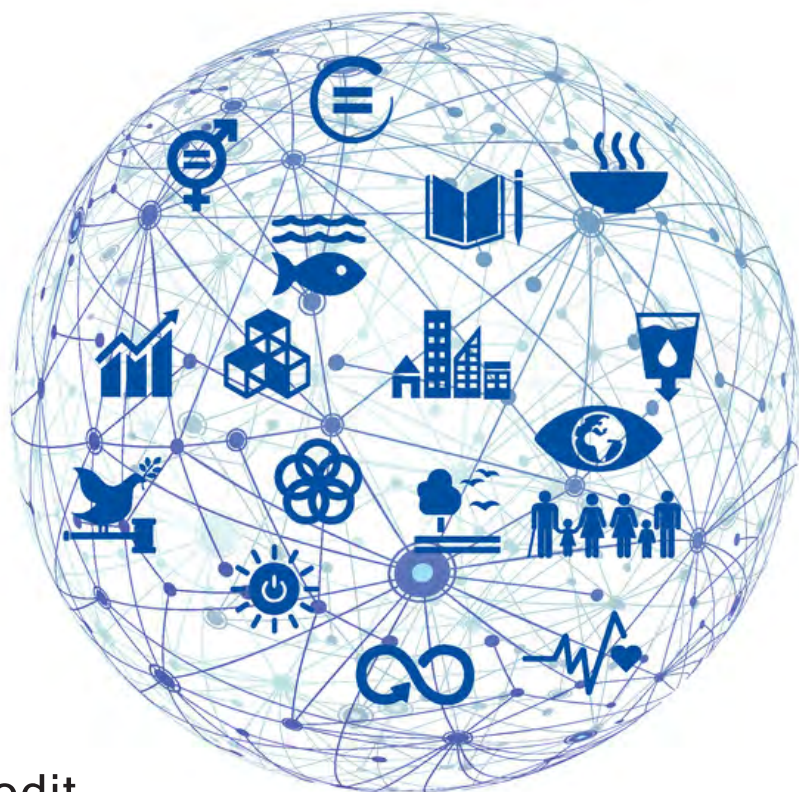


a cura di
PIERPAOLO LIMONE e DAVIDE PARMIGIANI

Modelli pedagogici e pratiche didattiche

per la formazione iniziale
e in servizio degli insegnanti



Progedit



Direttore scientifico:

Pierpaolo Limone (Università di Foggia)

Comitato scientifico e di referaggio:

José Luis Rodríguez Illera (Università di Barcellona)

Gunther Kress (Institute of Education - Londra)

Pierpaolo Limone (Università di Foggia)

Laura Messina (Università di Padova)

Nicola Paparella (Università del Salento)

Pier Cesare Rivoltella (Università Cattolica - Milano)

Luisa Santelli (Università di Bari)

Philippe Verhaegen (Università di Louvain-la-Neuve)

Paolo Vittoria (Università Federale di Rio de Janeiro)

© 2017 Progedit
Prima edizione marzo 2017

Progedit – Progetti editoriali srl
Via De Cesare 15 – 70122 Bari
Tel. 0805230627
Fax 0805237648
www.progedit.com
e-mail: info@progedit.com

Volume finanziato nell'ambito del P.A.R.
del Dipartimento di Studi Umanistici
dell'Università degli studi di Foggia

ISBN 978-886194-293-6

Proprietà letteraria
Progedit – Progetti editoriali srl, Bari

a cura di
Pierpaolo Limone
e Davide Parmigiani

Modelli pedagogici
e pratiche didattiche
per la formazione iniziale
e in servizio degli insegnanti

Saggi di:

M. Baldassarre, M. Banzato, I. Brunetti,
M. Brunetti, I. Bruni, G. Cappuccio,
A. Carenzio, R. Carro, F. Ceretti,
F. Chiusaroli, M. Cinque, B. Clemente,
F. Corona, L. Da Re, L. De Cani,
T. De Giuseppe, A.E. Ena, A. Epasto,
F. Falcinelli, M. Fantin, L. Fedeli,
C. Ferranti, S. Ferrari, M. Filomia,
C. Gaggioli, A. Garavaglia, E. Gülbay,
A. La Marca, L. Lattanzi, L. Longo,
L. Luciani, G.R. Mangione, L. Menichetti,
E. Messa, S. Mori, A.M. Murdaca,
E. Nini, A. Nuzzaci, P. Oliva,
M. Pentucci, C. Petrucco, L. Petti, M. Piva,
M. Ranieri, P.C. Rivoltella, A. Rosa,
P.G. Rossi, M. Sabatini, A. Scarinci,
N. Scognamiglio, C. Sirignano,
D. Smeriglio, P. Tosato, C. Zabaglio



La collana “Studi e Ricerche
sull’Educazione Mediale” è referata.

Ciascun manoscritto è sottoposto
a un attento processo
di *peer review* internazionale.

Il manoscritto, reso anonimo,
viene inviato al comitato scientifico
che seleziona i nomi dei due revisori esterni
tra docenti di chiara fama
dello specifico settore disciplinare.

Per ulteriori informazioni
sulla procedura del referaggio
o per sottoporre il manoscritto
si può consultare il sito:
www.eridlab.unifg.it

Ai sensi della legge sul diritto d’autore
e del codice civile, è vietata
la riproduzione di questo libro o di parte
di esso, realizzata con qualsiasi mezzo
(elettronico, meccanico, fotocopie, microfilms,
registrazioni o altro),
anche ad uso interno o didattico.

Le fotocopie per uso personale del lettore
possono essere effettuate nei limiti
del 15% di ciascun volume dietro pagamento
alla SIAE del compenso previsto
dall’art. 68, commi 4 e 5,
della legge 22 aprile 1941 n. 633.

INDICE

Introduzione <i>di P. Limone, D. Parmigiani</i>	1
Premessa <i>di P.G. Rossi</i>	3
LE TECNOLOGIE E IL CAMBIAMENTO NEI PROCESSI DI CONOSCENZA	
Una ricerca esplorativa sul coding e la valutazione del pensiero computazionale <i>di M. Baldassarre, I. Brunetti, M. Brunetti</i>	23
Jetl@g: Jordan education technologies leading @ generations. Tecnologie, scuola, adolescenti in Giordania tra tradizione e innovazione <i>di A. Carenzio, L. De Cani, C. Zabaglio, P.C. Rivoltella</i>	37
Le nuove tecnologie come strumenti per promuovere un cambiamento efficace della scuola: uno studio esplorativo <i>di R. Carro, S. Mori</i>	56
La scrittura in emoji per l'educazione linguistica e interculturale <i>di F. Chiusaroli</i>	68
Piano nazionale scuola digitale ed editoria scolastica <i>di F. Falcinelli, E. Nini</i>	79
La costruzione collaborativa di una conoscenza con Wikipedia: le percezioni degli studenti in un corso di laurea magistrale <i>di C. Petrucco, C. Ferranti, L. Da Re</i>	90

Paesaggi sonori. Un ambiente d'apprendimento significativo e intenzionale in remoto per educare all'ascolto sviluppando episodi di apprendimento situati <i>di N. Scognamiglio</i>	103
Fare coding per emanciparsi <i>di S. Ferrari, G.R. Mangione, A. Rosa, P.C. Rivoltella</i>	114
La didattica flipped for inclusion <i>di T. De Giuseppe, F. Corona</i>	132
LE TECNOLOGIE E LA FORMAZIONE INIZIALE E IN SERVIZIO DEGLI INSEGNANTI	
Self-efficacy degli insegnanti in attività di coding: uno studio di caso nella primaria e secondaria di primo grado <i>di M. Banzato, P. Tosato</i>	157
La competenza digitale dei futuri insegnanti della scuola primaria e dell'infanzia <i>di G. Cappuccio</i>	173
PNSD e eTwinning per la formazione degli insegnanti a garanzia di una buona scuola digitale <i>di A.E. Ena, B. Clemente, A. Scarinci</i>	186
Insegnare e imparare nelle classi digitali <i>di F. Falcinelli, C. Gaggioli</i>	198
Tecnologie e formazione in servizio degli educatori 0-6: un'indagine a livello europeo <i>di L. Fedeli, C. Sirignano</i>	210
Uno studio di caso sulla flipped classroom tra università e scuola <i>di E. Gülbay</i>	223
Il modello flipped learning per promuovere lo sviluppo delle competenze docimologiche dei futuri insegnanti di scienze	

della formazione primaria <i>di L. Longo</i>	229
Fattori individuali e atteggiamenti degli insegnanti per un uso efficace delle tecnologie digitali. Un progetto di ricerca <i>di A.M. Murdaca, A. Epasto, D. Smeriglio, P. Oliva</i>	240
ICT, beni culturali e formazione iniziale e in servizio degli insegnanti: il progetto “Il museo in...click!” <i>di A. Nuzzaci</i>	258
Mediatori digitali e trasformazioni nelle pratiche didattiche <i>di M. Pentucci</i>	272
Futuri insegnanti e competenza mediale: sperimentazione di training scenario in modalità blended <i>di M. Ranier, I. Bruni</i>	286
Formare i futuri docenti al pensiero computazionale attraverso un approccio laboratoriale <i>di F. Falcinelli, M. Filomia, M. Sabatini</i>	296
iTunes U: sperimentare la flipped nella formazione iniziale degli insegnanti <i>di A. La Marca</i>	309
Riprogettazione del setting del laboratorio di tecnologie didattiche per la formazione iniziale degli insegnanti <i>di A. Garavaglia, L. Petti</i>	327
 L'AGENDA DIGITALE	
Nuovi insegnanti per nuove discipline? Il caso dell'indirizzo audiovisivo e multimediale nei licei artistici e le classi di concorso riformate <i>di M. Piva</i>	343

Sviluppare le competenze digitali degli studenti: framework
e linee guida per un intervento didattico
di L. Menichetti 353

QUALI SPAZI E QUALE RUOLO PER LA MEDIA EDUCATION

Serious games e MOOCs: quale ruolo per il training
delle soft skills?
di M. Cinque 373

Approssimazioni: EAS, media education e partecipazione
nel territorio
di M. Fantin 383

Oltre il conformismo digitale
di L. Lattanzi 396

Educazione ai media e linguaggio delle immagini in movimento:
una proposta di curriculum per la scuola
di L. Luciani 406

Health promoting media literacy education. Promuovere comportamenti
alimentari e stili di vita salutari con social media e apps
di E. Messa, M. Cinque 417

Educare i media. Una buona “scuola” nella cultura digitale: dalla media
education alla meducazione
di F. Ceretti 429

LE TECNOLOGIE E IL CAMBIAMENTO
NEI PROCESSI DI CONOSCENZA

UNA RICERCA ESPLORATIVA SUL CODING
E LA VALUTAZIONE DEL PENSIERO
COMPUTAZIONALE

di Michele Baldassarre, Imma Brunetti, Maria Brunetti

Abstract

Diversi studi sulla ricerca educativa (Ring, 2006; Papera, 1975, 1981; Wilensky, 2001; Wilensky e Reisman, 2006; Wilensky e Resnik, 1999; Kolodner et al., 2003; Puntambekar & Kolodner, 2005; Kali & Linn, 2009; Linn, Lee, Tinker, Husic, & Chiu, 2006; Kafai, 2006; Kafai & Ching, 2001) dimostrano che gli studenti apprendono le strategie del pensiero computazionale mentre studiano e gli insegnanti riescono a modellare il pensiero verso queste strategie e adeguare misure di orientamento (Brennan & Resnick, 2012). In molti casi un elemento-chiave per l'insegnamento di queste strategie di pensiero è costituito dalla funzionalità offerta dagli ambienti di apprendimento idonei, particolarmente utili nel promuovere il pensiero computazionale (Wing, 2006). Affinché, però, ciò non risulti solamente una mera applicazione di giochi interattivi, il nostro studio verte principalmente sulla valutazione di quest'attività come prodotto e processo.

La ricerca proposta è stata svolta in una classe prima di una scuola primaria nella provincia di Bari. Seguendo il programma del corso base di *coding* dal sito implementato dal MIUR su "Programma il futuro", la docente ha sottoposto agli allievi un programma di *coding* attraverso giochi interattivi e collaborativi nel quale gli studenti si sono cimentati in un vero e proprio studio di programmazione informatica. Il nostro obiettivo principale verte sul modo in cui tali attività sostengono il pensiero computazionale e come valutarle. Abbiamo individuato delle dimensioni-chiave nel processo di insegnamento-apprendimento di *coding* rinvenibili nei concetti, nelle pratiche e nelle prospettive computazionali che verranno analizzate insieme alle interviste degli studenti e al loro prodotto.

Tali dati saranno successivamente triangolati con il pensiero dell'insegnante e i relativi indicatori sulla trasposizione didattica interna ed esterna.

I risultati attesi vogliono dimostrare che il pensiero computazionale ha natura interdisciplinare poiché sostiene il naturale processo di insegnamento e apprendimento di tutte le discipline ed è possibile valutarlo non come da-

to intuitivo ma come prodotto finale in relazione allo spazio, al tempo, alla lettura e alla scrittura.

1. *Il pensiero computazionale*

Computational thinking è il titolo di un articolo pubblicato da Jeanette Wing (Wing, 2006) su *Communications of ACM*, nel quale si pone enfasi sul concetto di *skills*, indispensabili per tutti, non solo agli informatici. Questo modo di pensare viene proposto come quarta abilità di base, dopo la lettura, scrittura ed il calcolo da insegnare ai bambini.

Già Saymond Papert (1996) introdusse la locuzione *computational thinking* dimostrando l'idoneità dell'insegnamento della matematica mediante la didattica della programmazione LOGO, proposta dal MIT, sulla base dei principi del costruzionismo. Dunque, il computer risulterebbe essere un ottimo strumento per realizzare gli artefatti cognitivi per mezzo del quale costruire rappresentazioni reali del mondo in cui si interagisce. Successivamente viene declinato come

the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent (Cuny, Snyder & Wing, 2010).

In definitiva è un'attività mentale per la risoluzione di un problema che necessita di una soluzione computazionale ad opera sia di una macchina che dell'uomo, oppure, e sembrerebbe l'aspetto più importante, dell'unione della macchina con l'uomo. La tecnologia deve sostenere la diffusione del pensiero computazionale per capire vari aspetti di un problema, valutare le diverse strategie computazionali applicando o adattando le diverse strategie in diversi usi. Gli studenti apprendono le strategie del pensiero computazionale mentre studiano e gli insegnanti riescono a modellare il pensiero verso queste strategie e adeguare misure di orientamento (Brennan & Resnick, 2012). Questo è possibile perché il pensiero computazionale è astrazione, è un'abilità fondazionale, è un modo di pensare creativo, stimola il pensiero metacognitivo, riguarda le idee, i concetti e gli approcci per risolvere un problema ma anche per gestire la propria vita (Wilensky, 2001; Wilensky & Reisman, 2006; Wilensky & Resnik, 1999; Kolodner et al., 2003; Puntambekar & Kolodner, 2005; Kali & Linn, 2009).

Il pensiero computazionale differisce dal pensiero matematico-critico perché pone nuove abilità nel risolvere un problema, usa tecniche per prove ed errori, iterazioni, il provare a caso, ed infine è legato ad uno strumento (Barr & Conery, 2011).

L'idea della programmazione in sé come linguaggio per imparare ad imparare non si è rivelata vincente; è, invece, vincente l'idea di legare gli oggetti e la programmazione come competenza contestualizzata generando nuovi ambienti che eliminano gli errori di sintassi e riducono la complessità del codice di programmazione (Linn, Lee, Tinker, Husic & Chiu, 2006; Kafai, 2006; Kafai & Ching, 2001).

2. *Il progetto in una scuola primaria*

La proposta parte dall'influenza di due lavori nel campo della ricerca informatica educativa. Il primo focalizzato su un costruttivismo radicale che sostiene l'importanza di tale pensiero non per l'apprendimento del linguaggio informatico, della sintassi o della semantica, ma per constatare la programmazione come un processo creativo ed intelligente di sintesi e *problem solving* per la risoluzione di problemi realistici in contesti autentici (Greening, 2000). Il lavoro di Ben-Ari (2001), invece, si focalizza su un costruttivismo più moderato e personale poiché dà importanza alla conoscenza pregressa nell'apprendimento della programmazione. Lo studente che incontra un nuovo sistema, costruisce un nuovo modello mentale indipendentemente dal fatto che questo venga insegnato o no. Diversi studi mostrano quanto gli studenti (agenti computazionali) imparino le strategie del pensiero computazionale mentre studiano e come i docenti adeguino misure di orientamento.

Dunque il nostro primo obiettivo verte sul modo in cui tali attività promuovono il pensiero computazionale e su come valutare tali attività. La domanda seguente è stata come valutare tali attività purché non risulti la sperimentazione solo una utilizzazione sterile e mera applicazione di giochi interattivi. Dunque il processo valutativo verte sul prodotto e sul processo.

La ricerca ha un impianto qualitativo; si è avvalsa di strumenti qualitativi tra i quali le video-osservazioni della lezione, le interviste semistrutturate e il prodotto progettuale con Scratch. Scratch è un linguaggio di programmazione che usa blocchetti colorati che rappresentano le istruzioni per raccontare una storia interattiva, giochi e animazioni; è anche una *community* per condividere le creazioni ed esplorare quelle degli altri membri. Scratch è sta-

to creato dal MIT Media Lab per studiare e valutare il pensiero computazionale. In particolare attraverso questo linguaggio è possibile sviluppare il pensiero computazionale, nell'ambito del costruzionismo, mediante tre dimensioni-chiave individuate nello stesso:

- concetti computazionali: i concetti che gli sviluppatori utilizzano quando programmano (sequenze, eventi, parallelismo, condizionali, operatori, dati);
- pratiche computazionali: pratiche che si apprendono programmando (essere incrementali e iterativi, *testing* e *debugging*, riuso e *remixing*, astrazione e modularizzazione);
- prospettive computazionali: i modi di vedere il mondo e se stessi che i ragazzi sviluppano grazie alla programmazione (esprimersi, connettersi, farsi domande).

2.1 Fasi della ricerca

Il percorso proposto al gruppo classe è stato quello presentato nel progetto Programma il Futuro, Corso 1 rivolto ai bambini che non sanno leggere; questo per agevolare gli alunni e non sovraccargarli di difficoltà in quanto l'abilità di lettura e comprensione era in via di acquisizione. Abbiamo seguito le diverse lezioni proposte in piattaforma che gradualmente hanno offerto agli alunni la possibilità di impadronirsi dei concetti necessari ad affrontare il compito finale del percorso: costruire una storia utilizzando dei comandi a blocchi per muovere e far parlare i personaggi utilizzando il programma Scratch.

La ricerca si è articolata in tre step.

Fase 0: in questa fase ha avuto inizio l'esplorazione del *coding* a partire dal 7 dicembre, ora del codice, data stabilita dal MIUR per iniziare il programma in tutta Italia. L'intero percorso si è articolato in ore di lavoro-gioco in laboratorio informatico, e ore di lezione tradizionale in palestra o in giardino. Gli alunni, prima che al pc, hanno sperimentato fisicamente con esercizi di gioco l'attività del programmare movimenti e azioni come se loro stessi fossero il famoso cane o gatto del programma e un compagno di volta in volta diverso fosse il programmatore che scrive il codice. Allo step teorico/tradizionale svolto in palestra si è associato lo step pratico con l'ausilio del pc.

Ciascun bambino ha realizzato la propria storia al pc; l'insegnante ha raccolto l'idea e le parole più coinvolgenti e chiamando l'autore di quelle particolari parole o movimenti, ha chiesto di riprodurle su un altro pc. Que-

sta operazione si è ripetuta fino a che abbiamo creato ciò che piaceva a tutti raccogliendo i suggerimenti di tutti. Alla fine è venuto fuori un programma di una decina di secondi dal tema: “Stare bene insieme anche nella differenza”. Il lavoro si è svolto attraverso le seguenti caratterizzazioni:

- decomposizione dei problemi in piccole parti facilmente risolvibili,
- astrazione: semplificare e generalizzare i problemi,
- negoziazione: lavorare in gruppi e poi lavorare per integrare le soluzioni in un tutto,
- costruzione del consenso: lavorare per avere l'appoggio del gruppo attorno a un'idea.

Fase 1: individuazione e descrizione dei concetti, delle pratiche e delle prospettive computazionali individuate dal MIT Media Lab mediante l'utilizzo di Scratch.

Fase 2: valutazione di queste dimensioni attraverso la loro analisi; somministrazione delle interviste semistrutturate (Bichi, 2002) tese ad individuare le reazioni comportamentali degli studenti, le pratiche che si apprendono programmando, ovvero l'essere incrementali, iterativi, *testing* o *debugging*, riuso e *remixing*, astrazione e modularizzazione, e le prospettive computazionali che ognuno di noi ha per organizzare la visione di se stessi e del mondo sviluppate grazie alla programmazione, come ad esempio l'esprimersi, il connettersi ed il farsi domande. Infine dati raccolti sulla riflessione dell'azione da parte del docente stesso e integrazione di queste con gli indicatori della trasposizione didattica (logica valoriale, didattica e dell'apprendimento) (Damiano, 2007; Altet, 2003; Develay, 1995; Festermacher, 1986; Shulman, 1986).

3. *Analisi dei dati: i concetti del pensiero computazionale*

Quando si progetta un media interattivo con Scratch, entrano in gioco una serie di concetti computazionali, comuni a molti linguaggi e identificati come mappatura di blocchi di programmazione: sequenze, *loop*, parallelismo, eventi, condizionali, operatori e dati. Per ogni concetto si cercherà di spiegare i passi effettuati.

Concetto: sequenza.

Un concetto fondamentale nella programmazione è che una particolare attività o compito viene espresso in una serie di singole fasi o istruzioni ese-

gite dal computer, come ad esempio una ricetta o una serie di istruzioni di programmazione che specificano un comportamento o un'azione. Ad esempio, un oggetto può essere programmato per muoversi a breve distanza attraverso una sequenza di istruzioni.

Concetto: *loop*.

Nell'esempio precedente l'oggetto viene programmato per spostarsi e attendere alcuni secondi. Nel concetto di *loop* invece l'oggetto deve spostarsi e attendere molte più volte. Un *loop* è un meccanismo di istruzione di esecuzione della stessa sequenza più volte, ovvero una sequenza di istruzioni.

Concetto: eventi.

Un evento, inteso come qualcosa che causa un'altra cosa, è molto importante nei media interattivi. Un esempio può essere il tasto di avvio di un video musicale oppure la collisione di due oggetti che aumenta il punteggio di un gioco; o, ancora, premendo il tasto spazio l'oggetto si muove nello spazio in su o in basso oppure cliccando il mouse verrà visualizzato il fumetto del personaggio per alcuni secondi.

Concetto: parallelismo.

La maggior parte dei linguaggi di programmazione moderni supportano il parallelismo, ovvero sequenze di istruzioni che accadono allo stesso tempo. Scratch supporta il parallelismo tra gli oggetti: ad esempio, una scena di festa da ballo potrebbe coinvolgere diversi personaggi che ballano contemporaneamente, ognuno con una sequenza unica di istruzioni di danza. Scratch supporta anche il parallelismo all'interno di un singolo oggetto. Nel nostro caso sperimentale mentre il cane si avvicina, il gatto dice ciao.

Concetto: condizionale.

Un altro concetto-chiave nei media interattivi è il condizionale, ovvero la capacità di prendere decisioni in base a determinate condizioni come espressione di molteplici risultati. Un esempio è stato utilizzare il blocco *if* per determinare la visibilità di un oggetto.

Concetto: operatore.

Gli operatori permettono al programma di eseguire manipolazioni numeriche tra cui l'addizione, la sottrazione, la divisione e le operazioni di stringa come la concatenazione e la lunghezza delle stringhe.

Concetto: dati.

Il dato comporta la memorizzazione, il recupero e l'aggiornamento dei valori. Scratch offre attualmente due contenitori per i dati: la variabile che può mantenere un singolo numero o una stringa e le liste che possono mantenere un insieme di numeri o stringhe.

Calcolare il punteggio in un gioco è una motivazione frequente per i giovani designer per esplorare le variabili.

4. *Le pratiche del pensiero computazionale*

Per avere un quadro chiaro ed esaustivo del pensiero computazionale è necessario ottenere altri dati che corrispondano alle pratiche progettuali, ovvero al processo del pensare e dell'imparare, dunque all'andare oltre ciò che si sta imparando. Le dimensioni-chiave delle pratiche sono state raccolte con l'aiuto delle interviste tese ad identificare appunto i processi di costruzione. Ci sono 4 gruppi principali di pratiche:

1. incrementali e iterativi;
2. *testing* e *debugging*;
3. riutilizzo e *remix*;
4. astrazione e modularizzazione.

Pratica: essere incrementale e iterativo.

Progettare un media interattivo non è un processo privo di errori e sequenziale poiché è necessario prima di tutto identificare un concetto per delineare il progetto, poi sviluppare un piano per la progettazione, e poi attuare il progetto in codice. È un processo adattivo perché la progettazione iniziale potrebbe cambiare in risposta all'avvicinarsi della soluzione che si svolge a piccoli passi. Nelle interviste si nota l'essere iterativi attraverso il desiderio di immaginazione e costruzione. Quindi giocare e creare, provare e sviluppare ulteriormente, in base alle proprie esperienze e nuove idee.

Pratica: *test* e *debug*.

Le cose raramente (se non mai) funzionano esattamente come le immaginiamo; è fondamentale per i progettisti sviluppare strategie per affrontare se non anticipare i problemi. Nelle video-osservazioni, gli alunni descrivono le loro diverse pratiche provando e riprovando attraverso tentativi ed errori o con il sostegno del docente.

Pratica: riutilizzo e *remix*.

Questa pratica è stata possibile poiché i bambini hanno adattato alla loro storia e soprattutto alle loro idee, i personaggi presi dalle linee guida che il MIUR ha identificato in questo programma (Code.org) su ciò che la comunità di Scratch condivide.

Pratica: astrazione e modularizzazione.

L'astrazione e la modularizzazione sono pratiche importanti per la progettazione di un *problem solving*. In Scratch, gli alunni dovevano progettare una situazione problematica o una storia, mettendo insieme piccole parti aventi ciascuno un obiettivo da raggiungere. I progettisti impiegano l'astrazione e la modularizzazione a più livelli, ovvero dal lavoro iniziale di concettualizzazione del problema al passo successivo di traduzione del concetto in pile di codici che definiscono comportamenti o azioni. Ci può essere la pila di codice che controlla il movimento dell'oggetto sullo schermo; la seconda pila di codice controlla l'aspetto visivo dell'oggetto; la terza pila di codice controlla i vari eventi associati con gli ostacoli, come ripristinare il livello se l'oggetto collide con un pericolo. In questa fase gli alunni si pongono dal punto di vista di chi legge.

5. *Le prospettive del pensiero computazionale*

Nelle interviste abbiamo voluto rilevare l'evoluzione nella comprensione di sé stessi, nelle relazioni con gli altri e con il mondo tecnologico che li circonda, ovvero i cambiamenti di prospettiva.

- La prima prospettiva la definisco di espressione perché gli alunni hanno visto questa strategia di pensiero come un mezzo per creare giocando: “posso giocare e creare”, “scrivere la storia” ed infine “mi piaceva quando programmavo di eseguire”.
- La seconda prospettiva la definisco di collaborazione poiché la totalità dei bambini ha risposto che gli è piaciuto molto collaborare e ricevere aiuto da altri compagni.

Questo metodo ha avuto esiti favorevoli perché ha rafforzato l'impegno di ciascuno a superare le difficoltà che di volta in volta rendevano l'esercizio “difficile” tanto quanto lo sono i giochi ai quali i nostri bambini sono abituati, questo per fare in modo che l'attenzione e la concentrazione non calassero; inoltre ha alimentato la capacità collaborativa del gruppo classe in una situazione di gioco altamente accattivante.

6. *La valutazione del pensiero computazionale*

Dopo aver analizzato il pensiero computazionale mediante i concetti, le pratiche e le prospettive, si è cercato di effettuare un'analisi valutativa sul processo di insegnamento della pratica computazionale attraverso alcuni indicatori della trasposizione didattica. Riporto di seguito due episodi della lezione tenuta dal docente in aula tecnologica.

1. I: Me ciao gatto spiegalo
2. A: Io come faccio a scrivere?
3. I: Nicolò, hai risolto?... Ciao gatto... Martina lì scrivi e lì lo porti
4. A: Maestra, io ho capito come bisogna fare la storia... Bisogna fare che se vuoi prendere un altro personaggio devi... fare di nuovo.
5. I: Spiega spiega
6. A: O mostragli come fare oppure puoi scrivere nascondi
7. I: Ok... se lo vuoi far sparire ovviamente
8. A: Andrea era troppo facile.
9. A2: Dove devo metterlo?
10. A1: Devi metterlo là. È facilissimo.
11. I: Pierpaolo, che storia stai facendo?
12. A: Sto ancora
13. I: Quanti ne stai mettendo? È una partita di pallone?
14. A: No
15. I: Hai messo lo sfondo con un campo di calcio?
16. A: No. Ah! devo mettere uno sfondo allora!
17. I: Ma Chiara glielo devi anche spiegare altrimenti lui non capisce e non lo saprà fare da solo. Spiegagli come si fa.
18. A: Devi mettere...
19. I: Dove? Ma perché devi cliccare Gaetano! Clicca dentro... prima avrai cliccato senza accorgertene... dai Gabri scervellati!!
20. A: Il gatto arancione... dai Marti devi aggiungere una cosa.
21. I: Ricomincia perché probabilmente Chiara non hai eseguito bene la consegna. Hai letto bene la consegna? Cosa dice?
22. A: Dice... fai arrivare il cane fino al gatto, fai dire ciao al gatto quando il cane arriva là
23. I: Quindi esegui, prova a fare esegui, Clicca su esegui, ricomincia... il cane non è arrivato al gatto ma tu hai spostato il gatto, poi fai dire ciao al gatto quando il cane arriva... stai attenta. Tu hai fatto prima quando arriva il cane, invece prima devi fare arrivare il cane. Dopo... il blocco

prima dice quando arriva il cane e il cane, vedi. C'è qualcosa che non va. Allora... elimina quei blocchi e ricomincia. Leggi bene la successione delle azioni.

24. A: Fai arrivare il cane fino al gatto e fai dire ciao al gatto quando il cane arriva là.

25. I: e Vai! Nicolò che cosa hai agli occhi. Hai dormito bene?

26. A: Ma questo lo devo mettere in mezzo, maestra?

27. I: E penso proprio di sì. Riprova, c'è qualcosa che non va. A te come va Gaetano? Gabri ci stai riuscendo?

28. A:...

29. I: Bravo!

In questo episodio che qui viene riportato, l'insegnante apre e chiude l'episodio. L'obiettivo del docente è la creazione della storia mediante la programmazione. Nelle battute dalla numero 11 alla numero 14 si evidenzia l'invito del docente alla verbalizzazione della storia incoraggiando l'alunno al ragionamento. Si evince la logica valoriale basata sulla ragione.

Dalle prime battute si evidenzia l'impostazione della lezione basata sui porsi le domande. Alla battuta 5 e alla battuta 17, c'è una chiara evidenza dell'invito della docente alla collaborazione e alla partecipazione di tutti gli allievi ma anche ad un implicito riferimento alla promozione della zona di sviluppo prossimale.

Ciò dimostra una tendenza verso valori democratici e partecipativi. Dalla battuta 19 alla battuta 21 l'insegnante funge da guida, finché alla battuta 23 esplicita le direzioni da seguire mettendo in evidenza la domanda della consegna. In questo episodio si evidenzia molto chiaramente la logica didattica dell'insegnante che guida seppur in maniera molto marginale verso la risoluzione del problema ponendo domande indirizzanti verso il suo obiettivo. La logica dell'apprendimento si basa sulla costruzione della conoscenza ed anche sullo sviluppo del pensiero logico e computazionale.

7. Conclusioni

Pochi hanno mostrato interesse incostante per l'alternanza di momenti "facili" ed altri "difficili".

La maggior parte ha mostrato interesse, curiosità e partecipazione ed ha contribuito alla risoluzione di alcuni quesiti di tipo tecnico da sciogliere.

In generale ciascuno in opera davanti al proprio pc è riuscito ad acquisire i primi strumenti di pensiero computazionale per programmare una pic-

cola storia che non solo doveva rispondere ai criteri della consegna, ossia far muovere il cane e il gatto per far dire qualcosa, ma doveva dare un senso compiuto e piacevole alla storia stessa utilizzando i pochi elementi a disposizione in piattaforma, in particolare sfondo e suono.

Gli alunni hanno compreso che solo se il codice (cioè la successione di comandi) è breve, semplice e chiara l'azione sarà immediata.

Alla classe è piaciuto molto il risultato finale soprattutto per la sorpresa di aver costruito personalmente un piccolo programma. Gli inconvenienti sono stati tanti, soprattutto quello legato alla sottomissione del prodotto al concorso.

In ogni caso la sperimentazione ha sortito una ricaduta positiva lasciando negli alunni la curiosità di continuare il percorso il prossimo anno scolastico.

Didatticamente parlando si ritiene che si possano fare diverse osservazioni al riguardo: il pensiero computazionale è un processo mentale per la risoluzione di problemi costituito dalla combinazione di metodi caratteristici e strumenti intellettuali, entrambi di valore generale. Questa generalità è il motivo principale del perché insegnare *coding* (Wing, 2006, pp. 33-35).

I metodi caratteristici includono:

- analizzare e organizzare i dati del problema in base a criteri logici;
- rappresentare i dati del problema tramite opportune astrazioni;
- formulare il problema in un formato che ci permetta di usare un “sistema di calcolo” (nel senso più ampio del termine, ovvero una macchina, un essere umano, o una rete di umani e macchine) per risolverlo;
- automatizzare la risoluzione del problema definendo una soluzione algoritmica, consistente in una sequenza accuratamente descritta di passi, ognuno dei quali appartenente ad un catalogo ben definito di operazioni di base;
- identificare, analizzare, implementare e verificare le possibili soluzioni con un’efficace ed efficiente combinazione di passi e risorse (avendo come obiettivo la ricerca della soluzione migliore secondo tali criteri);
- generalizzare il processo di risoluzione del problema per poterlo trasferire ad un ampio spettro di altri problemi.

Questi metodi sono importanti per tutti, non solo perché sono direttamente applicati nei calcolatori (computer), nelle reti di comunicazione, nei sistemi e nelle applicazioni software ma perché sono strumenti concettuali per affrontare molti tipi di problemi in diverse discipline.

Gli strumenti intellettuali includono:

- confidenza nel trattare la complessità;

- ostinazione nel lavorare con problemi “difficili”;
- tolleranza verso l'ambiguità (da riconciliare con il necessario rigore che assicuri la correttezza della soluzione);
- abilità nel trattare con problemi definiti in modo incompleto;
- abilità nel trattare con aspetti sia umani che tecnologici, in quanto la dimensione umana (definizione dei requisiti, interfacce utente, formazione ecc.) è essenziale per il successo di qualunque sistema informatico;
- capacità di comunicare e lavorare con gli altri per il raggiungimento di una meta comune o di una soluzione condivisa.

Bibliografia

- Altet, M. (2003). *La ricerca sulle pratiche di insegnamento in Francia*. Brescia: Editrice La Scuola.
- Barr, D., Harrison, J. & Conery, L. (2011) Computational thinking: A digital age skill for everyone. In *Learning & Leading with Technology*, 38(6), pp. 20-52.
- Ben-Ari, M. (2001). Constructivism in computer science education. In *Journal of Computers in Mathematics and Science Teaching*, 20(1), pp. 45-73.
- Bichi, R., (2002). *L'intervista biografica. Una proposta metodologica*. Milano: Vita e Pensiero.
- Brennan, K., & Resnick, M., (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. In *Paper presented at annual American Educational Research Association meeting*, Vancouver, BC, Canada.
- Cuny, J., Snyder, L., & Wing, J. M., (2010). Demystifying computational thinking for non-computer scientists. *Unpublished manuscript in progress, referenced in <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>*
- Damiano, E. (2007). *L'insegnante etico. Saggio sull'insegnamento come professione morale*. Assisi: Cittadella Editrice.
- Develay, M. (1995). *Savoir scolaires et didactique des disciplines*. Paris: ESF.
- Fenstermacher, G. (1986). Philosophy of research on teaching. Three aspect. In Wittrock, M. C. (Ed.), *Handbook of research on teaching*, 3, New York: Macmillan.
- Greening, T. (2000). Emerging Constructivist Forces in Computer Science Education: Shaping a New Future? In *Computer science education in the 21st century*, pp. 47-80. Springer.
- Kafai, Y. B. (2006). Playing and Making Games for Learning: Instructionist and Constructionist Perspectives for Game Studies. In *Games and Culture*, 1(1), pp. 36-40.
- Kali, Y., & Linn, M. C. (2009). Designing Effective Visualizations for Elementary School Science. In *Elementary School Journal*, 109(5), pp. 181-198.
- Kolodner, J. L., et al. (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by Design into Practice. In *Journal of the Learning Sciences*, 12(4), pp. 495-548.
- Linn, M. C., Lee, H. S., Tinker, R., Husic, F. & Chiu J. L. (2006). Teaching and Assessing Knowledge Integration in Science. In *Science*, 313, pp. 1049-1050.
- Papert, S. (1975). Teaching Children Thinking. In *Journal of Structural Language*, 4, pp. 219-29.
- Papert, S. (1981). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.
- Puntambekar, S., & Kolodner, J. L. (2005). Toward Implementing Distributed

- Scaffolding: Helping Students Learn Science by Design. In *Journal of Research in Science Teaching*, 42 (2), pp. 185-217.
- Kafai, Y.B., & Ching, C.C. (2001). Affordances of Collaborative Software Design Planning for Elementary Students' Science Talk. In *Journal of the Learning Sciences*, 10(3), pp. 323-363.
- Shulman, S. L. (1986). Those who understand: Knowledge Growth in teaching. In *Educational Researcher*, Vol. 15, n. 1.
- Wilensky, U., & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World. In *Journal of Science Education and Technology*, 8(1), pp. 3-19.
- Wilensky, U. (2001). Modeling Nature's Emergent Patterns with NetLogo. In *Proceedings of the Eurologo 2001 Conference*, Linz, Austria.
- Wilensky, U., & Reisman, K. (2006). Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories – an Embodied Modeling Approach. In *Cognition and Instruction*, 24(2), pp. 171-209.
- Wing, J. M. (2006). Computational Thinking. In *Communications of the ACM*, 49(3), pp. 33-35.

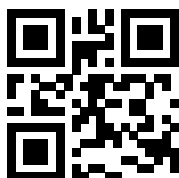
Il volume raccoglie gli atti, vale a dire le riflessioni, le relazioni e le comunicazioni che hanno percorso il convegno SIREM “L’educazione digitale. Modelli pedagogici e pratiche didattiche per la formazione iniziale e in servizio degli insegnanti”, che si è tenuto a Napoli nel marzo 2016.

Il convegno ha rappresentato una occasione importante in cui studiosi, ricercatori e insegnanti, mossi da diversi interessi e specializzati in svariati campi, si sono incontrati per dialogare, leggere, costruire percorsi didattici digitali che diano senso alla relazione educativa, quella fra insegnante e alunno, messa alla prova e resa complessa dall’avvento delle tecnologie digitali. Talvolta il digitale viene interpretato e accolto come un ospite inatteso, persino inquieto, che disturba la quotidianità scolastica. Talvolta, invece, esso viene percepito come la panacea, la soluzione dei mali e dei limiti scolastici.

La SIREM, Società Italiana di Ricerca sull’Educazione Mediale, intende superare i luoghi comuni sulle tecnologie digitali per affrontarli in modo deciso, realistico e, soprattutto, scientifico al fine di leggere la realtà, poter operare e fornire supporto a insegnanti ed educatori.

Pierpaolo Limone è professore ordinario di Pedagogia Sperimentale presso il Dipartimento di Studi Umanistici dell’Università di Foggia. La sua ricerca si sviluppa prevalentemente nell’ambito dell’educazione mediale, dell’e-learning e dei processi di assicurazione di qualità nei sistemi di istruzione. Tra le pubblicazioni più recenti, ricordiamo *Ambienti di apprendimento e progettazione didattica. Proposte per un sistema educativo transmediale* (Roma 2012); per Progedit, *Valutare l’apprendimento on-line. Esperienze di formazione continua dopo la laurea* (Bari 2012).

Davide Parmigiani è professore associato di Didattica Generale presso il Dipartimento di Scienze della Formazione dell’Università di Genova. Il suo ambito di ricerca ruota intorno alla progettazione, monitoraggio e valutazione di ambienti di apprendimento scolastici e sul territorio. Le sue ultime pubblicazioni sono: *Tra il dire e il fare. Come si decide a scuola: dalla riunione alla classe* (Milano 2010); *L’aula scolastica*, vol. I: *Come si insegna, come si impara* (Milano 2014); vol. II: *Come imparano gli insegnanti* (Milano 2016).



Euro 10,00

ISBN 978-88-6194-340-7



9 788861 943407