

Riflessioni per un possibile curriculum nella scuola italiana (traendo ispirazione dal curriculum inglese in vigore dal settembre 2014)

Per una nuova presenza dell'informatica nella scuola è senza dubbio da considerarsi un riferimento il curriculum nazionale inglese entrato in vigore nelle scuole di ogni livello e tipo dal settembre 2014 <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>.

Un elemento distintivo di tale curriculum è l'individuazione di tre aspetti: quelli scientifici (*computer science*), quelli tecnologici (*information technology*) e quelli relativi all'alfabetizzazione digitale (*digital literacy*). Riteniamo più adatto al contesto della scuola italiana parlare semplicemente di **pensiero computazionale** (PC) e di **tecnologie digitali** (TD). Con PC intendiamo la natura scientifica dell'informatica, con TD quella tecnologica. Per entrambe la missione della scuola è partire dalla prima alfabetizzazione per arrivare ai livelli di competenza adeguati alla società contemporanea, eventualmente differenziati per tipo di scuola.

Un ulteriore elemento caratterizzante l'approccio inglese è il fatto di essere basato su **quattro "stage"** (cioè: tappa/ fase/ stadio) e non sul livello della scuola. Questo deriva dalla consapevolezza che esiste un non breve transitorio iniziale in cui studenti di livelli di scuola diversi si trovano a fare attività che li conducono ad acquisire capacità simili. Durante il transitorio la velocità di acquisizione di tali capacità dipende presumibilmente più dal livello della scuola che dalla fase. A regime, le fasi corrisponderanno a diversi livelli di scuola ed età degli studenti. Riteniamo tale approccio ben adatto anche per la scuola italiana.

Per cominciare la riflessione relativa ad un possibile curriculum per la scuola italiana formuliamo nel seguito una prima proposta di obiettivi didattici relativi alla prima e alla seconda fase. Gli obiettivi sono distinti tra obiettivi rientranti nell'ambito del pensiero computazionale (PC) e obiettivi rientranti nell'ambito delle tecnologie digitali (TD).

È inoltre importante osservare che sia a regime sia, soprattutto, nel transitorio iniziale è fondamentale un **programma nazionale per l'aggiornamento degli insegnanti**, per portarli ad acquisire quella capacità di pensiero computazionale alla quale dovranno guidare i loro studenti.

La **prima fase** consiste nel condurre gli allievi a:

- PC-1 capire cos'è un algoritmo
- PC-2 capire come un algoritmo viene realizzato mediante un programma eseguito da un "automa"
- PC-3 capire che un automa esegue istruzioni precise non ambigue
- PC-4 realizzare e mettere a punto (ovvero convincersi della loro correttezza) programmi semplici (cioè programmi con sequenze di azioni, condizioni, ripetizioni di azioni per un numero dato di volte)
- PC-5 usare il ragionamento logico per dire quale è il comportamento di programmi semplici

- TD-1 usare l'informatica per creare, organizzare, gestire e recuperare contenuti digitali
- TD-2 riconoscere usi dell'informatica nella vita comune
- TD-3 usare l'informatica badando alla sicurezza propria ed al rispetto degli altri, tenendo riservate le informazioni private
- TD-4 sapere dove chiedere aiuto in caso di problemi relativi a materiali scaricati o contatti in cui si è coinvolti su internet o attraverso altre tecnologie online

A proposito degli obiettivi della prima fase si aggiungono alcuni possibili percorsi operativi

- a. (*capire cos'è un algoritmo*): A questo proposito aggiungiamo che bisogna cominciare col far scoprire che sono algoritmi alcuni dei modi di operare, nella vita di tutti i giorni o a scuola, che attuiamo (quasi) automaticamente. A tal fine sono da considerare:

- esempi di algoritmi nella vita quotidiana scolastica e non (a cominciare dalla “Tabella del comportamento in caso suoni la campanella dell’allarme” su cui si fanno esercitazioni in tutte le scuole)
 - l’esecuzione di semplici serie di regole in ambiente ludico,
 - l’esecuzione di semplici serie di regole in ambiente combinatorio (trovare una carta specifica in un mazzo non ordinato; trovare una carta specifica in un mazzo ordinato; ordinare 10 carte; e simili)
 - l’esecuzione di regole di media complessità in ambiente combinatorio.
- b. (*capire come un algoritmo viene realizzato mediante un programma eseguito da un automa*) usare un linguaggio di programmazione formale (sia “unplugged” sia “computer based” e visuale come Scratch, Alice e simili) per la descrizione di algoritmi come alternativa al linguaggio naturale
- trasporre esempi di descrizione in linguaggio naturale di semplici sequenze di regole non ripetitive in un semplice linguaggio di programmazione formale. Le storie composte negli storyboard e poi realizzate in ambienti di programmazione visuale sono esempi di trasposizioni di questo tipo adatti anche alla primaria. Altrettanto dicasi per le attività di programmazione di piccoli robot (si pensi alle danze e anche coi robot alle storie, inventate dagli studenti, di cui i robot sono protagonisti).
 - sperimentare l’esecuzione automatica di quanto espresso nel linguaggio di programmazione formale, in particolare usando interpreti per l’esecuzione (anche parziale per una messa a punto modulare)
 - scrivere semplici sequenze di regole ed eseguirle

La **seconda fase** consiste nel condurre gli allievi a:

- PC-1 progettare, scrivere e mettere a punto programmi più complessi di quelli previsti per la prima fase perché basati su ripetizioni in numero non prefissato (ripetizioni condizionali) e introducono le variabili insieme con varie forme di input ed output.
- PC-2 risolvere problemi mediante la loro decomposizione in parti più piccole
- PC-3 risolvere problemi mediante l’utilizzo di meccanismi di astrazione (funzioni e parametri)
- PC-4 usare il ragionamento logico per spiegare il funzionamento di alcuni semplici algoritmi
- PC-5 usare il ragionamento logico per trovare e correggere errori in algoritmi e programmi
- PC-6 capire i principi alla base del funzionamento di un computer
- PC-7 capire i principi alla base del funzionamento di Internet
- PC-8 capire i principi alla base del funzionamento del Web
- PC-9 capire i principi alla base del funzionamento dei motori di ricerca

TD-1 usare le tecnologie di ricerca in modo efficace

TD-2 selezionare, usare e combinare

TD-3 capire le opportunità che Internet e le reti sociali offrono per la comunicazione e la collaborazione

TD-4 essere in grado di discernere nella valutazione del contenuto digitale

TD-5 usare la tecnologia digitale in modo sicuro, rispettoso e responsabile

TD-6 riconoscere comportamenti accettabili/inaccettabili nell’uso della tecnologia digitale

TD-7 conoscere una varietà di modi per riferire problemi riguardanti contenuti digitali o contatti

A proposito degli obiettivi della seconda fase si aggiungono alcuni possibili percorsi operativi

- a. (*progettare, scrivere e mettere a punto programmi più complessi di quelli previsti per la prima fase*) Sono molto adatte a questo scopo le storie interattive raccontate in ambienti tipo Scratch o Alice e la programmazione di piccoli robot o di Arduino (magari usando S4A Scratch for Arduino) dotati di sensori ed attuatori elementari. Si dovranno per questo considerare:
- esempi di semplici programmi che coinvolgono cicli.
 - esempi in cui diventa necessario introdurre il concetto di variabile.
 - primi cenni all’astrazione (funzioni e parametri).

Le attività svolte nel progetto “Programma Il Futuro”, consistenti in esercizi “guidati” sul pensiero computazionale (svolti sia in un ambiente visuale di programmazione sia in modalità “unplugged”) contribuiscono ad acquisire alcune delle capacità sopra citate. Ma lo fanno seguendo un percorso completamente controllato che se, da un lato, li ha resi immediatamente usabili da subito in migliaia di classi non offre, dall’altro, quella flessibilità necessaria affinché i docenti esplichino al meglio la loro azione educativa.

Pertanto, mentre sarà necessario per gli anni futuri continuare con le attività del progetto “Programma il Futuro” (che sono realizzabili in classe da qualunque insegnante senza necessità di aggiornamento) sarà necessario complementarle con la frequentazione di ambienti di programmazione visuale (che siano Scratch o Snap o Alice o Python per gli studenti piú vecchi o altro) e con le attività che in tali ambienti sono state sviluppate da tanti utenti in modo tale che iniziando con strategie di tipo usa-copia-modifica gli insegnanti arrivino ad essere indipendenti nel costruire attività proprie, arrivino cioè a cominciare ad operare in modo autonomo. E arrivino anche a provare capacità di ideazione e progetto di programmi. Bisogna inoltre considerare che attorno a questi, come ad altri ambienti, vivono comunità di utenti che hanno un ruolo insostituibile nello stimolare curiosità, indicare nuovi obiettivi, mettere in comune risorse da cui imparare nuovi concetti, last but not least: nell’offrire aiuto.