



I CIRCOLO DIDATTICO
"GENNARO SEQUINO"

Via Roma, 80 – 80018 MUGNANO DI NAPOLI

Telefono 081-5710410/Fax 081-7451142

Distretto 26 – Codice NAEE15100R – Codice fiscale 80101770636

@-mail: naee15100r@istruzione.it

@-mail certificata: NAEE15100R@PEC.ISTRUZIONE.IT

web site: www.cdsequino.gov.it



FSE
FESR

Piccole variAzioni

Percorso di *coding* per gli alunni delle classi quinte

10 Reasons to Teach Coding
By Brian Aspinall @mrspinall

1 Coding allows students to create content, not just consume it.

2 Coding empowers students and gives them tools to express themselves in really cool ways.

3 Coding teaches storytelling with games and animations.

4 Coding is a place for students to take risks & fail safely.

5 Coding is inclusive & builds self-confidence.

6 Coding supports many principles of mathematics.

7 Coding teaches problem-solving and critical/analytical thinking skills.

8 Coding is a new type of literacy and will be a large part of future jobs.

9 Coding develops teamwork & collaborative skills.

10 Coding can help humanity.

BONUS: Coding gives you SUPERPOWERS!

@sylvia duckworth

Anno scolastico 2016/2017

Premessa

Una delle definizioni più convincenti dell'espressione "***pensiero computazionale***" è stata utilizzata per la prima volta dalla studiosa e informatica statunitense Jeannette Wing per indicare "il processo mentale che sta alla base della formulazione dei problemi e delle loro soluzioni così che le soluzioni siano rappresentate in una forma che può essere implementata in maniera efficace da un elaboratore di informazioni sia esso umano o artificiale". Si tratta, in altre parole, dello sforzo che un individuo deve mettere in atto per fornire a un altro individuo, ad una macchina tutte -e solo- le "istruzioni" necessarie affinché questi, eseguendole, sia in grado di portare a termine il compito dato. Come si vede chiaramente non si tratta di una competenza connessa esclusivamente all'informatica, ma di una *skill* che dovrebbe appartenere al bagaglio individuale di ciascuno studente per meglio affrontare lo studio di qualsiasi disciplina. È altrettanto evidente, però, come l'informatica costituisca un terreno privilegiato per consentire agli alunni di esercitare e sviluppare questa fondamentale funzione del pensiero. Nella programmazione informatica, infatti, vengono definite, mediante uno specifico *linguaggio*, le *procedure* che vengono poi attuate da un *esecutore*, che opera nell'ambito di un *contesto prefissato*, per raggiungere gli *obiettivi assegnati*.

Uscendo poi dallo specifico contesto delle macchine, è evidente che tale processo mentale, costituito dalla combinazione di *metodi caratteristici* e di *strumenti intellettuali* (cfr. programmailfuturo.it), abbia connotati di valore generale e generalizzabile. Proprio questa caratteristica rappresenta il motivo principale della scelta, operata in sede ministeriale, di introdurre il pensiero computazionale fin dalla scuola primaria.

Il percorso proposto come arricchimento dell'offerta formativa (AOF) mira pertanto, attraverso il coding, a favorire la creatività, ad aiutare i bambini ad essere

maggiormente consapevoli delle proprie risorse personali ma anche a valorizzare le differenze individuali che emergono nelle scelte divergenti all'interno delle esercitazioni pratiche. Al termine del percorso ogni alunno partecipante avrà appreso i primi rudimenti del pensiero computazionale, primo passo per interagire in maniera più consapevole con i dispositivi tecnologici non più da semplici fruitori, ma anche per incrementare la capacità di risolvere problemi che è trasversale a tutte le discipline. Non si tratta, dunque, di insegnare ai bambini a programmare, ma di arricchire il loro bagaglio di strumenti utili a pensare, riflettere, imparare.

Metodologia di intervento

La convinzione, ormai universalmente accettata, che il pensiero computazionale debba essere insegnato a tutti i bambini non ha ancora trovato una risposta univoca alla domanda su quale sia il metodo migliore per insegnarlo in maniera strutturata ed efficace. È tuttavia diffusa l'opinione che, in attesa di un metodo scientificamente validato, la pratica stessa della programmazione, a vari livelli di complessità in funzione dell'età degli allievi, possa rappresentare il veicolo più immediato efficace. Ci si rifà, in tal senso, all'insegnamento di Seymour Papert, che per primo utilizzò l'espressione "*computational thinking*". La sua teoria dell'apprendimento, nota come **costruzionismo**, si basa sull'assunto che la mente umana, per poter imparare, ha bisogno di creare "artefatti", ovvero oggetti e dispositivi che facilitano una rappresentazione concreta del mondo con cui interagisce. Il computer, secondo Papert, è in tal senso un prezioso strumento didattico poiché, grazie alla programmazione, consente la creazione di artefatti di questa natura. Gli allievi di Papert, nel MediaLab del MIT, hanno successivamente realizzato una struttura *ad hoc* per l'insegnamento del pensiero computazionale mediante la creazione di storie interattive e videogiochi (gli artefatti di cui si diceva), da cui è nato **Scratch**, un ambiente di *programmazione visuale* che consente ai bambini di creare in maniera

semplice e intuitiva storie animate, giochi e simulazioni. La piattaforma, oggi diffusa in tutto il mondo, rappresenta lo strumento di riferimento per insegnare ai bambini il pensiero computazionale attraverso la programmazione ed è alla base delle stesse attività contenute nella piattaforma **programmailfuturo.it** promossa dal MIUR in collaborazione con il CINI (Consorzio Nazionale Interuniversitario per l'Informatica). Ispirata al pensiero di Papert e sostenuta dall'utilizzo degli strumenti appena menzionati, la metodologia dell'intervento proposto prevede che i docenti si trasformino in facilitatori e i bambini, resi il più possibile indipendenti, diventino i veri responsabili del proprio apprendimento, secondo l'impostazione della *flippedclassroom*. Il gruppo, infatti, è chiamato a comunicare e condividere le idee in modo tale che ciascuno possa apprendere dall'altro anche mediante una corretta gestione degli errori. Quest'ultima funzione sarà perseguita privilegiando ove possibile il lavoro a coppie o in piccolo gruppo secondo le modalità del *cooperative learning* e del *pairprogramming*.

Destinatari

Il progetto si rivolge a due gruppi di 20/25 alunni delle classi quinte dei plessi Sequino e San Giovanni per un totale di 50 bambini coinvolti.

Spazi e strumenti

Il *setting* prevede l'utilizzo dei laboratori di informatica dei due plessi e della disponibilità di PC con collegamento a internet (un computer per ciascun alunno partecipante o eventualmente uno ogni due partecipanti in funzione delle attività e/o delle disponibilità). Si prevede inoltre l'utilizzo di schermo e videoproiettore o, in alternativa, della LIM.

Si farà infine ricorso alle risorse disponibili in rete e accreditate dal MIUR:

- Scratch

- programmailfuturo.it
- code.org

Obiettivi educativi

Gli obiettivi educativi dell'intervento puntano a contribuire allo sviluppo di competenze trasversali a tutte le discipline che, al termine del percorso, saranno osservabili nei seguenti comportamenti assunti dall'alunno:

- persegue con costanza gli obiettivi proposti;
- lavora in gruppo rispettando ruoli e funzioni;
- gestisce positivamente i propri errori;
- impara dagli altri;
- utilizza le tecnologie in maniera costruttiva;
- possiede maggiore autostima;
- possiede maggiori *skill* per affrontare e risolvere i problemi.
- esprime se stesso più facilmente;
- comunica e lavora con gli altri per raggiungere un obiettivo o una soluzione condivisa;
- pone domande coerenti e appropriate.

Obiettivi didattici

Con riferimento agli specifici concetti base del pensiero computazionale (secondo il *framework* sviluppato dal *Lifelong Kindergarten del MIT MediaLab*), al termine del percorso, l'alunno ha familiarizzato con i seguenti:

- **Sequenza:** un'attività può essere espressa attraverso una serie consecutiva di singoli step o istruzioni.
- **Ciclo:** meccanismo per eseguire più volte la medesima sequenza in maniera iterativa.

- **Evento:** il verificarsi di un'azione causa lo scatenarsi di un'altra azione.
- **Parallelismo:** eseguire sequenze di istruzioni differenti allo stesso tempo.
- **Condizione:** possibilità di prendere decisioni sulla base del verificarsi di determinate situazioni.
- **Operatori:** supporti per la manipolazione di numeri e stringhe di caratteri.
- **Dati:** valori che possono essere salvati, recuperati e modificati durante l'esecuzione di un programma.

Con riferimento, invece, alle pratiche di pensiero computazionale, l'alunno :

- **è incrementale e interattivo:** progetta in modo adattativo modificando la pianificazione a mano a mano che si avvicina alla soluzione del problema;
- **testa e debugga:** individua problemi ed errori e li corregge;
- **riusa:** riconosce come alcune parti di soluzione possono essere riusate nella stessa o riapplicate a problemi simili;
- **remixa:** prende spunto da idee di altri per costruire cose più complesse di quelle che avrebbe potuto realizzare per conto proprio, dando un'ulteriore spinta alla propria creatività;
- **astrae:** riduce la complessità dei problemi affrontati mantenendo solo alcuni aspetti e tralasciandone altri;
- **modularizza:** scompone un problema complesso in problemi più semplici e, risolvendo i problemi più semplici, risolve anche il problema complesso.

Tempi e docenti coinvolti

Si prevede la realizzazione di n.12 incontri da 2,5 ore per ogni gruppo. Per ciascun incontro è prevista la contemporanea presenza di n. 2 docenti.

Verifica e valutazione

La verifica e la valutazione dell'intervento prevedono due azioni:

- Valutazione dei lavori realizzati dei ragazzi e delle competenze di programmazione raggiunte.
- Attività di feedback con i ragazzi per verificare il gradimento e l'efficacia del progetto e analisi dei risultati.

Il primo punto sarà sviluppato mediante pratiche di osservazione sistematica e mediante la somministrazione di prove, strutturate sulla scorta delle attività svolte, sia in itinere che al termine del percorso.

Con riferimento, invece, al gradimento dell'attività, si procederà alla somministrazione a tappeto di un questionario strutturato di gradimento riferito alla qualità percepita in relazione a:

- docenti (puntualità, qualità dell'interazione, ...);
- attività (qualità della proposta didattica);
- spazi e attrezzature (adeguatezza di spazi e strumenti utilizzati);
- organizzazione (orari, tempi didattici, flessibilità ...)

I dati così raccolti costituiranno la base per la realizzazione di un **report finale** con l'individuazione di punti di forza e criticità e indicazioni per la futura progettazione di attività analoghe.