

# Il gatto e l'arte della programmazione

BARBERO Alberto

*Docente di Informatica e di Tecnologie informatiche - I.I.S. "G.Vallauri"*

*via San Michele 68, 12045 Fossano CN*

*barbero@vallauri.edu*

*Con la partenza nel presente a.s. della Riforma Gelmini nelle Scuole Medie Superiori, la nuova disciplina Tecnologie informatiche permette di introdurre concetti di logica e di programmazione di base già nella classe prima. Un'occasione da non perdere e da non sprecare per finalmente valorizzare e dare dignità ad una disciplina quale l'Informatica senza travestirla o trasformarla in qualcosa d'altro. Quale curriculum prospettare a degli studenti quattordicenni senza perdersi in complicate elucubrazioni mentali, evitando di banalizzare concetti che banali non sono e, nello stesso tempo, fornendo degli strumenti che ne sappiano catturare l'interesse e l'attenzione? Un aiuto ci viene fornito da Scratch, un linguaggio di programmazione inventato dai ricercatori dell'M.I.T. di Boston, che sa unire la semplicità d'uso alla formalità dei linguaggi artificiali, il tutto condito con molta fantasia.*

## 1. Introduzione

Il corrente anno scolastico passerà alla storia per aver visto l'avvio della Riforma Gelmini nelle Scuole Medie Superiori di tutto il paese, con tutto il seguito di polemiche che hanno accompagnato il disegno di legge sin dai suoi albori. Solo col passare del tempo potremo verificare le ricadute positive o negative del riordino sul sistema scolastico nazionale. Tra le novità introdotte spicca la proposizione di una nuova disciplina chiamata Tecnologie informatiche negli istituti tecnici (ma lo stesso dicasi per la disciplina Informatica nei licei scientifici opzione scienze applicate) che nelle linee guida ministeriali prevede l'obiettivo prioritario di far acquisire allo studente le seguenti competenze di base:

- individuare le strategie appropriate per la soluzione di problemi;
- analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico;
- essere consapevole delle potenzialità e dei limiti delle tecnologie nel contesto culturale e sociale in cui vengono applicate.

Tali competenze vengono poi articolare in conoscenze ed abilità, per fornire ai

docenti della materia indicazioni di massima sul curriculum da progettare in dipartimento e da sviluppare in classe. In particolare, per la prima competenza che fa specificatamente riferimento al tema del *problem solving*, conoscenze ed abilità vengono dettagliate nel seguente modo:

Conoscenze	Abilità
Concetto di algoritmo. Fasi risolutive di un problema e loro rappresentazione. Fondamenti di programmazione.	Raccogliere, organizzare e rappresentare informazioni. Impostare e risolvere problemi utilizzando un linguaggio di programmazione

Viene quindi – per la prima volta - esplicitamente richiesto al docente di introdurre già nella classe prima concetti di logica algoritmica e di programmazione di base, correlati ad attività di laboratorio in cui si affrontino le problematiche legate all'utilizzo di un linguaggio di programmazione per la risoluzione di problemi ed il trattamento dei relativi dati. Quale migliore occasione per dare finalmente dignità ad una disciplina quale l'Informatica, il più delle volte non valorizzata e travestita, se va bene, da Patente Europea del Computer o trasformata in ore di Matematica, Fisica o altro a seconda del docente incaricato all'insegnamento?

## 2. Il problema del curriculum

Compito dei docenti, a questo punto, è quello di progettare un curriculum che, partendo dai contenuti esplicitati nelle linee guida, sia in grado di coniugare le conoscenze ed abilità indicate in un percorso che non risulti troppo ostico per degli studenti in età adolescenziale e che non banalizzi concetti basilari per lo sviluppo di capacità di *problem solving* utili in tutti i contesti di studio del quinquennio ed oltre. Il problema del curriculum è stato risolto (e al momento attuale è in fase di sperimentazione e monitoraggio nelle classi prime) anticipando parte del curriculum della classe terza informatica, estrapolandone i concetti teorici di partenza che costituiscono i fondamenti della programmazione iterativa, tralasciando formalismi e ragionamenti più complessi. Parlando di algoritmi, ad esempio, si farà solo un accenno alle proprietà di generalità e di complessità, così come non si pretenderanno al momento soluzioni algoritmiche del tutto ottimizzate. Trattando di cicli strutturati, non si porrà enfasi sulla differenza esistente tra il postcondizionale ed il precondizionale, ma ci si accontenterà di far capire a cosa serve un ciclo e quando lo si utilizza.

Pertanto, la programmazione del modulo didattico dal titolo "Problemi ed algoritmi" elaborato dai docenti del Dipartimento di Informatica del Vallauri per la disciplina Tecnologie informatiche delle classi prime prevede nel dettaglio il seguente percorso:

---

*Prerequisiti/conessioni con moduli e/o unità didattiche precedenti:*

- Conoscenza del concetto di informazione e sua rappresentazione

---

*Competenze finali del modulo:*

- Imparare ad utilizzare degli strumenti di descrizione formale degli algoritmi (diagrammi a blocchi)
- Imparare a risolvere semplici problemi con algoritmi utilizzando una corretta metodologia di lavoro

- Codificare un programma usando il linguaggio *Scratch* e saperne verificare il funzionamento

#### Contenuti:

- Analisi dettagliata di un problema
- Definizione dei passi fondamentali da compiere per giungere alla soluzione
- Definizione di algoritmo e le sue caratteristiche
- Rappresentazione degli algoritmi con diagrammi a blocchi
- Strutture di controllo fondamentali: sequenziali, condizionali, iterative
- Realizzazione di semplici programmi usando il linguaggio *Scratch*

### 3. Dalla teoria alla pratica

Ma come rendere i contenuti teorici di questo modulo interessanti ai più e con quali strumenti implementare le applicazioni in laboratorio facendo crescere la passione e la curiosità per la programmazione a dei quattordicenni? La soluzione comincia a delinearsi dopo un'accurata ricerca di strumenti per la programmazione diversi dai soliti linguaggi usati nel triennio o all'università. E la scelta ricade su *Scratch*, un linguaggio di programmazione visuale sviluppato nel 2007 dai ricercatori del *Lifelong Kindergarten Group* dell'M.I.T. MediaLab di Boston, guidati dal prof. Mitchel Resnick. Le caratteristiche principali di funzionamento sembrano essere tutte quelle che fanno al nostro caso. Si tratta infatti di un linguaggio creato apposta per introdurre a studenti nella fascia



(fig.1)

dell'obbligo scolastico i concetti di base della programmazione e del *problem solving* attraverso uno strumento che ad un primo approccio sa molto di ludico ma che in realtà esercita i discenti alla logica e al ragionamento. Infatti permette di elaborare variabili e liste di valori, offre controlli per la selezione e l'iterazione, permette di realizzare animazioni via via più complesse, offre la possibilità di far eseguire più processi

contemporaneamente e di farli interagire, permette l'implementazione del paradigma ad eventi e molto altro ancora. La codifica dello *script* avviene impilando blocchi di forma e colore diverso, a seconda della funzione e della categoria di appartenenza, che vanno ad incastrarsi (fig.1) come nel gioco del Lego. *Scratch* è inoltre un prodotto scaricabile liberamente dal sito <http://scratch.mit.edu>, stabile, potente (ogni giorno gli studenti scoprono qualche nuova funzionalità), duttile e ideale per lo sviluppo di applicazioni ludiche, animazioni grafiche, ipertesti ma anche per l'implementazione delle classiche applicazioni che si sviluppano quando si impara la programmazione da zero. Il sito di *Scratch*, in pieno stile web 2.0, ospita manuali gratuiti, gallerie di progetti, video esplicativi, forum di discussione, e permette di scaricare più di 1.623.937

(al 3 marzo 2011) progetti completamente gratuiti con licenza *Creative Commons* o di caricare i propri progetti condividendoli con gli altri utenti sparsi per il mondo.

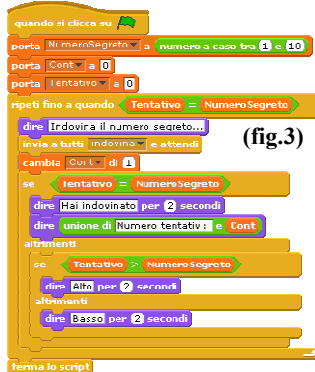
#### 4. Fantasia in cattedra

Tutto risolto? Sarebbe di sì. Ma come spesso capita, “avere una bici” non significa per forza “essere capaci a pedalare”. Infatti, uno dei principali problemi a cui sta lavorando il Dipartimento di Informatica del Vallauri è quello di inquadrare la didattica del modulo “Problemi ed algoritmi” in un’ottica di rinnovamento ed innovazione, partendo dal presupposto che non possiamo introdurre la programmazione nelle classi prime con le stesse modalità che seguiamo nelle classi terze dell’indirizzo informatico, anche per il ridotto monte ore a disposizione. Un aiuto ci viene come sempre scartabellando il materiale informativo che si trova sul sito di *Scratch* dove ci viene suggerito che un’iterazione *repeat until* può essere spiegata facendo muovere lo *sprite* della stella marina un passo dopo l’altro sino a toccare lo *sprite* della vongola, per poi salutare. Oppure come far interagire più processi tra di loro inviando un messaggio attraverso il comando *broadcast* e ricevendolo attraverso il comando *when I receive*. Da queste idee di partenza ne possono nascere altre per modulare la didattica coniugando, quando possibile, aspetti ludici al rigore



scientifico. Nell'esempio seguente, il classico problema del gioco “Indovina il numero segreto” può essere implementato (fig.2) con una iterazione *ripeti fino a quando* facendo giocare l’utente contro il programma, mentre in quella sotto (fig.3) lo stesso problema viene realizzato facendo interagire due processi, uno che chiede di indovinare il numero e l’altro che prova ad indovinare, associati ciascuno ad uno *sprite* diverso, mediante i comandi *invia a tutti* e *quando ricevo* rendendo l’esercitazione più interessante e introducendo il tema della comunicazione tra processi. Lo sforzo del docente deve essere quindi quello di affiancare alle classiche esercitazioni iniziali la sperimentazione di nuove

metodologie di proposizione



di problemi che sfruttino le potenzialità offerte da *Scratch* per introdurre concetti anche non banali senza perdere l’interesse della classe. Una scommessa non facile che richiede al docente di rimettersi in gioco per appassionare una platea non facile e per non buttare via un’occasione per dimostrare capacità e professionalità.