

# Tecnologie e pensiero computazionale: proposte per il curriculum di tecnologia nella scuola primaria

Cinzia Ferranti<sup>1</sup>, Vincenzo Agosto<sup>2</sup>  
FISPPA – Università di Padova

<sup>1</sup>Piazza Capitaniato, 3 - 35139 Padova PD  
[cinzia.ferranti@unipd.it](mailto:cinzia.ferranti@unipd.it)

Dipartimento di Scienze Statistiche - Università di Padova  
Via C. Battisti, 241 - 35121 Padova  
Email - [vincenzo.agosto@stat.unipd.it](mailto:vincenzo.agosto@stat.unipd.it)

*L'articolo propone una riflessione sul valore delle tecnologie per progettare attività volte a sviluppare il pensiero computazionale all'interno di quella preziosa ora di tecnologia presente nella scuola primaria. Si tratta di considerazioni preliminari e del resoconto della fase iniziale di una ricerca quali-quantitativa volta a ricavare una mappatura delle esperienze didattiche nel Veneto. L'obiettivo finale è quello di proporre degli spunti alla progettazione di attività di sviluppo del pensiero computazionale durante i cinque anni della scuola primaria.*

## 1. Introduzione

Il recente Piano Nazionale per la Scuola Digitale (PNSD) fa esplicito riferimento, in più punti, al pensiero computazionale. Si tratta di una competenza che studenti di ogni età dovrebbero mostrare di possedere, quindi risulta basilare iniziare a consolidarla già a partire dalla scuola primaria. Il Piano nazionale propone idee concrete per sostenere tale didattica, innestandosi in una situazione generale in cui le competenze degli insegnanti, necessarie per proporre attività di sviluppo del pensiero computazionale, sono distribuite, nel nostro paese, a macchia di leopardo.

In base a queste premesse, l'articolo presenta alcune considerazioni relative ad una didattica che intende stimolare e proporre strumenti per il pensiero computazionale. In particolare si vedranno quali processi di pensiero sono coinvolti e come si possa, a partire dal ragionamento qualitativo, progettare laboratori che implicino l'uso di software o di linguaggi formali per studenti della scuola primaria. Tale riflessione rappresenta una delle fasi preliminari di una ricerca quali-quantitativa su scala regionale, che coinvolge insegnanti di tecnologia e animatori digitali del Veneto della scuola primaria.

L'indagine, nella sua interezza, intende comprendere e divulgare lo stato dell'arte delle pratiche di insegnamento del pensiero computazionale nella scuola primaria e intende anche chiarire in che misura sia possibile ripensarne la didattica superando la logica, ancora molto radicata, dei programmi, per

proporre una didattica per competenze. L'obiettivo della ricerca prevede anche di considerare alcuni requisiti per formare gli insegnanti tenendo conto che il curriculum dovrebbe integrare le intenzioni presenti nella normativa nazionale e nel Piano Nazionale Scuola Digitale, cercando di superare i limiti temporali di sperimentazioni brevi o occasionali per trasformarle in pratiche didattiche stabili e consolidate.

## **2. La ricerca: obiettivi e metodi**

La ricerca sulle pratiche didattiche legate al pensiero computazionale nel Veneto ha un'impianto quali-quantitativo. Si avvale di strumenti di rilevazione dei dati qualitativi di seguito descritti:

- interviste a testimoni privilegiati (insegnanti di tecnologia e animatori digitali della scuola primaria e alcuni esperti come formatori, docenti universitari);
- osservazione partecipante;
- analisi documentale (siti web di progetti educativi internazionali e nazionali, normativa nazionale).

Gli strumenti quantitativi sono un questionario destinato sempre ad animatori digitali e ad insegnanti di tecnologia.

La domanda fondamentale che ha stimolato l'avvio della ricerca è: "E' possibile mappare le pratiche di didattica del pensiero computazionale presenti nelle scuole primarie del Veneto?" Da questa domanda ne segue un'altra domanda: "Quali sono gli strumenti progettuali più adatti per un insegnante che voglia introdurre o incrementare attività legate al pensiero computazionale?" Tutta la ricerca cercherà di rispondere a queste domande articolandosi in 3 fasi:

1. Fase di ricerca della letteratura internazionale e nazionale, di risorse web e analisi documentale;
2. Fase rilevamento e analisi dati quali-quantitativi,
3. Fase di modellazione per formare alla progettazione di un curriculum verticale di tecnologia e per delineare attività di didattica del pensiero computazionale nella scuola primaria.

L'obiettivo finale è quello raccogliere testimonianze che permettano di avviare una discussione sulla didattica del pensiero computazionale in relazione a diverse tipologie di proposte, *unplugged* e *plugged*, nella scuola primaria, adottando un approccio per competenze strutturato e sistemico relativo ai cinque anni del primo ciclo.

## **3. Tecnologie e pensiero: un binomio efficace per una didattica di sviluppo delle capacità di pensiero**

La tecnologia in quanto strumento di azione per l'uomo, di risoluzione di problemi di varia natura e mezzo di soddisfazione dei bisogni che egli storicamente inventa e utilizza, potenzia le nostre capacità d'azione [Mc Luhan, 1990], chiedendone una diffusione e una formazione al suo uso.

Partendo proprio dalla relazione tra tecnologia e pensiero, sembra importante richiamare alcune funzioni attribuibili al pensiero, prima di introdurre quella forma specifica che viene denominata *pensiero computazionale*.

Il pensiero, se considerato concretamente, si lega ad alcuni processi e funzioni particolarmente importanti per la conoscenza e l'attività umana. Si tratta della funzione organizzativa, attentiva, anticipatoria, immaginativa, rappresentativa, pianificatoria, simulatoria per citarne solo alcune che, attraverso l'uso di appropriati segni, manipola contenuti come se fossero oggetti reali [Shepard, 1978; Kosslyn, 1995].

In questo senso riteniamo pertinente l'intuizione Vygotskijana che vede nel segno (linguaggio) e nello strumento (materiale o tecnologico) il passaggio obbligato di una mediazione tra un soggetto e un oggetto (obiettivo da raggiungere) nella sua teoria storico culturale dell'attività in cui l'"azione è mediata dall'artefatto e orientata sull'oggetto" [Vygotskij, 1978]. Sempre Vygotskij pose un accento particolare sulla funzione del linguaggio e dei segni come mezzi per internalizzare le conoscenze (passare dal mondo al pensiero) ed externalizzare (passare dal pensiero al mondo). Non è solo per richiamare le intuizioni di un grande pensatore del XX secolo che abbiamo citato Vygotskij, ma perchè riteniamo che questo richiamo permetta di illuminare ciò che ci accingiamo a dire, ovvero che le tecnologie didattiche acquistano il massimo del loro valore qualora vengano utilizzate e considerate **tecnologie del pensiero**. Se ci riferiamo ad alcuni contenuti del pensiero, possiamo considerare come i modelli, i diagrammi, i segni, i simboli, i linguaggi, i disegni, le mappe necessitano di strumenti e linguaggi per essere espressi e diffusi. Essi stessi possono a loro volta diventare strumenti cognitivi nelle situazione d'apprendimento e di insegnamento.

#### **4. Tecnologie e pensiero computazionale: caratteristiche di una forma specifica di pensiero**

Riducendo la complessità del tema, potremmo considerare il "pensiero" come un articolato processo invisibile o implicito legato a meccanismi materiali neurologici e strettamente connesso a numerosi sottoprocessi e operazioni mentali quali: la creazione di categorie, concetti, processi di ragionamento, l'attività di immaginazione, le inferenze, le rappresentazioni, la progettazione, e così via [Giroto e Legrenzi, 1999; Tversky e Kahneman, 1974].

L'aggettivo *computazionale* avrebbe la funzione di definire una **forma di pensiero specifica** che si basa su processi e operazioni simili a quelli utilizzati da un elaboratore di informazioni [Wing, 2006].

Si tratterebbe quindi di un pensiero che richiama una competenza chiave, fondamentale per programmare, che ha permesso la diffusione di conoscenze e artefatti provenienti dalla *computer science* e che ha esplicitato e resi concreti processi che possono considerarsi oramai a se stanti e, in parte, presenti in altre forme di pensiero, trasferibili ad altri contesti e discipline. La discussione attuale sul valore del pensiero computazione verte proprio su questa raggiunta

autonomia tanto da considerarsi un “*corpus* di conoscenze da approcciare già dai primi anni di scuola” [Lodi, 2014].

Come è noto va a Papert [1990; 1996] il merito di aver introdotto l'espressione *computational thinking* in quanto tipologia di pensiero logico matematico. Risulta comunque successiva la volontà di farne un programma educativo, da proporre nella scuola dell'obbligo, spostando l'applicazione del pensiero computazione da un ambito strettamente connesso alla *computer science* e alla programmazione ad uno meno specifico. In questa maniera viene considerato una competenza autonoma applicabile a molti contesti di conoscenza pertanto da acquisire sin dalla scuola primaria, considerandola la quarta abilità di base da acquisire alla fine della scuola primaria [Lodi, 2013].

L'espressione “pensiero computazionale” nasce quindi dal collegamento attuato tra le modalità procedurali tipiche della programmazione informatica e un pensiero che, se applicato alle attività e ai problemi di qualsiasi natura, richiede la capacità di avviare operazioni di scomposizione e di organizzazione procedurale delle informazioni per giungere ad una soluzione.

E' quindi la domanda chiave della Wing [2006] “What is computable?” che ha lanciato una sfida didattica a tutti coloro che vogliono cimentarsi con una progettazione disciplinare, interdisciplinare o metadisciplinare basata su questa forma di pensiero, cercando di esplicitare cosa e come si possa elaborare le informazioni “alla maniera” dei computer.

Un tentativo di dare maggiore significatività all'aggettivo *computazionale* potrebbe essere quello di richiamare una definizione operativa proveniente da diverse fonti. Il pensiero computazionale consente di formulazione problemi in modo che un computer o altri strumenti possano risolverli, permette di organizzare e analizzare dati in maniera logica, rappresentare i dati in maniera astratta utilizzando modelli e simulazioni; automatizzare le soluzioni attraverso algoritmi, analizzare e implementare soluzioni attraverso una combinazione efficiente di step e risorse, generalizzare e trasferire i processi e le soluzioni ad altre classi di problemi [Barr e Stephenson, 2011; CSTA]

Ritornando all'espressione *pensiero computazionale* la presenza del termine *pensiero* richiama costantemente *skill* di tipo cognitivo che diventano obiettivo per il suo stesso sviluppo. Il fatto di averlo denominato *pensiero computazionale* e non pratiche computazionali o modellazione computazionale o linguaggio computazionale attribuisce, a nostro avviso, valore ai processi cognitivi ad esso connessi e permette, dal punto di vista di una progettazione didattica, una grande varietà di strade percorribili.

Il pensiero computazione è strettamente connesso con il linguaggio computazionale, ma in base al grado di scuola saranno diversi gli approcci e le conoscenze relative ai codici coinvolte. Sicuramente resta in primo piano il fatto che si tratta di un approccio ai problemi, per questo motivo è importante partire dalla seguente domanda “quali aspetti sono “intrinseci” del pensiero computazionale?” [Lodi, 2013], lasciando spazio, soprattutto nella scuola primaria, anche a modalità didattiche che non necessitano dell'apprendimento di linguaggi di programmazione.

Un altro elemento importante per una didattica del pensiero computazionale è l'uso della metacognizione, che dovrebbe accompagnare strategie didattiche basate maggiormente sul gioco e sulla scoperta, per consolidare gli apprendimenti [Borkowski & Muthukrishna, 2011] e incentivare il trasferimento degli stessi ad altri problemi o contesti [Bransford et al. 2000].

#### **4. L'integrazione del pensiero computazionale nel curricolo di tecnologia della scuola primaria**

Nel panorama internazionale sono molteplici le esperienze didattiche e la produzione scientifica relativa alla didattica del pensiero computazionale nella scuola primaria [Seiter e Foreman, 2013; Hill, 2014; Shelton, 2014, Cole, 2015; Duncan e Bell, 2015]. Per la struttura disciplinare presente nella scuola italiana, sicuramente la proposta di curricoli orizzontali potrebbe essere particolarmente adatta a sostenere lo sviluppo del pensiero computazionale. E' comunque relativamente all'insegnamento della "tecnologia" nella scuola primaria che può rivelarsi lo strumento più adatto. Molti insegnanti di scuola primaria stanno da alcuni anni proponendo attività che in varia misura interessano lo sviluppo del pensiero computazionale. Dalla fase qualitativa della ricerca ancora in corso (interviste a insegnanti di tecnologia come testimoni privilegiati) emerge che la maggior parte di essi si basano su esperienze pratiche di laboratori di coding, uso di Scratch o robotica. Tuttavia emerge anche che ci sono delle difficoltà a recepire le indicazioni della normativa nazionale e a progettare un curricolo verticale di tecnologia che tenga conto di essa, delle esperienze diffuse e di obiettivi di organicità nei cinque anni di scuola. Ci sembra interessante riportare alcune loro considerazioni esplicitate durante le interviste:

- Le proposte di didattica del pensiero computazionale nel Veneto sono "distribuite a macchia di leopardo", anche all'interno di un medesimo istituto comprensivo. La mancanza di riferimenti comuni e precisi dà la possibilità di interpretare le indicazioni per il curricolo del 2012 in maniera molto libera, soprattutto considerando che non si fa esplicito riferimento al pensiero computazionale e tale libertà lascia disorientati alcuni insegnanti;
- Ci sono esperienze sul campo di matura ristrutturazione del curricolo che possono fungere da *best practice* e che, se trasferite, darebbero un ottimo impulso ad una sistemica progettazione di attività di sviluppo del pensiero computazionale;
- Si percepisce l'esigenza di una formazione seria su questa competenza che guardi contemporaneamente ad una progressiva evoluzione verso il passaggio alla scuola secondaria di primo grado;
- La tecnologia nella scuola primaria, oltre a essere disciplina a se stante, chiede una costante integrazione con le altre discipline, quindi anche sul fronte del pensiero computazionale, applicandola ad altri ambiti e alla vita quotidiana.

- C'è la necessità di proporre attività che siano basate sul gioco e su una didattica per scoperta, in quanto ritenute strategie particolarmente adatte ad una didattica del pensiero computazionale.
- Non tutti gli insegnanti hanno una propensione alla sperimentazione autonoma, per cui sarebbe opportuno fornire loro delle risorse già strutturate, ma anche metodologie adatte.
- Molti insegnanti ritengono che problemi concreti legati al laboratorio di informatica limitino le potenzialità di una didattica della tecnologia (problemi legati al numero ridotto di postazioni, alla turnazione, al funzionamento e mantenimento delle macchine, ecc.);
- E' ancora diffusa la scelta di sacrificare l'ora di tecnologia per fare altro o rifarsi ai vecchi programmi di informatica per la primaria proponendo esclusivamente attività di videoscrittura (alcuni insegnanti portano in aula informatica i bambini per imparare a usare la tastiera per diverse settimane o fanno compilare un file word lungo l'arco di tempo di qualche mese);
- Spesso ostacoli epistemologici e abitudini consolidate creano resistenze alla progettazione e all'introduzione di nuove pratiche legate al pensiero computazionale;
- C'è bisogno di delineare passaggi incrementali e progressivi nelle attività proposte mano a mano che si passa dalla prima alla quinta classe.

Queste sono alcune delle considerazioni che gli insegnanti hanno fatto durante le prime interviste e mettono sul piatto della bilancia questioni legate ad una scuola e ad un paese che si sta ponendo degli obiettivi che guardano al futuro, ma che ha delle situazioni che frenano il raggiungimento degli stessi. Alcuni insegnanti fanno esplicita richiesta di strumenti per una nuova progettazione. Molti si sentono frenati dalla mancanza di nuove dotazioni, mostrando di non aver avuto accesso alla documentazione di progetti in cui il pensiero computazionale si insegna in modalità *unplugged* (Computer Science Unplugged, Debras, Code.org per citarne solo alcuni) o con l'uso di *tablet*, svincolandosi dalle dotazioni di un'aula di informatica. In base allo stato di avanzamento attuale della ricerca è ancora prematuro proporre delle soluzioni articolate che provengano dalla mappatura delle pratiche, ma a questo punto vorremmo provare a fare un'esperimento mentale, una ipotesi che deriva anche dall'analisi delle osservazioni fatte dagli insegnanti finora intervistati. Supponiamo che un insegnante di tecnologia volesse progettare un curriculum verticale in cui cercare lo spazio per l'apprendimento di processi di pensiero computazionale, procedendo nella seguente maniera:

- 1) Dapprima effettua una analisi e una elaborazione delle indicazioni provenienti dai principali documenti normativi italiani come le indicazioni nazionali sul curriculum (2012), la legge della Buona scuola (2014) e il PNSD (2015);

- 2) Poi ricerca di esperienze mature di didattica del pensiero computazione (risorse in rete, comunità di pratica, formazione e aggiornamento professionale);
- 3) Infine intende costruirsi un set di strumenti per una progettazione autonoma del proprio curricolo.

Al punto 1) l'ipotetico insegnante potrebbe voler incrociare gli obiettivi di apprendimento alla fine del ciclo presenti nelle indicazioni nazionali del 2012 con una delle definizioni operative del pensiero computazionale per individuare delle strade praticabili di progettazione autonoma. Potrebbe anche porsi domande molto concrete come: quanto tempo intendo dedicare al pensiero computazionale nell'arco dell'anno? Come riesco a strutturare un percorso durante tutto l'arco dei cinque anni che mi permetta di aggiungere gli obiettivi di apprendimento in maniera incrementale? Se proseguire il nostro esperimento cerchiamo di organizzare in una tabella l'incrocio tra gli obiettivi d'apprendimento e l'acquisizione delle competenze base legate al pensiero computazionale (Tab.1)

Vedere e osservare	Processi chiave del PC	Saper fare con il PC
Eseguire semplici misurazioni e rilievi fotografici sull'ambiente scolastico o sulla propria abitazione.	?	
Leggere e ricavare informazioni utili da guide d'uso o istruzioni di montaggio.	Procedure e algoritmi	Decodificare istruzioni e creare istruzioni con linguaggio naturale e/o con linguaggio visuale a blocchi
Impiegare alcune regole del disegno tecnico per rappresentare semplici oggetti.	Rappresentazione di dati; Riconoscimento di modelli e astrazione	Riconoscere caratteristiche degli oggetti e saper astrarne la forma, creare sequenze di passi per eseguire il disegno o utilizzare ambienti per creare disegni
Effettuare prove ed esperienze sulle proprietà dei materiali più comuni.	Riconoscimento di modelli e astrazione	Attuare processi di riconoscimento di pattern
Riconoscere e documentare le funzioni principali di una nuova applicazione informatica.	Procedure e algoritmi	Analisi funzioni software procedure e meccanismi automatizzazione, rilevare la presenza di processi ricorsivi
Rappresentare i dati dell'osservazione attraverso tabelle, mappe, diagrammi, disegni, testi.	Rappresentazione di dati; Scomposizione di un	Rappresentare anche attraverso l'uso di strumenti tecnologici dopo aver

	problema	riconosciuto modelli, compiuto processi di astrazione, mappato un fenomeno complesso scomponendolo
Prevedere e immaginare		
Effettuare stime approssimative su pesi o misure di oggetti dell'ambiente scolastico.	?	
Prevedere le conseguenze di decisioni o comportamenti personali o relative alla propria classe.	Scomposizione di un problema	Uso di diagrammi di flusso per rappresentare sequenze di azioni e conseguenze; trasformare decisioni o comportamenti in storie: storytelling e coding
Riconoscere i difetti di un oggetto e immaginarne possibili miglioramenti.	Procedure e algoritmi (debugging)	Analisi metacognitiva di errori con la programmazione visuale a blocchi
Pianificare la fabbricazione di un semplice oggetto elencando gli strumenti e i materiali necessari.	Scomposizione di un problema; Procedure e algoritmi	Immaginare sequenze di azioni semplici per costruire l'oggetto e scrivere istruzioni dettagliate (coding o utilizzo di software per realizzare semplici App)
Organizzare una gita o una visita ad un museo usando internet per reperire notizie e informazioni.	Scomposizione di un problema; Procedure e algoritmi	Creare una connessione tra information literacy e pensiero computazionale; creare app per gestire le informazioni
Intervenire e trasformare		
Smontare semplici oggetti e meccanismi, apparecchiature obsolete o altri dispositivi comuni.	Scomposizione di un problema	Usare concetti modulari (tipo lego) per riconoscere elementi simili di un oggetto
Utilizzare semplici procedure per la selezione, la preparazione e la presentazione degli alimenti.	Procedure e algoritmi	Decodificare istruzioni e creare istruzioni con linguaggio naturale per seguire ricette
Eeguire interventi di decorazione, riparazione e manutenzione sul proprio corredo scolastico.	?	
Realizzare un oggetto in cartoncino descrivendo e documentando la sequenza delle operazioni.	Procedure e algoritmi	Decodificare istruzioni e creare istruzioni con linguaggio naturale



Cercare, selezionare, scaricare e installare sul computer un comune programma di utilità.	Scomposizione di un problema; Procedure e algoritmi	Scomporre un processo di ricerca e selezione nel web e seguire istruzioni per scaricare e installare
---	--	--

Tab 1 – Obiettivi di apprendimento e processi del pensiero computazionale

La tabella mostra come si possa riconoscere la presenza implicita di processi legati al pensiero computazionale anche se molto probabilmente non era nelle intenzioni esplicite delle indicazioni del 2012. Potremmo osservare che alcuni processi sono maggiormente presenti, si tratta ad esempio della “scomposizione di un problema”, del “riconoscimento di modelli e astrazione”, della individuazione di “procedure e algoritmi” e della “rappresentazione di dati”. Altri invece come la parallelizzazione, la sincronizzazione o i processi di debugging non siamo riusciti a riscontrarli.

Se il nostro ipotetico insegnante provasse ora ad integrare le indicazioni presenti ne “La buona scuola” (vedi art. 1 punto 7 lettera h) poco riuscirebbe ad aggiungere alla sua capacità di progettare un curricolo verticale di tecnologia che dedichi uno spazio consistente al pensiero computazionale. Se infine aggiungiamo il PNSD la situazione si presenta in vivace evoluzione; oltre alla possibilità di aderire all’iniziativa “Programma il futuro”, riusciamo a trovare un obiettivo temporale relativo ad un “corpus di 10 ore annuali di logica e pensiero computazionale” esteso a tutti gli studenti.

Dopo questa prova, che lascia delle questioni aperte, sempre il nostro ipotetico insegnante potrebbe seguire il seguente iter: cercare tra le pratiche esistenti un certo numero di attività già strutturate e inserirle nella sua programmazione, ovvero passare al punto 2). Ecco che potrebbe cadere nella “trappola” della collezione di ricette [Shelton, 2014] senza raggiungere quella capacità di programmazione autonoma che darebbe strutturazione coerente e valore situazionale alla sua progettazione.

L’esperienza mentale appena descritto, mette in luce alcune questioni che lasciamo aperte:

- E’ valido un procedimento deduttivo che mira a raggiungere un livello più concreto di progettazione cercando di incrociare obiettivi d’apprendimento e processi come abbiamo fatto?
- Con quali criteri un insegnante riesce a scegliere quanta parte del tempo dedica al pensiero computazionale?
- E’ possibile raggiungere in futuro un livello di ricerche didattiche e di risorse progettuali strutturate per la scuola primaria?

## 5. Conclusioni

La nostra ricerca sta facendo emergere molte questioni interessanti legate al contesto scolastico nazionale, alla normativa di riferimento e al valore che sta assumendo il pensiero computazionale nella scuola. Essa pone domande anche di tipo teorico sulla natura di tale pensiero e sugli strumenti linguistici e tecnologici più adatti per essere insegnato. E’ emerso dalle interviste il valore

delle esperienze locali, portate avanti spesso con spirito pionieristico. Tuttavia permane l'esistenza di molti ostacoli epistemologici, legati a difficoltà materiali ma anche alla difficoltà di progettare curricoli verticali di tecnologia per mancanza di riferimenti o strumenti progettuali.

Si evidenzia quindi la necessità di una formazione che proponga non tanto ricette da seguire, ma strategie per progettare creativamente percorsi organici e sistemici nell'arco dei cinque anni. I dati finora rilevati non permettono di fornire ancora una mappatura delle pratiche didattiche o delle soluzioni articolate, ma certamente di porre l'attenzione sulla trasformazione in atto partendo da questo punto fermo: gli insegnanti hanno la necessità di orientarsi meglio e di mettere a punto strumenti di progettazione, diremo basati essi stessi sul pensiero computazionale, che permettano un adattamento a linguaggi e tecnologie in continua evoluzione. Abbiamo quindi per ora intravisto alcune direzioni che necessitano in futuro di venire sondate meglio e sperimentate.

## Bibliografia

[1] Barr V., Stephenson C., Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *Inroads*, 2, 1, 2011, 48- 54.

[2] Borkowski J. G., Muthukrishna N., *Didattica metacognitiva. Come insegnare strategie efficaci di apprendimento*, Edizioni Erickson, Trento, 2011.

[3] Bransford, J. D., Brown, A. L., & Cocking, R. R., *How people learn*, National Academy Press, Washington DC, 2000.

[4] Cole, E. C., On Pre-requisite Skills for Universal Computational Thinking Education. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*, 253-254, ACM, 2015.

[5] Computer Science Teachers Association. *Computational Thinking*. 2012. <http://csta.acm.org/Curriculum/sub/CompThinking.html>

[6] Computer Science Unplugged - <http://csunplugged.org>

[7] Debras - <http://www.bebras.org/?q=goodtask>

[8] Code - <http://www.code.org>

[9] Girotto V., Legrenzi P., *Psicologia del Pensiero*, Bologna, Il Mulino, 1999.

[10] CSTA - <https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>

[11] Duncan, C., & Bell, T., A Pilot Computer Science and Programming Course for Primary School Students. In *Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ*, 39-48, ACM, 2015.

[12] Hill, C., Computational thinking curriculum development for upper elementary school classes. In *Proceedings of the tenth annual conference on International computing education research*, 151-152, ACM, 2014.

[13] Kosslyn, S. M. *Mental imagery. Visual cognition: An invitation to cognitive science*, 2, 267-296, 1995.

- [14] Lodi, M., Imparare il pensiero computazionale imparare a programmare, tesi di Laurea, [http://www.olimpiadiproblemsolving.com/documenti/lodi\\_michael\\_tesi.pdf](http://www.olimpiadiproblemsolving.com/documenti/lodi_michael_tesi.pdf)
- [15] Papert S., An Exploration in the Space of Mathematics Educations, International Journal of Computers for Mathematical Learning, 1, 1, 95-123, 1996.
- [16] Programma il futuro - <http://www.programmailfuturo.it/>
- [17] Seiter, L., Foreman, B., Modeling the learning progressions of computational thinking of primary grade students. In Proceedings of the ninth annual international ACM conference on International computing education research, 59-66, ACM, 2013.
- [18] Shelton C., Beyond lesson recipes: first steps towards a repertoire for teaching primary computing, 2014.
- [19] Shepard, R. N., The mental image. American psychologist, 33, 2,125, 1978.
- [20] Turkle S., Papert S., Epistemological Pluralism and the Revaluation of the Concrete, SIGNS: Journal of Women in Culture and Society, 16, 1, 1990.
- [21] Teaching London Computing - <http://teachinglondoncomputing.org/contact/>
- [22] Tversky, A., Kahneman, D., Judgment under uncertainty: Heuristics and biases. science, 185, 4157, 1124-1131, 1974.
- [23] Vygotskij L. S., Mind in society. The development of higher Psychological processes, edito da M. Cole et al, Cambridge, Mass., Harvard University Press, 1978, [Trad. it., Il processo cognitivo, , Boringhieri, Torino, 1980].
- [24] Vygotskij L.S., Thought and Language, Massachussets Institut of Technology, 1962 [Trad. it. Pensiero e linguaggio, Laterza, Bari]
- [25] Wing, J. M., Computational thinking, CACM 49, 3, 33-35, 2006.