

# “Se dovessi dare un voto a questo laboratorio, darei nove”

## Introduzione del Computational Thinking nella scuola secondaria di primo grado: risultati dell'esperienza

Ilenia Fronza, Nabil El Ioini, Andrea Janes,  
Alberto Sillitti, Giancarlo Succi, Luis Ricardo Corral Velazquez  
*Libera Università di Bolzano*  
*Piazza Domenicani, 3, 39100, Bolzano (Italia)*  
{Ilenia.Fronza, Nabil.ElIoini, Andrea.Janes,  
Alberto.Sillitti, Giancarlo.Succi, Luis.Corral}@unibz.it

*Spesso l'Informatica, a differenza di altre discipline (come la matematica o la filosofia), non è percepita come una disciplina scientifica ma solo come un utilizzo della tecnologia. L'educazione al pensiero algoritmico, o Computational Thinking, mira a far comprendere il modello computazionale sottostante. Questo articolo descrive la struttura e i risultati di un laboratorio extra-scolastico che ha coinvolto 25 alunni di due scuole secondarie di primo grado. I risultati dell'esperienza vengono discussi e, sulla base dei risultati dell'esperienza, vengono proposti possibili sviluppi e miglioramenti della struttura del progetto.*

### 1. Introduzione

L'Association for Computing Machinery (ACM) definisce l'Informatica come “lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione: la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione”. Basandosi su questa definizione, l'ACM fornisce un curriculum modello per l'insegnamento dell'Informatica a scuola [Tucker et al, 2003]; l'idea di base di questo curriculum è quella di non puntare né all'insegnamento della programmazione, né all'alfabetizzazione informatica. Piuttosto, il curriculum mira allo studio dei computer e dei processi algoritmici, inclusi i loro principi di base, del loro hardware e della progettazione del software, delle applicazioni, e del loro impatto sulla società [Barr e Stephenson, 2011].

Nonostante questo, spesso la didattica dell'Informatica si concentra sull'informatica di consumo, fornendo solamente nozioni sull'utilizzo dei computer [Bizzarri et al, 2011]. Uno degli obiettivi didattici più diffusi è infatti lo sviluppo di abilità operative, ossia il miglioramento delle capacità di utilizzo del computer. Sviluppare queste capacità, tuttavia, non significa apprendere l'Informatica; per apprendere l'Informatica, infatti, è necessario acquisire competenze concettuali e non solo il corretto utilizzo di un computer. Il software

---

DIDAMATICA 2014

e i sistemi operativi, infatti, evolvono molto velocemente e alcuni di loro hanno una vita media molto breve. Ne consegue che l'apprendimento di specifiche tecniche d'uso in campo digitale è completamente inutile, in quanto quelle tecniche potrebbero essere superate o dimenticate a breve [Penge, 2004].

A differenza di altre discipline (come la matematica, la fisica, o la filosofia), spesso l'informatica non è quindi percepita come una disciplina scientifica e, di conseguenza, vengono ignorati i suoi contributi scientifici e culturali [Leoni, 2011]. Emerge quindi la necessità di introdurre l'informatica come disciplina autonoma, sia per i suoi valori propri sia per le risorse da essa offerte, indispensabili per risolvere problemi di altre discipline. In questo contesto, si inserisce l'educazione al pensiero algoritmico, o Computational Thinking [Wing, 2008], inteso come comprensione del modello computazionale sottostante alla soluzione di un problema; in quest'ottica, non interessa tanto che l'alunno comprenda l'algoritmo risolutivo di un problema, quanto il modello mentale che ha portato alla soluzione per poter essere in grado di proporre una propria soluzione [Bizzarri et al, 2011], [Contarini, 2006].

Il Center for Applied Software Engineering (CASE) della Libera Università di Bolzano, collabora con i dirigenti scolastici della Provincia Autonoma di Bolzano per trasmettere i principi del Computational Thinking. Iniziando dallo studio del problema e dalla progettazione di una soluzione ottimizzata, ben strutturata e facilmente modificabile, gli studenti imparano a organizzare e decomporre problemi, astrarre modelli e soluzioni, riconoscere strutture e forme. Gli studenti apprendono che, prima di utilizzare un computer per risolvere automaticamente un problema, si deve con la propria mente definire il procedimento risolutivo (algoritmo) che il computer utilizzerà per fornire la soluzione e che la propria materia grigia è il componente fondamentale su cui si radica ogni futura elaborazione, materia grigia che si rafforza e raffina con lo studio delle materie tradizionali specialmente se affrontate con l'ottica del Computational Thinking. Questa consapevolezza, inoltre, contribuisce ad avvicinare alla scienza e all'ingegneria dell'informazione studenti che potrebbero altrimenti rimanere indifferenti alla programmazione.

Questo articolo descrive la struttura e i risultati di un progetto che ha coinvolto 25 alunni di due scuole secondarie di primo grado che hanno partecipato a due laboratori pomeridiani, in orario extra-scolastico. La sezione 2 introduce il Computational Thinking e le fasi da esercitare per stimolare questa competenza. Inoltre, viene descritta la struttura delle attività svolte nei laboratori; la sezione 3 discute i risultati dell'attività. La sezione 4 è dedicata alle conclusioni e agli eventuali sviluppi futuri.

## 2. Introduzione del Computational Thinking nelle scuole

Il progetto si proponeva come obiettivo principale quello di promuovere situazioni che favoriscano la crescita di abilità e competenze di Computational Thinking negli studenti della scuola secondaria di primo grado. L'obiettivo didattico da perseguire era quindi quello di introdurre e allenare il Computational Thinking, mediante esercizi (anche "con carta e penna") di difficoltà crescente [Casadei e Teolis, 2011].

“Se dovessi dare un voto a questo laboratorio, darei nove”

A questo scopo, il laboratorio è stato impostato in modo da esercitare con attività mirate le nove fasi descritte nella Tabella 1 [CSTA, 2011]. Il laboratorio è stato quindi organizzato come segue:

- fase introduttiva: cos'è l'Informatica? (1 ora);
- introduzione alla programmazione in Scratch [MIT Media Lab, 2009] ed esercizi (4 ore);
- costruzione di un'animazione: la catena alimentare (8 ore).

Ciascuna fase del progetto è descritta di seguito in questa sezione. In ogni fase, è stata attribuita particolare importanza al lavoro ordinato e alla documentazione.

**Tab.1 – Computational Thinking: definizione delle fasi.**

<b>Fase</b>	<b>Definizione</b>
<b>Raccolta dei dati</b>	Processo di raccolta delle informazioni necessarie
<b>Analisi dei dati</b>	Dare un significato ai dati, trovare dei modelli e trarre delle conclusioni
<b>Rappresentazione dei dati</b>	Rappresentare e organizzare i dati appropriatamente in grafici, parole, e immagini
<b>Scomposizione del problema</b>	Scomporre il problema in parti più piccole e gestibili
<b>Astrazione</b>	Ridurre la complessità per definire l'idea principale
<b>Algoritmi e procedure</b>	Serie ordinata di fasi per risolvere un problema o raggiungere uno scopo
<b>Automazione</b>	Usare computer per svolgere lavori ripetitivi e noiosi
<b>Simulazione</b>	Rappresentazione o modello di un processo. La simulazione include l'esecuzione di esperimenti usando modelli
<b>Parallelizzazione</b>	Organizzare le risorse in modo che lavorino contemporaneamente per raggiungere un obiettivo comune

## **2.1 Fase introduttiva: “Cos'è l'informatica?”**

Lo scopo di questa fase introduttiva era comprendere fino a che punto gli alunni partecipanti al progetto percepissero l'Informatica come disciplina scientifica. A questo scopo, ad ogni alunno è stato chiesto di scrivere la propria definizione di Informatica specificando, secondo la propria opinione, dove si possa applicare l'informatica e dove possa essere utile.

Il wordcloud nella Fig.1 è stato creato utilizzando i testi delle 25 risposte fornite dagli alunni. La dimensione di una parola dipende dal numero di volte in cui essa compare nelle risposte. Il wordcloud conferma la tendenza degli alunni ad associare l'informatica all'utilizzo (“usa”) di “computer” e “tecnologia”. La



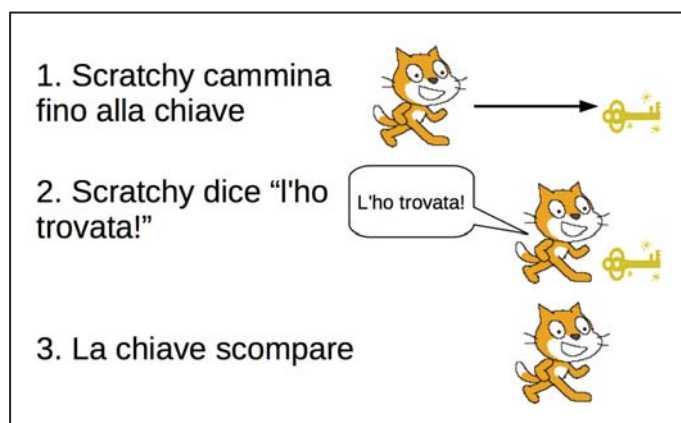


Fig.2 – Esercizio introduttivo: schema della soluzione.

### 2.3 Costruzione di un'animazione: la catena alimentare

L'obiettivo della terza fase del laboratorio era la creazione di una simulazione di diversi livelli della catena alimentare. Il tema dell'animazione è stato scelto in modo che tutti gli alunni avessero una buona conoscenza teorica delle dinamiche da rappresentare e potessero quindi concentrarsi sulla risoluzione del problema. In questo modo la programmazione è stata utilizzata per mostrare un concetto semplice e noto.

Durante questa fase sono state adottate le seguenti strategie:

- gli alunni hanno lavorato in gruppi di due;
- ogni gruppo ha scelto uno scenario (ad esempio, l'oceano o la savana);
- dapprima è stato creato un sistema originale semplice, che è stato in seguito reso più complesso;
- gli alunni sono stati invitati a riutilizzare e adattare i concetti imparati durante la seconda fase;
- ogni gruppo ha pensato e implementato diverse simulazioni (ad esempio, mancanza di erba o eccesso di carnivori).

La Tab. 2 descrive le attività svolte, mappandole con le fasi descritte nella Tab. 1.

Tab.2 – Costruzione di un'animazione: descrizione delle attività.

Fase	Durata (ore)	Attività
Raccolta dei dati	0.5	Scelta di uno scenario (ad esempio, campagna, oceano, savana) e raccolta di informazioni sulla catena alimentare (conoscenze preliminari e libri di scienze)
Analisi dei dati	0.5	Discussione sulle caratteristiche degli attori.

Fase	Durata (ore)	Attività
<b>Rappresentazione dei dati</b>	0.5	Creazione di un diagramma delle relazioni nella catena alimentare. Discussione dei vari fattori che favoriscono l'equilibrio nell'ecosistema. 1. Cosa può rendere più complesso questo ecosistema? 2. Quali sono le condizioni che mettono in pericolo le specie nell'ecosistema?
<b>Scomposizione del problema</b>	0.5	Identificare i sotto-problemi
<b>Astrazione</b>	0.5	Ridurre la complessità, ad esempio identificando un rappresentante per ogni livello della catena
<b>Algoritmi e procedure</b>	0.5	Creazione di uno "schema della soluzione" (vedi Fig.2)
<b>Automazione</b>	4	Programmazione di una versione semplice della soluzione identificata
<b>Simulazione</b>	1	Implementazione della simulazione di diversi scenari. Ad esempio, cosa accade quando ci sono troppi carnivori?
<b>Parallelizzazione</b>	8*	I gruppi pianificano le attività trovando i compiti da svolgere in parallelo e i momenti di riunione

\* I gruppi hanno esercitato la parallelizzazione durante tutta la durata del progetto.

Al termine del progetto, ogni gruppo aveva terminato una propria versione della catena alimentare.

### 3. Discussione

La prima fase dell'attività svolta ha confermato che l'Informatica è concepita come utilizzo della tecnologia più che come scienza. Andrebbe quindi valutata la possibilità di dedicare più tempo alla fase introduttiva, facendo riflettere gli alunni sul ruolo dell'informatica nella loro vita e sulla possibilità di utilizzare l'informatica per risolvere i problemi di tutti i giorni.

La seconda fase, nella quale vengono introdotti il pensiero algoritmico e le basi della programmazione (in Scratch), potrebbe essere estesa per rinforzare queste competenze prima di passare al progetto della terza fase. In questa esperienza, si è puntato a far progettare la soluzione usando "carta e penna", per far comprendere agli alunni che la programmazione si semplifica notevolmente con una soluzione strutturata, chiara e ordinata. Gli alunni, però, si sono spesso dimostrati impazienti di utilizzare il computer; in alcuni casi quindi è stato necessario convincerli a soffermarsi sulla progettazione, che stavano volutamente trascurando e accelerando.

La scelta dell'argomento della terza fase, la catena alimentare, ha permesso di ridurre notevolmente i tempi necessari a raccogliere ed analizzare le informazioni per risolvere il problema. La scelta di un argomento meno noto richiederebbe tempi maggiori e metodi diversi per organizzare e rappresentare le proprie conoscenze su uno specifico argomento. In questo caso, andrebbero dedicate più ore alla terza fase svolgendo ad esempio attività di brainstorming e creando mappe mentali. Bisogna inoltre considerare che queste attività potrebbero richiedere più iterazioni per ottenere elaborati sempre più raffinati.

Al termine delle attività, abbiamo chiesto ai partecipanti di scrivere la loro opinione (vedi Tab.3). In particolare, in più commenti sono emersi tre punti (ad esempio, vedi Fig.3):

- ai partecipanti, cosa certamente non scontata, è piaciuto programmare;
- i partecipanti hanno compreso, e apprezzato, che una soluzione si può raffinare in più passi. In particolare, se il computer “non fa quello che ci eravamo immaginati”, probabilmente abbiamo fatto un errore e sarà sufficiente trovarlo e risolverlo per ottenere una soluzione migliore;
- il laboratorio è stato gradito, tanto da meritare un nove nel commento da cui è tratto il titolo di questo articolo.

**Tab.3 – Commenti dei partecipanti: principali aspetti positivi e suggerimenti.**

<b>Commenti</b>	<b>Esempi</b>
<b>Positivi</b>	<ul style="list-style-type: none"><li>• imparare a programmare</li><li>• cercare e migliorare una soluzione</li><li>• imparare ad usare scratch</li><li>• usare il computer</li><li>• lavorare ad un'idea “geniale”</li></ul>
<b>Suggerimenti</b>	<ul style="list-style-type: none"><li>• il tema dell'animazione andrebbe cambiato, evitando la catena alimentare</li><li>• il laboratorio dovrebbe essere più lungo</li></ul>

#### **4. Conclusioni**

In questo articolo abbiamo descritto la struttura e i risultati di un laboratorio extra-scolastico che ha coinvolto 25 alunni di due scuole secondarie di primo grado. L'obiettivo del laboratorio era l'educazione al pensiero algoritmico, o Computational Thinking; a questo scopo, le attività sono state programmate in modo da far comprendere il modello computazionale sottostante alla soluzione.

Questa esperienza ha ottenuto l'apprezzamento dei partecipanti, e ha fornito numerosi spunti per eventuali sviluppi futuri. Ad esempio, il monte ore dedicato al laboratorio andrebbe aumentato, in particolar modo se il laboratorio fosse incentrato su un tema non così noto come la catena alimentare. In questo caso, andrebbero introdotti momenti di brainstorming e creazione di mappe mentali per organizzare le idee e le conoscenze.

Questo laboratorio mi è piaciuto molto perché mi ha ~~aiutato~~ insegnato il corretto uso di ~~di~~ SCRATCH. La cosa che ~~mi~~ mi è piaciuta di più è stata capire perché il programma non faceva quello che mi ero immaginato, come la soluzione.

QUESTO LABORATORIO MI PIACIUTO MOLTO PERCHÉ ABBIAMO USATO MOLTO I COMPUTER. MI È PIACIUTO MOLTO ANCHE PROGRAMMARE.

Ho imparato nuove cose e adesso so usare il computer un po' meglio. L'unica cosa non bella è che duri poco. I professori e la professoressa sono stati sempre disponibili. Se dovessi ~~gli~~ dargli un voto (al laboratorio) gli darei: 9

Fig.3 – Alcuni commenti dei partecipanti.

## Bibliografia

[CSTA, 2011] CSTA. Computational Thinking. Teacher resources. Second Edition. 2011. [http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources\\_2ed-SP-vF.pdf](http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf). Accessed: 02/04/2014.

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

[Bizzarri et al, 2011] Bizzarri G., Forlizzi L., Proietti G. Informatica: didattica possibile e pensiero computazionale. Proc. of DIDAMATICA 2011.

[Casadei e Teolis, 2011] Casadei G. e Teolis A. Establishing software measurement programs. Annali della pubblica istruzione, vol. 4-5, 2011.

[Contarini, 2006] Contarini S. Un approccio al Problem Solving. Master's thesis, University of Padova (Italy), 2006.

[Leoni, 2011] Leoni L. Competenze e competizioni di informatica: valutazioni sperimentali. Master's thesis, University of Bologna (Italy), 2011.



“Se dovessi dare un voto a questo laboratorio, darei nove”

[MIT Media Lab, 2009] Lifelong Kindergarten Group at the MIT Media Lab. Scratch reference guide. 2009. <http://download.scratch.mit.edu/ScratchReferenceGuide14.pdf>. Accessed: 02/04/2014.

[Penge, 2004] Penge, S. 2004. Apprendimento della video-scrittura: come, quando ma soprattutto perché? <http://www.altrascuola.it/staff/steve/public/docu/Videoscrittura.doc>. Accessed: 08/10/2013.

[Tucker et al, 2003] Tucker, A., McCowan D., Deek F., Stephenson C., Jones J., and Verno A. 2003. A model curriculum for K-12 computer science: Report of the ACM K-12 Task Force Computer Science Curriculum Committee. Association for Computing Machinery.

[Wing, 2008] Wing, J.M. 2008. Computational thinking and thinking about computing. Phil. Trans. R. Soc. A.