

**BRICKS
FUORI NUMERO**

Coding e Robotica: una proposta di percorso per la scuola secondaria di primo grado

a cura di:

Giovanni Cielo



Coding, Robotica, Pensiero computazionale, Processing, Arduino

Introduzione

Il *pensiero computazionale* è la capacità di un individuo di risolvere un problema suddividendolo in parti più semplici e costruendo una procedura. Si tratta dunque di un'abilità di *problem solving* e di una *forma mentis* che si può insegnare fin dai primi anni dell'infanzia e che accompagnerà l'individuo per tutto il resto della vita. Lo possiamo considerare come una ulteriore possibilità che l'uomo ha nelle sue mani per dare una risposta ai problemi che dovrà affrontare nella vita, di qualsiasi tipo essi siano.

Al giorno d'oggi vi è una grande quantità di proposte di percorsi didattici e attività che riguardano il pensiero computazionale per le scuole di ogni ordine e grado ma sento il bisogno di proporre una strada differente cercando di non perdere di vista l'obiettivo principale che è quello della costruzione di tale tipo di abilità e utilizzando strumenti efficienti, efficaci e attraenti quali il *coding* e la robotica in una maniera che in parte si discosta da tutti quelli già esistenti.

Voglio dunque partire dalla costruzione del concetto di programma attraverso l'uso di diagrammi di flusso per realizzarlo poi utilizzando un linguaggio testuale quale [Processing](#) che ci permette di utilizzare fin da subito in maniera semplice forme geometriche anche in movimento. Questo, da una parte rende la programmazione subito attraente per un ragazzo di 11 anni, dall'altra ci permette collegamenti con discipline come la geometria o il disegno tecnico. L'aver appreso il linguaggio di programmazione Processing ci porta, a partire dalla seconda metà della seconda classe e, certamente nella terza, ad introdurre la robotica attraverso l'uso di Arduino che viene programmato, appunto, con una versione semplificata di Processing. In tale modo i ragazzi si troveranno già pronti dal punto di vista della programmazione e potranno focalizzare i loro sforzi sugli aspetti dell'elettronica. Il risultato finale è quello di voler tendere verso una *competenza digitale* che costituisce anche una buona base per il mondo del lavoro, oggi certamente molto interessato a figure professionali che si sappiano orientare nel campo della robotica.

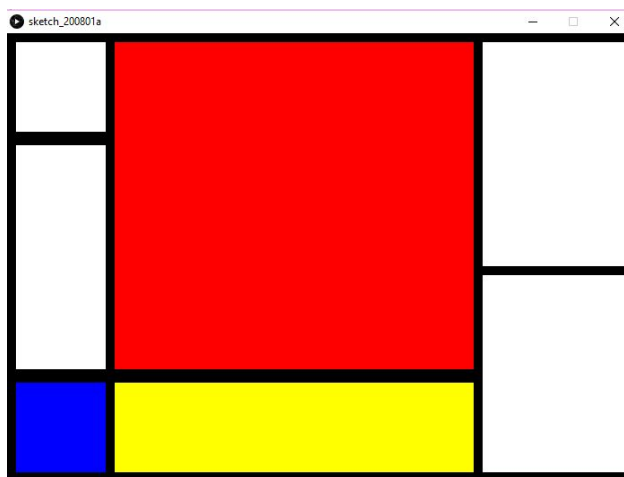


Figura 1 - Esempio di programma realizzato con Processing.

Perché programmare?

La scuola secondaria di primo grado è una scuola di tipo *orientativo*, il suo scopo deve essere dunque quello di fornire delle basi generali e conoscitive di diverse discipline e aiutare in questo modo i ragazzi ad individuare la propria strada, educarli a fare la scelta che caratterizzerà il loro futuro professionale. Sembrerebbe fuori luogo dunque insegnare la programmazione a tutti gli studenti, anche a coloro che magari un giorno sceglieranno un corso di studi che li porterà a diventare avvocati, medici, elettricisti, guardie forestali, ufficiali dell'esercito, ecc. ecc.. L'insegnamento di un linguaggio di programmazione, nei

suoi dettagli, sembrerebbe più idoneo, e di fatto lo è, per una scuola secondaria di secondo grado di tipo tecnico, dove si presume che gli alunni che la frequentano abbiano già fatto questa scelta.

C'è da dire però che un altro aspetto fondamentale della scuola secondaria di primo grado è l'insegnamento di un metodo di studio. Oserei dire che un ruolo molto importante di questo tipo di scuola è *l'educazione allo studio*: portare i ragazzi verso la competenza dell'*imparare ad imparare*, chiave fondamentale per il successo della vita, qualsiasi sia l'orientamento allo studio che poi sceglieranno. Ecco, lo scopo della programmazione nella scuola secondaria di primo grado è proprio in questo: nell'offrire allo studente un ulteriore modo di formulare il suo pensiero. Il progettare un algoritmo come soluzione ad un qualsiasi tipo di problema è un'ottima palestra per la formazione di un modo di affrontarli diverso e complementare da quello fornito attraverso la pratica di altre discipline. Anche se a primo acchitto vedersi costretti a trovare una soluzione utilizzando solo e sempre gli stessi costrutti di base, opportunamente incastrati tra loro, potrebbe sembrare un tarpare le ali alla propria creatività, la programmazione offre una strada ordinata, procedurale, facilmente verificabile, replicabile, efficace e perfino efficiente per raggiungere lo scopo. Associata, combinata, interrelata con gli altri metodi se vogliamo più classici, la programmazione dunque è in grado di ampliare le capacità di *problem solving* dei ragazzi. Inoltre essendo, per la sua natura, strettamente procedurale può essere un valido aiuto per gli alunni con *bisogni educativi speciali*.

Il diagramma di flusso

Se l'interesse dell'insegnante è quello di far maturare nell'alunno una *competenza digitale*, ovvero la capacità di scrivere procedure che possano risolvere un qualsiasi tipo di problema - non solo informatico - scomponendolo in sotto-problemi più semplici, l'attenzione non può essere concentrata sull'insegnamento di uno specifico linguaggio di programmazione bensì sulla metodologia di approccio. Dovendosi dunque astrarre dalla programmazione è fondamentale introdurre fin da subito il concetto di *diagramma di flusso* come vero strumento di progettazione della soluzione del problema. In questo modo si fornisce uno strumento universale, ovvero indipendente dal linguaggio di programmazione utilizzato, semplice perché composto da elementi grafici collegati tra loro, simili, tra l'altro, a quelli utilizzati in una mappa concettuale ma allo stesso tempo molto potente perché allena la mente a trovare soluzioni facendo riferimento ad un numero piuttosto limitato di elementi da combinare tra loro in infiniti modi.

E' possibile sintetizzare l'utilità didattica del diagramma di flusso nei seguenti punti.

- **Il diagramma di flusso costituisce un *linguaggio universale***, con i suoi formalismi e le sue regole. Non diverso nella sostanza dal formalismo del disegno tecnico. Fornisce dunque uno strumento di comunicazione alternativo alla parola scritta o orale.
- **Il diagramma di flusso stimola la *fantasia***. Avendo a disposizione un numero limitato di elementi da combinare tra loro per risolvere qualsiasi tipo di problema è proprio la capacità personale nel trovare la giusta combinazione dei primi per risolvere il secondo l'elemento fondamentale del successo.
- **Si può accedere al diagramma di flusso per *livelli***. L'alunno si può limitare a trovare una soluzione *funzionante (efficacia)* o può continuare a ragionare sul modo di combinare gli elementi base per migliorarla (*efficienza*). Inoltre possono essere pensati infiniti modi per raggiungere lo scopo, più o meno efficaci ed efficienti. Questo costituisce un buon proposito per la valutazione del lavoro degli studenti potendo, ad esempio assegnare una valutazione insufficiente se non si è riusciti a trovare una soluzione al problema, sufficiente se la soluzione è stata trovata ma non è la migliore, buona se la soluzione è anche efficace, ottima se la soluzione proposta contiene elementi *personalizzati* dallo studente (dimostrazione di raggiungimento di una competenza).

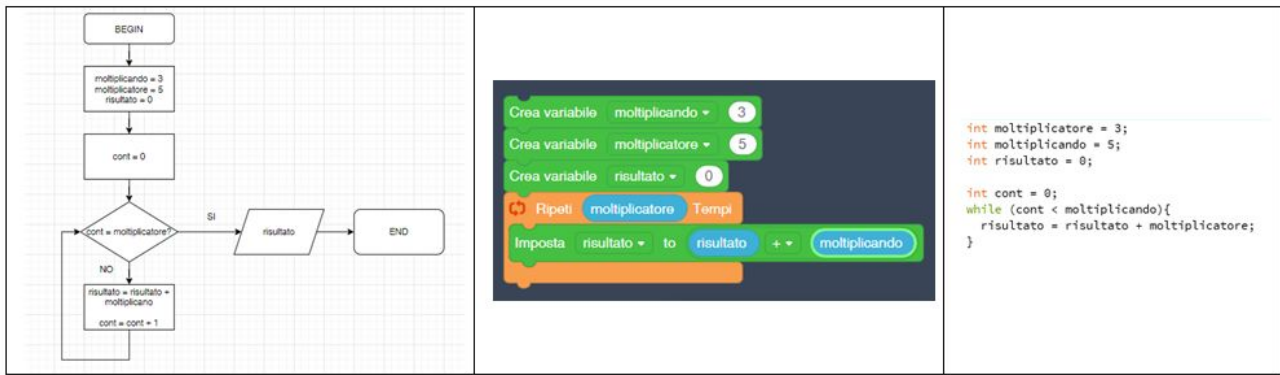


Figura 2 - Diagramma di flusso, linguaggio di programmazione a blocchi e linguaggio di programmazione testuale.

La scelta di Processing

Processing è un linguaggio di programmazione *open source*, di tipo testuale, particolarmente orientato verso le arti grafiche per le quali si propone particolarmente semplice da utilizzare.



Figura 3 - Il logo del linguaggio di programmazione Processing.

Questa sua particolarità nell'approcciare in maniera diretta e particolarmente semplice l'aspetto grafico della programmazione di un software consente sia di stimolare maggiormente gli alunni, in quanto fin dalle prime righe di codice possono trovare un riscontro grafico attraente, sia di trovare fin da subito collegamenti tra diverse discipline come ad esempio geometria, arte, matematica e tecnologia. Se lo stesso formalismo viene utilizzato da più docenti contemporaneamente lo sforzo necessario da fare per insegnare il linguaggio stesso viene più facilmente *ammortizzato*.

La scelta di utilizzare un linguaggio testuale piuttosto che un più semplice linguaggio a blocchi in stile Scratch fin dai primi anni della scuola secondaria di primo grado, quindi molto probabilmente come prima esperienza di programmazione, è stata fatta in considerazione della difficoltà riscontrata da diversi alunni nel passaggio da un linguaggio di programmazione a blocchi ad uno testuale quando si sono trovati davanti a questa necessità.

In altre parole possiamo affermare che, prima o poi, chi si interesserà di programmazione per qualsiasi motivo, dovrà essere in grado almeno di leggere un linguaggio testuale che ha un uso reale mentre i linguaggi a blocchi possono assolvere solo un compito didattico: sarebbe davvero impensabile poter realizzare un software complesso utilizzando un linguaggio a blocchi. Il passaggio dalla semplice scelta di un blocco preconfigurato piuttosto di un altro alla scrittura di una riga di codice con tutto il formalismo che ne consegue (obbligo di parentesi, punteggiatura, convenzioni per i simboli di confronto o di operatori matematici e logici, ecc. ecc.) non è affatto immediata né semplice. Nell'ottica dunque di voler portare gli alunni ad una *competenza digitale*, ovvero alla capacità di utilizzare *conoscenze* e *abilità* per

operare in *autonomia* e *originalità*, pensiamo che valga la pena investire inizialmente un tempo di apprendimento maggiore per far conoscere uno strumento decisamente più utile nel futuro.

Un ulteriore e non meno importante fattore che mi ha portato alla scelta di Processing tra i tanti linguaggi di programmazione esistenti è che Arduino viene programmato con un linguaggio che è una sua semplificazione e, per questo, del tutto simile. Nel momento in cui si proporrà ai ragazzi della classi più grandi della scuola secondaria di primo grado o alle prime classi della scuola secondaria di secondo grado le prime esperienze di robotica con la scheda di prototipazione di Massimo Banzi, certamente si incontreranno molte meno difficoltà nella sua programmazione, aspetto finora ritenuto critico nell'uso di Arduino nelle scuole.



Figura 4 - La scheda di prototipazione Arduino UNO R3.

La robotica costituisce un interessante strumento didattico per la scuola secondaria di primo e secondo grado in quanto è senza ombra di dubbio una modalità di proposta di attività di problem solving che riscuote interesse ed entusiasmo nei ragazzi e nelle ragazze nella fascia di età preadolescenziale e adolescenziale e al tempo stesso è in grado di fornire la basi per una competenza spendibile nel mondo del lavoro, oggi sempre più alla ricerca di personale qualificato in tale ambito. Naturalmente si possono proporre attività di robotica in diversi modi e con diversi strumenti, più o meno semplici, più o meno efficaci.

Arduino non è sicuramente la piattaforma a disposizione più semplice da utilizzare ma, essendo *open hardware*, è tra le più economiche e questo permette anche in una scuola pubblica di poter allestire un laboratorio di robotica educativa con cifre anche al di sotto dei 100 euro; è aperta e quindi consente una quantità pressoché infinita di esperimenti offrendo opportunità di creatività ai ragazzi. Massimo Banzi, inventore di Arduino, può essere infine presentato come figura di riferimento nella *competenza di imprenditorialità* per i ragazzi che oggi sono spesso disorientati e paurosi nei confronti del loro posto nel mondo del lavoro. Riteniamo quest'ultima considerazione piuttosto rilevante nel contesto di forte disagio che vivono gli adolescenti di oggi.

Una proposta di percorso didattico

Alla luce di quanto esposto nei paragrafi precedenti si vuole proporre di seguito una prima proposta di percorso didattico commentandola nei punti più importanti. Il lettore dovrà tenere ovviamente in considerazione che, al momento in cui scrivo, si tratta solo di un progetto di percorso didattico interdisciplinare al quale manca ancora la parte di sperimentazione. Per questo motivo si ritiene piuttosto interessante uno scambio di opinioni volte al perfezionamento della proposta e l'attuazione in un'ottica di ricerca-azione che porti ad una riflessione su quanto nel progetto iniziale ha avuto un successo educativo e quanto, invece, ha necessità di essere migliorato.

Classe Prima

L'attività principale da svolgere nella prima classe è quella della **scoperta** della programmazione, della sua utilità e complessità e della necessità della definizione di un formalismo - il linguaggio - con delle regole a cui attenersi affinché possa essere comprensibile da tutti e dalla macchina da programmare. Particolare attenzione andrà rivolta alle fasi di **innesco** (o *warm-up*) in modo che l'attività da effettuare possa essere il più coinvolgente ed entusiasmante possibile. Trattandosi di un contesto nuovo per gran parte degli studenti si ha la possibilità di coinvolgere tutti, anche coloro che nelle attività più tradizionali sono risultati meno brillanti. L'immediato riscontro pratico tipico delle attività laboratoriali avrà sicuramente un importante peso riguardo alla motivazione degli allievi. Ma attenzione a quando si dovranno affrontare le prime vere difficoltà (programma che non compila, errore incomprensibile, risultato non raggiunto, ...) perché si potrebbe avere una forte demotivazione. E' necessario allora insistere a non demordere affinché i ragazzi imparino a superare, con l'impegno, le difficoltà incontrate.

| Unità di apprendimento #1 - Programmare | | | |
|--|--|--------------------------|--|
| Tempi | 1h di lezione in classe + 1h di esercitazione a casa | | |
| Spazi | classe | Periodo ipotetico | ottobre |
| Discipline coinvolte | Tecnologia | Metodologia | lezione frontale cooperative learning |
| <p>Far percepire agli studenti l'importanza e la difficoltà della programmazione prescindendo da ogni forma di linguaggio. Si può proporre, ad esempio, un'azione comune come camminare, scrivere, mangiare, preparare un caffè e chiedere ai ragazzi di dare una serie di istruzioni ad un automa non intelligente ma capace solo di eseguire le istruzioni che gli vengono impartite e che è in grado di comprendere in modo da farlo riuscire nell'impresa. I ragazzi noteranno che anche un semplice comando come "prendi la penna" diventa incomprensibile ad un automa e che bisogna dettagliare ogni singola procedura (scomposizione di un problema in più sotto-problemi più semplici). "Prendi la penna" dovrà diventare la sequenza di ogni singolo movimento che ogni singolo dito dovrà compiere fino a riuscire ad afferrare la penna.</p> | | | |
| Elementi di programmazione: concetto di programmazione. | | | |

| Unità di apprendimento #2 - Il diagramma di flusso | | | |
|--|--|--------------------------|--|
| Tempi | 2h di lezione in classe + 3h di esercitazione a casa | | |
| Spazi | classe | Periodo ipotetico | ottobre/novembre |
| Discipline coinvolte | Tecnologia | Metodologia | lezione frontale cooperative learning |
| <p>Presentazione del formalismo del diagramma di flusso come strumento per progettare e comunicare in maniera universale la soluzione ad un problema. Principali blocchi logici e modo di collegamento tra di essi. Proposta di formalizzare una soluzione ad un semplice problema sia per gruppi che individualmente come compito per casa.</p> | | | |

Elementi di programmazione: il diagramma di flusso.

Unità di apprendimento #3 - Creiamo un nostro linguaggio: la sintassi
[FACOLTATIVO]

Tempi 3h di lezione in classe + 3h di esercitazione a casa

Spazi classe **Periodo ipotetico** novembre/dicembre

Discipline coinvolte Italiano
Tecnologia **Metodologia** lezione frontale
cooperative learning

Scoperta della sintassi e delle regole di composizione di una frase. Ipotizzare di costruire un linguaggio dando delle regole. Il diagramma di flusso ha una sintassi? Quale?

Elementi di programmazione: la sintassi.

Unità di apprendimento #4 - Le coordinate cartesiane

Tempi 4h di lezione in classe + 3h di esercitazione a casa

Spazi classe **Periodo ipotetico** gennaio

Discipline coinvolte Geometria
Disegno tecnico **Metodologia** lezione frontale
laboratorio

Le coordinate cartesiane nel piano. Individuazione di un punto nel piano. Individuazione di un segmento nel piano. Individuazione di poligoni nel piano (triangoli, rettangoli, poligoni in generale). Individuazione di una circonferenza nel piano.
Disegno tecnico di poligoni date le coordinate cartesiane su foglio quadrettato e poi su foglio da disegno. L'ellisse del giardiniere.

Elementi di programmazione: -

Unità di apprendimento #5 - Introduzione a Processing

Tempi 1h di lezione in classe

Spazi lab. informatica **Periodo ipotetico** febbraio

Discipline coinvolte Tecnologia **Metodologia** laboratorio

Introduzione a Processing. Installazione. GUI. Salvataggio e caricamento dei file. Condivisione dei file.

Elementi di programmazione: l'uso dell'IDE.

| Unità di apprendimento #6 - Disegniamo con Processing | | | |
|---|--|--------------------------|-------------|
| Tempi | 1h di lezione in classe + 1h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | febbraio |
| Discipline coinvolte | Tecnologia | Metodologia | laboratorio |
| Preparazione di un foglio da disegno a quadretti: costruito for, funzioni line, rect, ellipse, size, stroke. Disegno di segmenti, rettangoli, ellissi e cerchi date le coordinate. Personalizzazione con i colori. Replica della tavola di disegno tecnico effettuata nell'unità di apprendimento precedente. | | | |
| Elementi di programmazione: concetto di funzione, funzioni line, rect, ellipse, size e stroke. | | | |

| Unità di apprendimento #7 - Mondrian | | | |
|--|--|--------------------------|-------------|
| Tempi | 1h di lezione in classe + 2h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | marzo |
| Discipline coinvolte | Arte e immagine Tecnologia | Metodologia | laboratorio |
| Mondrian. Riproduzione di un dipinto di Mondrian sia su tavola che con Processing. | | | |
| Elementi di programmazione: funzioni background, setup, draw, e fill. | | | |

| Unità di apprendimento #8 - In movimento | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 1h di lezione in classe + 2h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | marzo |
| Discipline coinvolte | Tecnologia Scienze | Metodologia | lezione frontale laboratorio |
| Realizzare un programma che simula la caduta per forza di gravità di una pallina e il suo controllo attraverso le frecce della tastiera. | | | |
| Elementi di programmazione: concetto di variabile, tipi di variabile, incremento di una variabile, costruito if, funzione keyPressed. | | | |

| Unità di apprendimento #9 - Orologio | | | |
|---|--|--------------------------|---------------------------------|
| Tempi | 6h di lezione in classe + 6h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | aprile/maggio/giugno |
| Discipline coinvolte | Matematica Geometria Tecnologia | Metodologia | lezione frontale laboratorio |
| Partendo dalla definizione di angoli e dalle operazioni su di essi, realizzare un orologio programmando il movimento angolare delle lancette. | | | |
| Elementi di programmazione: funzioni fill, map, beginShape, endShape, vertex, costruito for. | | | |

Classe Seconda

Nella seconda classe, dopo aver avuto l'occasione di scoprire il mondo della programmazione e di Processing, si vuole portare gli alunni ad assumersi le proprie **responsabilità** chiedendo loro di applicarsi nello studio e realizzazione di attività più complesse e, introducendo in maniera graduale la robotica utilizzando la scheda di prototipazione Arduino UNO R3. Il passaggio non è banale soprattutto perché dal contesto *ideale* della realizzazione di un software si passa a quello *reale* della programmazione di un oggetto fisico con tutte le sue inevitabili imprecisioni.

| Unità di apprendimento #1 - Le proiezioni ortogonali | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 3h di lezione in classe + 3h di esercitazione a casa | | |
| Spazi | classe lab. informatica | Periodo ipotetico | ottobre |
| Discipline coinvolte | Disegno tecnico | Metodologia | lezione frontale laboratorio |
| Spiegazione della teoria delle proiezioni ortogonali. Disegno delle proiezioni ortogonali di una scatola di caramelle su foglio da disegno. Replica dello stesso disegno con Processing. | | | |
| Elementi di programmazione: Ripasso degli elementi di programmazione appresi nella classe precedente. | | | |

| Unità di apprendimento #2 - Robotica! | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 3h di lezione in classe + 3h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | novembre |
| Discipline coinvolte | Tecnologia | Metodologia | lezione frontale laboratorio |
| Introduzione alla robotica: il linguaggio delle cose, cos'è un robot, sensori ed attuatori. Conoscere Arduino, la sua IDE, realizzare il primo esperimento: controllo del lampeggiamento di un led. | | | |
| Elementi di programmazione: IDE di Arduino, differenze e punti in comune con Processing, funzione setup e loop. | | | |

| Unità di apprendimento #3 - Controlliamo Arduino con Processing | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 3h di lezione in classe + 3h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | dicembre/gennaio |
| Discipline coinvolte | Tecnologia Scienze Matematica | Metodologia | lezione frontale laboratorio |
| Programmare Arduino con Processing per sfruttare le capacità grafiche di questo e realizzare un dispositivo capace di misurare la temperatura e riportarla in un grafico.. | | | |
| Elementi di programmazione: Concetto di funzione, concetto di libreria, import di una libreria con Processing e Arduino IDE, la libreria Firmata. | | | |

| Unità di apprendimento #4 - Allunaggio controllato | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 3h di lezione in classe + 3h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | gennaio/febbraio |
| Discipline coinvolte | Tecnologia | Metodologia | lezione frontale laboratorio |
| Riprendere l'unità di apprendimento "In movimento" realizzata in classe prima e sostituire il sistema di comando dell'astronave utilizzando quattro pulsanti montati su una breadboard controllati da Arduino. | | | |
| Competenze in gioco: Digitale | | | |

| Unità di apprendimento #5 - Interruttore crepuscolare | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 6h di lezione in classe + 6h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | marzo/aprile |
| Discipline coinvolte | Tecnologia Scienze Matematica | Metodologia | lezione frontale laboratorio |
| <p>Agenda 2030, cambiamenti climatici, efficientamento energetico. Realizzare un dispositivo capace di rilevare la quantità di luce e decidere di conseguenza se accendere o meno la luce elettrica. Ipotizzare il risparmio energetico ottenuto con l'adozione di un sistema del genere nella scuola.</p> | | | |
| Competenze in gioco: Digitale, Cittadinanza attiva | | | |

| Unità di apprendimento #6 - Termostato | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 6h di lezione in classe + 6h di esercitazione a casa | | |
| Spazi | lab. informatica | Periodo ipotetico | aprile/maggio |
| Discipline coinvolte | Tecnologia Scienze Matematica | Metodologia | lezione frontale laboratorio |
| <p>Agenda 2030, cambiamenti climatici, efficientamento energetico. Realizzare un dispositivo capace di rilevare la temperatura e decidere di conseguenza se accendere o meno il riscaldamento. Ipotizzare il risparmio energetico ottenuto con l'adozione di un sistema del genere nella scuola.</p> | | | |
| Competenze in gioco: Digitale, Cittadinanza attiva | | | |

Classe Terza

La terza classe della scuola secondaria di primo grado è quella in cui i ragazzi dovrebbero poter raggiungere la **competenza** ed essere capaci poi di metterla al servizio degli altri in uno spirito di **cittadinanza attiva**. E' dunque il momento in cui mettere a frutto tutte le conoscenze e le abilità acquisite e maturate nei due anni precedenti per realizzare un vero progetto (compito di realtà) che possa "lasciare un segno", ovvero essere realmente utile a qualcuno e non più un semplice esercizio fine a se stesso. Si propongono in tal senso un paio di idee come spunto di partenza ma si lascia alla fantasia del docente la scelta dell'attività che meglio si adatta alla classe e che possa essere utile alla comunità di appartenenza.

| Unità di apprendimento #1 - Un impianto di domotica a scuola | | | |
|--|--|--------------------------|---------------------------------|
| Tempi | 20h di lezione in classe + 20h di esercitazione a casa | | |
| Spazi | classe lab. informatica | Periodo ipotetico | primo quadrimestre |
| Discipline coinvolte | Tecnologia Scienze Matematica | Metodologia | lezione frontale laboratorio |
| Realizzare un impianto di domotica per la scuola costituito da una serie di dispositivi capaci di automatizzare alcune operazioni manuali. Ad esempio: campanella elettronica, interruttore crepuscolare, termostato, sismografo, indicatore di bagno occupato, apri-porta automatico, apertura/chiusura serrande automatico, ecc. ecc.. | | | |
| Competenze in gioco: Digitale, Cittadinanza attiva | | | |

| Unità di apprendimento #2 - Impianto di irrigazione automatico | | | |
|---|--|--------------------------|---------------------------------|
| Tempi | 20h di lezione in classe + 20h di esercitazione a casa | | |
| Spazi | classe lab. informatica | Periodo ipotetico | secondo quadrimestre |
| Discipline coinvolte | Tecnologia Scienze Matematica | Metodologia | lezione frontale laboratorio |
| Realizzare un impianto di irrigazione automatico per innaffiare una serra o l'orto della scuola. In base ai valori di umidità del terreno e di temperatura, l'impianto dovrà decidere se è necessario innaffiare ad un determinato orario prestabilito oppure no. | | | |
| Competenze in gioco: Digitale, Cittadinanza attiva | | | |

La valutazione

Aspetto importante quanto difficile è la valutazione delle attività svolte e dei livelli raggiunti dagli studenti. E' fondamentale in quanto è necessario che ciascun ragazzo abbia la giusta percezione di quanto si sia avvicinato al risultato più soddisfacente e quanto invece debba ancora fare per migliorare. Una valutazione eccessivamente avara potrebbe demoralizzare lo studente facendogli credere di essere incapace e, per questo motivo, desistere dall'obiettivo di raggiungere lo scopo che si è prefissato, tuttavia una valutazione eccessivamente generosa darebbe una falsa sensazione di averlo raggiunto e quindi di non aver nulla da migliorare, che va bene così, cosa invece non vera.

Ritengo che la valutazione in base decimale sia la più chiara e meno ambigua possibile, ed inoltre la ritengo estremamente pratica per arrivare al voto finale in pagella, per questo motivo è quella utilizzata nelle tabelle valutative riportate di seguito. Tuttavia, poiché diversi istituti scolastici utilizzano una valutazione per livelli (base, intermedio, ecc. ecc.) viene aggiunta, in coda, la corrispondenza di tali livelli con il voto in decimi.

Al fine di rendere i ragazzi consapevoli e, attraverso un processo metacognitivo, spingerli a *imparare ad imparare* si propone per prima cosa uno schema di autovalutazione da sottoporre agli studenti al termine di ogni unità di apprendimento.

Schema per l'autovalutazione

| Descrizione | Si | No |
|---|----|-------|
| Ho capito bene il compito assegnato? | | |
| Se no, ho chiesto ulteriori spiegazioni all'insegnante o ai miei compagni? | | Si No |
| Ho partecipato attivamente all'attività proposta, seguendo le istruzioni fornite dall'insegnante? | | |
| Sono riuscito a portare a termine il compito assegnato? | | |
| Il progetto realizzato funziona come dovrebbe? Come potevo contribuire al suo miglioramento? | | |
| Risposta: _____ | | |
| Ho collaborato con il mio gruppo di lavoro? | | |
| Nel gruppo di lavoro ho fatto proposte che si sono rivelate importanti per il buon esito del compito? | | |
| Ho contribuito ad apportare personalizzazioni e migliorie al compito assegnato? | | |

Schema per la valutazione

| Voto in decimi | Descrizione |
|----------------|---|
| 4 | L'alunno/a non si lascia coinvolgere nell'attività proposta, non inizia nemmeno il lavoro assegnato. |
| 5 | L'alunno/a si applica con grosse difficoltà nel compito assegnato non riuscendo a portarlo a termine. |
| 6 | L'alunno/a riesce a portare a termine in maniera essenziale anche se con qualche imprecisione il compito assegnato. Lavora in autonomia o partecipa in maniera passiva al gruppo di lavoro. |
| 7 | L'alunno/a porta a termine il compito assegnato in maniera essenziale ma corretta partecipando in maniera attiva al gruppo di lavoro. |
| 8 | L'alunno/a porta a termine il compito assegnato in maniera efficace partecipando in maniera attiva al gruppo di lavoro e proponendo soluzioni brillanti. |
| 9 | L'alunno/a dimostra di essere un punto di riferimento all'interno del gruppo di lavoro, porta a termine in maniera efficace ed efficiente il compito assegnato. |
| 10 | L'alunno/a dimostra di essere un punto di riferimento all'interno del gruppo di lavoro, porta a termine in maniera efficace ed efficiente il compito assegnato apportando personalizzazioni e migliorie rispetto alla consegna. |

Tabella di corrispondenza tra il voto in decimi e i livelli di valutazione

| Voto in decimi | Livello | Livello |
|----------------|---------|------------|
| 4 | D | Base |
| 5-6 | C | Iniziale |
| 7-8 | B | Intermedio |
| 9-10 | A | Avanzato |

Conclusioni

Quanto esposto in questo articolo vuole essere soprattutto una proposta per utilizzare Arduino nella scuola secondaria di primo grado cercando, attraverso un percorso didattico articolato nei tre anni, di superare le difficoltà nella programmazione che spesso emergono quando viene usata la nota scheda di prototipazione italiana. Difficoltà dovuta, naturalmente, alla mancanza di conoscenze nella programmazione dei ragazzi nella fascia di età dagli 11 ai 13 anni.

Tuttavia il percorso qui proposto è solo un'idea progettuale e quindi, per poter affermare di essere efficace e, soprattutto, utile al processo di apprendimento degli alunni, ha bisogno di essere sperimentato e modificato in base alle diverse esigenze e alle diverse risposte che si hanno nelle varie classi. Per questo motivo sarò ben lieto di ricevere commenti e suggerimenti di chi vorrà tentare questa strada al mio indirizzo e-mail giovanni.cielo@ic1chieti.it.



Giovanni Cielo

giovanni.cielo@ic1chieti.it - giovanni.cielo.ch@gmail.com

Docente di tecnologia presso la scuola secondaria di primo grado dell'Istituto Comprensivo n. 1 di Chieti. Ingegnere elettronico con esperienza di progettazione, e realizzazione di sistemi informativi ha trovato nella robotica educativa l'incontro dei suoi interessi professionali: l'insegnamento, l'informatica e l'elettronica. Si interessa di trovare strategie didattiche che aiutino l'apprendimento sviluppando il pensiero computazionale attraverso attività di coding e di robotica educativa.