



**Modulo Progettazione tecnologico-didattica**  
**Documento di progetto**  
**Modulo Progettazione tecnologico-didattica**  
**Documento di progetto**

**Grazia Raitano**

Gruppo di progetto Animatori Digitali:

M.Maggiordomo, G. De Amicis, M. Vaccaro, S. Pipitone, S. Cerami, G. Raitano

<b>Titolo</b>	<b>START WHIT CODING</b>
<b>Abstract</b>	<p>Il percorso didattico risponde all'esigenza di promuovere l'educazione al pensiero computazionale fin dalla scuola del Primo Ciclo, così come previsto dal Piano Nazionale Scuola Digitale. Le piattaforme dedicate al pensiero computazionale, come Programma il Futuro e Code.org, rendono disponibili percorsi di apprendimento destinati ad alunni di ogni fascia d'età.</p> <p>Presentano attività strutturate, chiare e che possono essere svolte dagli alunni mentre contestualmente costituiscono uno stimolo ed uno strumento di auto-aggiornamento per i docenti, sia da un punto di vista strettamente connesso con il coding, sia da un punto di vista di acquisizione di generali competenze digitali</p> <p>Il progetto trova legittimazione da quanto riportato nelle Indicazioni Nazionali che esplicita: "La diffusione delle tecnologie...è una grande opportunità e rappresenta la frontiera decisiva per la scuola. Si tratta di una rivoluzione epocale..."</p> <p>Per la disciplina della tecnologia si precisa che "Quando possibile, gli alunni potranno essere introdotti ad alcuni linguaggi di programmazione particolarmente semplici e versatili che si prestano a sviluppare il gusto per l'ideazione e la realizzazione</p> <p>di progetti e per la comprensione del rapporto tra codice sorgente e risultato visibile."</p> <p>Nell'educazione formale le competenze digitali si sviluppano attraverso attività trasversali che interessano tutte le discipline insegnate. Lo sviluppo del pensiero computazionale si coniuga con l'attivazione delle capacità logiche e cognitive valide in ogni area disciplinare.</p>

	Lo sviluppo del pensiero computazionale si realizza attraverso la progettazione e la pianificazione di attività che investono ogni disciplina.
<b>Parole chiave</b>	<ul style="list-style-type: none"> <li>• Interazione: come capacità di sperimentare l'ambiente con un approccio di problem-solving</li> <li>• Performance: come abilità di adottare prospettive alternative al fine di improvvisare e scoprire</li> <li>• Cognizione distribuita: come capacità di interagire in modo significativo con gli strumenti ed espandere le proprie capacità mentali</li> <li>• Intelligenza distribuita: intesa come capacità di mettere insieme l'informazione e mettere a confronto i vari punti di vista nell'ottica di un comune obiettivo</li> <li>• Multitasking: intesa come capacità di scansionare il proprio ambiente mantenendo più fuochi di attenzione.</li> </ul> <p>Gli approcci pedagogici fanno riferimento a modelli di tipo cognitivo-costruttivista di insegnamento.</p>
<b>Docenti</b>	
<b>Nome/i docenti coinvolti e disciplina</b>	Grazia Raitano (docente di sostegno e AD), Rita Greco (matematica), Rosa Cuccia (Italiano), Caterina Marchisello (tecnologia).
<b>Studenti</b>	
<b>Numero</b>	44
<b>Età</b>	9
<b>Genere</b>	maschi e femmine
<b>Tipologia di scuola</b>	Direzione didattica
<b>Caratteristiche specifiche</b>	21 femmine di cui due diversamente abili, 23 maschi

<b>Finalità generali</b>
Promuovere e sviluppare la creazione di criteri logici per "leggere" la realtà e risolverne i problemi.

<b>Conoscenze</b>	Il linguaggio della programmazione a blocchi L'utilizzo di strumenti e materiali per progettare e costruire un sistema che risolve un problema specifico I dati per analizzare e valutare un problema
<b>Capacità e processi cognitivi</b>	Riconoscere il senso di una informazione e saperla trasformare Conoscere, generare e confrontare le possibili soluzioni ad un problema che soddisfano i criteri e i vincoli del problema stesso. Utilizzare materiale conosciuto per risolvere problemi nuovi Riunire elementi al fine di formare una nuova struttura organizzata e coerente
<b>Organizzazione</b>	
<b>Durata</b>	16 ore
<b>In presenza</b>	4 + 1 ora per la verifica
<b>On-line</b>	6 ore
<b>A casa (se previsto)</b>	5 ore
<b>Strumenti tecnici</b>	LIM, PC, notebook, piattaforma web code.org .

<b>Sviluppo dei contenuti</b>	
<b>Descrizione</b>	Utilizzo della risorsa code.org
<b>Prerequisiti</b>	Leggere, conoscere le principali figure geometriche, utilizzare Il mouse e/o Il touch pad, collaborare con i compagni, comprendere il significato di intero, metà, un quarto
<b>Struttura</b>	Una Unità di Apprendimento organizzata in 7 sequenze

<b>Unità A / Algoritmo</b>	
<b>Argomento</b>	Comprendere il significato di algoritmo come serie di istruzioni

<b>Valutazione</b>	Scomporre un'azione complessa in sequenze semplici e rappresentarle in algoritmi
--------------------	--

<b>Sequenza A-1</b>	<b>Algoritmi e Programmazione</b>
<b>Obiettivi specifici</b>	<ul style="list-style-type: none"> <li>• Comprendere la difficoltà di tradurre problemi reali in programmi</li> <li>• Comunicare idee attraverso codici e simboli</li> <li>• Scomporre attività complesse in una serie di eventi minori</li> <li>• Organizzare gli eventi in sequenza nel loro ordine logico</li> </ul>
<b>Tempi e modalità</b>	In classe: 2 ore. A casa: 1 ora circa per produrre un disegno da proporre:
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, cooperative learning, attività di gruppo
<b>Contenuti</b>	Algoritmi : un elenco di step successivi per realizzare un compito Programmazione : il codice di un algoritmo che può essere letto da una macchina
<b>Risorse</b>	Schede didattiche strutturate dal sito code.org, carta quadrettata, matite, colori
<b>Supporto tecnico</b>	Notebook, LIM, stampante

<b>Unità B / Introduzione al coding</b>	
<b>Argomento</b>	Comprendere che programmi e algoritmi devono essere espressi in maniera semplice, chiara e precisa e servono per dare istruzioni ad una macchina secondo l'obiettivo voluto e che: dall'algoritmo alla sequenza di istruzioni
<b>Valutazione</b>	Comprendere e completare un algoritmo Individuare un errore in un algoritmo

<b>Sequenza B-1</b>	<b>Le sequenze: Angry Birds</b>
<b>Obiettivi specifici</b>	<ul style="list-style-type: none"> <li>• Esprimere il movimento come una serie di comandi</li> <li>• Ordinare i comandi di un movimento come passaggi sequenziali in un programma</li> <li>• Rappresentare un algoritmo come un programma per computer</li> <li>• Contare il numero di volte che deve essere eseguita un'azione e rappresentarla come istruzioni in un programma</li> <li>• Ricordare e applicare in coppia le regole di programmazione</li> <li>• Identificare situazioni in cui le regole di programmazione non sono</li> </ul>

	rispettate
<b>Tempi e modalità</b>	n. 1 ora in classe
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, collaborative learning, cooperative learning, e-learning, lavoro individuale, a coppie e/o di gruppo
<b>Contenuti</b>	Algoritmi sequenziali, concetti topologici, blocchi di programma, gli attributi di triangoli, quadrati e rettangoli; concetti di: la metà, il quarto
<b>Risorse</b>	Corso n. 2 code.org lezioni n. 3
<b>Supporto tecnico</b>	Computer, LIM, notebook

<b>Sequenza B-2</b>	<b>Le sequenze: Artista</b>
<b>Obiettivi specifici</b>	<ul style="list-style-type: none"> <li>• Creare un programma per completare un'immagine utilizzando passi sequenziali</li> <li>• Selezionare un argomento per un determinato comando</li> <li>• Distinguere tra ciò che definisce e non definisce gli attributi di triangoli, quadrati e rettangoli e disegnarli</li> <li>• Spiegare la differenza tra quadrati e rettangoli ed usare comandi per disegnare le diverse forme</li> <li>• Confrontare e contrapporre quadrati e rettangoli in base al numero di lati ed alle loro lunghezze</li> <li>• Comporre forme bidimensionali (rettangoli, quadrati, trapezi, triangoli) per creare una forma composta, come ad esempio due quadrati per comporre un rettangolo e due rettangoli per comporre un quadrato</li> <li>• Comporre forme nuove dalle forme composte</li> <li>• Disegnare le partizioni in un rettangolo e descrivere le partizioni usando le parole metà, quarto, quarti, la metà di un quarto e quarti di...</li> </ul>
<b>Tempi e modalità</b>	n. 2 ore in classe n. 1 ora a casa per lavoro individuale e/o in coppia
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, collaborative learning, cooperative learning, e-learning, lavoro individuale, a coppie e/o di gruppo
<b>Contenuti</b>	Algoritmi sequenziali, forme geometriche e loro attributi, gli attributi di triangoli, quadrati e rettangoli; la metà, il quarto

<b>Risorse</b>	Blocchi logici, carta a quadretti, corso n. 2 code.org lezione n. 4
<b>Supporto tecnico</b>	Computer, LIM , notebook

### Unità C / Semplificare le cose

<b>Argomento</b>	Comprendere che programmi e algoritmi servono per dare istruzioni ad una macchina secondo l'obiettivo voluto e che devono essere espressi in maniera semplice, chiara e precisa: dall'algoritmo alla sequenza di istruzioni
<b>Valutazione</b>	Debugging-correzione di errori: Individuare un errore in un programma Modificare un programma esistente per risolvere gli errori Identificare un algoritmo errato in base al disordine dei passaggi Identificare le diverse strategie utilizzate e condividere con l'intera classe.
<b>Sequenza C-1</b>	<b>Ripetizioni</b>
<b>Obiettivi specifici</b>	<ul style="list-style-type: none"> <li>• Ripetere una serie di azioni avviate dal docente</li> <li>• Tradurre un programma disegnato in un movimento fluido</li> <li>• Convertire una serie di azioni multiple in un singolo ciclo</li> <li>• Identificare i vantaggi di utilizzare una struttura ciclica invece della ripetizione manuale</li> </ul>
<b>Tempi e modalità</b>	n. 1 ora in classe
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, collaborative learning, cooperative learning, e-learning, lavoro individuale e/o a coppie
<b>Contenuti</b>	La ripetizione di un'istruzione, la struttura ciclica, scomposizione e sintesi di istruzioni
<b>Risorse</b>	Corso n. 2 code.org lezione n. 5
<b>Supporto tecnico</b>	Computer, LIM, stampante

<b>Sequenza C-2</b>	<b>Cicli: Labirinto</b>
<b>Obiettivi specifici</b>	<p>Identificare i vantaggi di utilizzare una struttura ciclica invece della ripetizione manuale</p> <p>Creare un programma per un determinato compito che ripete un singolo comando</p> <p>Suddividere una lunga sequenza di istruzioni nel minor numero di sequenze ripetibili e possibili</p> <p>Creare un programma per un determinato compito che ripete una</p>

	sequenza di comandi Impiegare una combinazione di comandi sequenziali e ciclica per raggiungere la fine di un labirinto
<b>Tempi e modalità</b>	n. 1 ora in classe n. 1 ora a casa lavoro individuale e/o in coppia
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, collaborative learning, cooperative learning, e-learning, lavoro individuale e/o a coppie
<b>Contenuti</b>	Le sequenze elementari di una forma, la ripetizione di un comando per realizzare una forma complessa
<b>Risorse</b>	Corso n. 2 code.org lezione n. 6
<b>Supporto tecnico</b>	Computer, LIM , notebook

<b>Sequenza C-3</b>	<b>Cicli: Artista, Ape</b>
<b>Obiettivi specifici</b>	Contare il numero di volte che un azione dovrebbe essere ripetuta e rappresentarlo come un ciclo Scomporre una forma nella sua più piccola sequenza ripetibile Creare un programma che disegna forme complesse ripetendo le sequenze semplici  Scrivere un programma per un determinato compito che ripete un singolo comando Identificare quando un ciclo può essere utilizzato per semplificare un'azione ripetitiva Impiegare una combinazione di sequenze e comandi ripetuti ciclicamente per eseguire azioni
<b>Tempi e modalità</b>	n. 2 ore in classe n. 2 ore a casa per lavoro individuale e/o in coppia
<b>Strategie didattiche</b>	Apprendimento tra pari, problem solving, collaborative learning, cooperative learning, e-learning, lavoro individuale e/o a coppie
<b>Contenuti</b>	Le sequenze elementari di una forma, la ripetizione di un comando per realizzare una forma complessa
<b>Risorse</b>	Corso n. 2 code.org lezioni n. 7 e 8
<b>Supporto tecnico</b>	Computer, LIM, notebook

RISORSE E RISULTATI

<b>Risorse interne</b>	Computer, LIM e/o videoproiettore, stampante, fotocopiatrice, materiale di facile consumo, schede didattiche strutturate, palestra e attrezzi sportivi (mattoncini, cerchi, coni, ...).
<b>Risorse esterne</b>	Sito <a href="http://www.code.org">www.code.org</a>
<b>Risultati</b>	Gli alunni alla fine del percorso riconoscono il linguaggio della programmazione a blocchi, analizzano e valutano una situazione problematica e utilizzano strumenti e materiali per costruire un sistema