (peer review). Nella maggior parte dei casi sono state utilizzate versioni diverse dello stesso ambiente di programmazione di tipo visuale: SCRATCH JNR per i più piccoli per fa apprendere ai bambini il concetto di pianificazione e sequenza e SCRATCH per i più grandi (per richiamare sia l'astrazione sia funzioni, variabili, iterazione, strutture di dati indicizzate, esecuzione condizionale) permettendo agli studenti di monitorare la sequenza di azioni eseguite, coinvolgendoli nella manipolazione degli oggetti e fornendo scaffolding immediati sull'impatto delle loro azioni.

Le strategie di reinforcement e di riflessione tra pari adottate dal docente sono state funzionali alla buona riuscita dell'esperienza. Le esperienze didattiche improntate a tali metodologie si sono rivelate estremamente inclusive durante la sperimentazione: i ragazzi più dotati e quelli con difficoltà di apprendimento (in particolare i DSA) sono stati in questo modo accomunati nell'elaborazione di storie con Scratch o nella progettazione, nella costruzione, nella programmazione di piccoli robot lavorando in gruppo e mettendo in mostra i propri talenti e le proprie intelligenze diverse. Valorizzando la peer education è venuto meno quell'imperativo psicologico ad essere corretti (Postman, 2001) e gli errori sono diventati l'elemento portante di un percorso (Popper, 2000) in grado di alimentare ipotesi con cui confrontarsi in una concezione dinamica e non statica della conoscenza (Postman, 2001).

Riferimenti bibliografici

- Balanskat, A., & Engelhardt, K. (2014). Computing Our Future: Computer Programming and Coding-Priorities, School Curricula and Initiatives Across Europe. *Technical report European Schoolnet*. October. Retrived from: http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887. [Ultima consultazione 11/07/2016].
- Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., et al. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw, & E. Care (Eds.). *Assessment and teaching of 21st century skills* (pp. 17–66). Dordrecht, Netherlands: Springer.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. In *Annual American Educational Research Association meeting*. AERA 2012. Vancouver, BC, Canada.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as writing in the middle school classroom. *Journal of Media Literacy Education*, 4(2), 121–135.
- Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1), 240–249.
- Dougherty, D. (2012). The maker movement. *Innovations*, 7(3), 11-14.
- Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 60-69). New York, NY: ACM.
- Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*, 42(4), 624–637.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.

- Garner, S. (2009). A quantitative study of a software tool that supports a part complete solution method on learning outcomes. *Journal of Information Technology Education*, 8, 285–310.
- Hsiao, I. H., & Brusilovsky, P. (2011). The role of community feedback in the student example authoring process: An evaluation of AnnotEx. *British Journal of Educational Technology*, 42(3), 482–499.
- Hui, T. H., & Umar, I. N. (2011). Does a combination of metaphor and pairing activity help programming performances of students with different self-regulated learning level? *The Turkish Online Journal of Educational Technology*, 10(4), 122–129.
- Hung, Y.-C. (2012). The effect of teaching methods and learning style on learning program design in web-based education systems. *Journal of Educational Computing Research*, 47(4), 409–427.
- Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). The effects of mind mapping with cooperative learning on programming performance, problem solving skill and metacognitive knowledge among computer science students. *Journal of Educational Computing Research*, 42(1), 35–61.
- Jiau, H. C., Chen, J. C., & Ssu, K.-F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education*, 52(4), 555–562.
- Kahn, K., Sendova, E., Sacristán, A. I., & Noss, R. (2011). Young students exploring cardinality by constructing infinite processes. *Technology, Knowledge and Learning*, 16(1), 3–34.
- Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371–391.
- Kose, U., Koc, D., & Yucesoy, S. A. (2013). Design and development of a sample “computer programming” course tool via story-based e-learning approach. *Educational Sciences: Theory & Practice*, 13(2), 1235–1250.
- Kyungbin, K., & Jonassen, D. H. (2011). The influence of reflective self-explanations on problem-solving performance. *Journal of Educational Computing Research*, 44(3), 247–263.
- Lee, Y.-J. (2010). Developing computer programming concepts and skills via technology-enriched language-art projects: A case study. *Journal of Educational Multimedia and Hypermedia*, 19(3), 307–326.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- Ma, L., Ferguson, J., Roper, M., & Wood, M. (2011). Investigating and improving the models of programming concepts held by novice programmers. *Computer Science Education*, 21(1), 57–80.
- Moreno, J. (2012). Digital competition game to improve programming skills. *Educational Technology & Society*, 15(3), 288–297.
- Pennazio, V., Traverso A., Parmigiani D. (2013). Digital literacies a scuola. I profili digitali degli studenti. *TD Tecnologie Didattiche*, 21 (1), 35–40.
- Popper, K. (2000) *I due problemi fondamentali della teoria della conoscenza*. Milano: Mondolibri.
- Postman, N. (2001) Fallibilismo ed educazione. In L. Binanti (a cura di). *Pedagogia, epistemologia e didattica dell'errore*. Cosenza: Rubbettino.
- Søndergaard, H., & Mulder, R. A. (2012). Collaborative learning through formative peer review: Pedagogy, programs and potential. *Computer Science Education*, 22(4), 343–367.
- Urquiza-Fuentes, J., & Velazquez-Iturbide, J. A. (2013). Toward the effective use of educational program animations: The roles of student’s engagement and topic complexity. *Computers & Education*, 67, 178–192.
- Wang, L. C., & Chen, M. P. (2010). The effects of game strategy and preference matching on flow experience and programming performance in game-based learning. *Innovations in Education and Teaching International*, 47(1), 39–52.
- Wang, Y., Li, H., Feng, Y., Jiang, Y., & Liu, Y. (2012). Assessment of programming language learning based on peer code review model: Implementation and experience report. *Computers & Education*, 59(2), 412–422.
- Wing, J. M. (2006). *Computational thinking*. Communications of the ACM, 49(3), 3–35.