

# Informatica e pensiero computazionale: una proposta costruttivista per gli insegnanti

A. Formica, G. Garbin, S. Patania

June 8, 2018

## 1 Presentazione della proposta didattica

Nell'ultimo decennio l'idea di diffondere la cultura informatica nelle scuole primarie e secondarie è diventata un *trend topic*. Sia le direttive europee che quelle nazionali sono concordi nel sottolineare l'esigenza di fornire una conoscenza informatica di base già a partire dalla scuola primaria (vedi PNSD). Spesso però, quando si parla di informatica, è facile cadere in un fraintendimento. Si tende, infatti, a identificare l'informatica con la capacità di utilizzare specifici applicativi software. Questa idea dell'informatica sterile e fuorviante è sicuramente inadatta a coglierne la vera natura e le reali potenzialità insite nel pensiero computazionale.

Il percorso didattico delineato nel seguito si propone di fornire ai docenti strumenti didattici e metodologici per far comprendere agli studenti cosa sia realmente l'informatica partendo dalla scoperta dei temi che compongono la sua stessa definizione. Possiamo definire l'informatica come la scienza che si occupa dell'elaborazione automatica dell'informazione. Ma cosa si intende per elaborazione, automazione e informazione? L'offerta didattica si compone di diversi workshop, ognuno dei quali si propone di fornire una chiave di lettura che avvicini i docenti e gli studenti alla comprensione di uno di questi concetti.

I laboratori proposti non sono innovativi solo riguardo alle tematiche proposte ma anche in relazione alla metodologia di somministrazione. Quest'ultima si basa sull'idea che la conoscenza, e quindi l'apprendimento, si costruisca mediante l'esperienza diretta e attraverso una sintesi tra le conoscenze già possedute e le nuove. Svilupperemo meglio questo aspetto nel paragrafo 3.

La proposta che andremo a delineare nel seguito, giustificandone forma e contenuto, viene presentata agli insegnanti delle scuole primarie e secondarie di primo e secondo grado. Per ogni attività sarà indicato il target al quale il laboratorio è rivolto.

Sebbene le attività siano pensate per insegnanti di materie informatiche o affini, i materiali che metteremo a disposizione sono fruibili da qualsiasi docente. Infatti, come vedremo, le competenze sviluppabili attraverso questa offerta sono fortemente interdisciplinari.

## **2 L'importanza dello sviluppo del pensiero computazionale**

Una volta convinti che l'informatica ha niente o poco a che vedere con l'utilizzo di specifici software, cerchiamo di capire quale contributo può dare alla formazione di uno studente della scuola primaria o secondaria.

Il contributo culturale più significativo che l'informatica offre è il cosiddetto *pensiero computazionale*, l'insieme dei processi mentali che mette in atto un informatico nella sua tipica attività di *problem solving*, ovvero pensare in modo algoritmico e a livelli multipli di astrazione. Si tratta di competenze trasversali e declinabili in tutti gli ambiti disciplinari: formulare i problemi in modo che possano essere risolti in maniera automatica da agenti autonomi, organizzare e analizzare logicamente le informazioni, rappresentarle attraverso modelli e astrazioni, automatizzare lo svolgimento di compiti tramite sequenze di passi ordinati, generalizzare e trasferire processi risolutivi a una grande varietà di situazioni diverse. Questa realizzazione implica un'approfondita conoscenza dei problemi.

Da questo appare chiaro come l'informatica racchiuda a sé un grande

potenziale. Pensare da informatici significa riflettere su ciò che si fa e su come lo si fa. Come dice Papert, informatico e pedagogista, “pensare a pensare rende ogni bambino un epistemologo”.

Il pensiero computazionale può essere riassunto con la domanda “Come faccio a far risolvere questo a un computer?”. Riuscire a rispondere a questa domanda implica tre competenze fondamentali:

- **Astrazione**, ovvero la capacità di generalizzazione a partire da specifiche istanze.
- **Automazione**, quindi definire delle operazioni ripetitive da delegare a un computer sulla base delle astrazioni compiute.
- **Analisi**, la pratica riflessiva sulle astrazioni effettuate.

In conclusione, appare chiaro come l’uso di astrazioni e modelli nonché l’analisi di problemi e artefatti costituiscano elementi imprescindibili del pensiero computazionale, ponendosi allo stesso tempo come strumenti efficaci e spendibili in qualsiasi ambito disciplinare.

### 3 Metodologia

La filosofia alla base dei laboratori proposti si collega alle teorie socio-costruttiviste. Secondo il paradigma costruttivista la conoscenza non è qualcosa che possa essere trasmessa attraverso la semplice esposizione di contenuti. La classica dinamica insegnante-discente prevede che il discente rivesta un ruolo di soggetto passivo incaricato solo di ricevere il messaggio della comunicazione. Il costruttivismo radicale, negando l’esistenza di una verità ontologica, considera unica la conoscenza che si sviluppa nel soggetto che apprende. È inutile quindi ricercare la rappresentazione di una realtà indipendente da trasferire in qualche modo allo studente, questa infatti semplicemente non esiste. Il reale si costituisce nella mente del discente incastonandosi nell’universo semantico preesistente. In questo senso, attraverso il rovesciamento del proprio ruolo da soggetto passivo ad attivo, il discente modificherà le proprie rappresentazioni mentali per adattarle alle nuove esperienze.

È importante che in questo percorso il docente stimoli lo studente a riflessioni metacognitive. La consapevolezza di come si sviluppi la propria conoscenza è un punto fondamentale. È possibile realizzare questo obiettivo alternando momenti di apprendimento attivo nel quale l'alunno sviluppa nuova conoscenza a momenti di riflessione su come la sua conoscenza stia cambiando.

Un secondo aspetto evidenziato da queste teorie riguarda l'importanza della componente dell'interazione sociale nella costruzione della conoscenza. Il confronto con gli altri implica la capacità di un'opportuna formulazione linguistica, costringendo i soggetti dell'interazione all'esplicitazione di passaggi logici non banali.

Altro punto tenuto in considerazione nella pianificazione delle attività è il concetto di apprendimento allosterico. Si pone forte enfasi sull'ambiente nel quale si svolge il processo di apprendimento attivo. La cura della preparazione del setting fa sì che i ragazzi si ritrovino in un ambiente che li guidi passo passo nello svolgimento dell'attività mettendogli a disposizione gradualmente gli strumenti necessari. I laboratori sono strutturati in modo che gli studenti debbano imparare (sperimentando) qualcosa di nuovo per passare allo step successivo.

Ciascun laboratorio si svolge seguendo un format ben preciso che consta di 4 fasi: clusterizzazione, attività algomotoria, attività al computer, ricapitolazione.

- 1 Prima di tutto, il conduttore spiega che l'attività non prevede una prova finale e che le domande che saranno poste non sono parte di un'interrogazione. Poi si discute interattivamente l'etimologia del termine "informatica" focalizzandosi su un aspetto specifico (a seconda dell'attività) e si chiede ai ragazzi una loro definizione di questo aspetto. Le definizioni dovranno essere scritte su dei post-it anonimi che verranno raccolti dall'insegnante e suddivisi per livello di affinità su di una lavagna in modo che siano visibili a tutti gli studenti.
- 2 La seconda fase è quella della didattica algomotoria. In questa fase i ragazzi si mettono in gioco in prima persona nella risoluzione di un problema attraverso l'esplorazione libera: mediante tentativi ed errori

modificano e ampliano la propria conoscenza costruendo un modello del problema.

- 3 In questa fase il problema modellato in maniera *unplugged* viene rivisto attraverso l'utilizzo di un software specifico. È importante che questa fase avvenga successivamente alle altre, così che si colga l'indipendenza tra informatica e utilizzo del computer.
- 4 Infine, il conduttore dell'attività riprende il concetto dal quale ha preso vita l'attività contestualizzando il lavoro svolto alla luce di quella tematica.

## 4 Indicazioni pratiche sull'utilizzo del materiale

Per ciascun laboratorio abbiamo disposto un kit di strumenti che guidi l'insegnante che voglia proporlo alla propria classe nella strutturazione dell'attività. Se lo schema sintetico è pensato per essere letto per primo, il resto dei materiali invece può essere consultato nell'ordine che si ritiene più opportuno.

Ciascun kit si compone nel modo seguente:

- **Schema sintetico.** Qui si troverà un prospetto dell'attività contenente una breve descrizione del tema, gli obiettivi formativi, i fruitori ideali dell'attività e utili informazioni su spazi e tempi necessari. È consigliabile consultare questa parte prima delle altre per farsi un'idea del laboratorio e della sua fattibilità in determinati contesti.
- **Spiegazione del tema informatico.** Si tratta di una panoramica generale del tema informatico affrontato. È importante che gli insegnanti abbiano chiara la tematica entro cui ci si muove sia per capire l'importanza da un punto di vista prettamente informatico dell'attività svolta sia per essere in grado di indirizzare correttamente il flusso di lavoro.

- **Spiegazione attività.** Il laboratorio è spiegato nel dettaglio in modo che l'insegnante abbia chiaro ogni passaggio. Niente è dato per scontato.
- **Materiali.** Sono elencati i materiali necessari e sono messe a disposizione eventuali schede di supporto agli alunni.
- **Errori, domande e soluzioni comuni.** Questo è uno strumento che può rivelarsi determinante nel guidare l'attività dell'insegnante. Avere già un'idea di quelle che sono le criticità comuni e trovare dei suggerimenti per superarle sarà un grande supporto per il docente.

## 5 Approfondimenti

Per chi fosse interessato al dibattito sull'insegnamento dell'informatica nelle scuole, forniamo i seguenti riferimenti:

M. Lodi, S. Martini, and E. Nardelli. Abbiamo davvero bisogno del pensiero computazionale? *Mondo digitale*, (72):1–15, Nov. 2017.

J. Hromkovič and R. Lacher. The computer science way of thinking in human history and consequences for the design of computer science curricula. In *Proceedings of ISSEP 2017*, pages 3–11. Springer, 2017.

V. Dagien e and S. Sentance. It's computational thinking! Bebras tasks in the curriculum. In *Proceedings of ISSEP 2016*, pages 28–39. Springer, 2016.

The Royal Society. Shut down or restart? The way forward for computing in UK schools., Jan. 2012.

O. Hazzan, T. Lapidot, and N. Ragonis. *Guide to Teaching Computer Science: An Activity-Based Approach*. Springer London, 2011.

Per quanto riguarda il pensiero computazione segnaliamo i seguenti articoli:

R. Marchignoli and M. Lodi. EAS e pensiero computazionale. La Scuola, 2016.

J. M. Wing. Computational thinking. Communications of the ACM, 49(3), 33-35, 2006.

Inoltre, a chi volesse maggiori informazioni sull'approccio costruttivista suggeriamo:

A. Carletti and A. Varani, editors. Didattica costruttivista. Dalle teorie alla pratica in classe. Erickson, 2005.

E. V. Glasersfeld. Idee costruttiviste. Riflessioni Sistemiche, (2):179-181, 2010.