



Istituto Comprensivo  
“Giovanni Pascoli”  
Silvi - TE



# CURRICOLO VERTICALE CODING E ROBOTICA EDUCATIVA

---

SEZIONE POFT



## *Introduzione*

Il Curricolo è un percorso formativo, con traguardi da raggiungere strada facendo, che occorre adeguatamente pianificare. Si deve tener conto di obblighi dati dal raggiungimento di obiettivi e di strategie didattiche per l'apprendimento nelle varie età. La progressione didattica relativa al Coding, che sta a noi qui definire nei vari livelli di passaggio da un ordine di scuola all'altro, va sviluppata in verticale dalla più tenera infanzia fino all'Università e anche oltre, in riferimento all'apprendimento lungo tutto l'arco della vita (lifelong learning). La diffusione del Coding si espande in un compendio di progettazione e pianificazione trans-curricolare che investe tutte le discipline.

Il Coding, che può essere oggi insegnato attraverso piattaforme disponibili come code.org, va studiato e capito non tanto per formare dei professionisti della programmazione, bensì perché la conoscenza dei fondamenti contribuisce a formare il bagaglio tecnico scientifico e culturale di ogni persona. Esso, pertanto, assume nell'insegnamento una duplice funzione: da un lato ha un ruolo culturale formativo di base sul piano scientifico, accompagnando la Matematica e le altre scienze e dall'altro quello di strumento trasversale a tutte le discipline che favorisce lo sviluppo logico del pensiero, un approccio curioso di fronte alla realtà e la capacità di provare a risolvere i problemi o di ripartire dagli errori o dagli ostacoli incontrati nei processi formativi.

Questo vale per ogni ordine e grado di scuola. Ecco dunque le ragioni del presente curricolo verticale rivolta a insegnanti e studenti dell'Istituto Comprensivo "G. Pascoli" di Silvi.

## *Il pensiero computazionale come competenza trasversale*

Il termine "**pensiero computazionale**", dopo essere stato ufficialmente accettato nel mondo giuridico con la pubblicazione della legge 107/2015 ("La Buona Scuola") è entrato anche nella pratica didattica col Piano Nazionale Scuola Digitale (PNSD), che ha riconosciuto attività di questo tipo come **essenziali per la formazione** degli studenti nell'era digitale.

Linguaggio naturale e matematica sono, giustamente, insegnati fin dal primo anno delle elementari, perché costituiscono delle abilità "trasversali", cioè utili e applicate in qualunque altra materia. Per riuscire bene in storia, geografia, scienze, arte, il linguaggio naturale è indispensabile, altrimenti non riusciremmo a spiegare e descrivere i soggetti che stiamo trattando, ed è altrettanto necessaria la matematica, pena l'impossibilità di confrontare quantità, stabilire relazioni numeriche, ordinare valori.

Ma non per questo si insegnano il "pensiero linguistico" o il "pensiero matematico" come disciplina ortogonale a tutte le scienze. Si insegnano, giustamente, "italiano" e "matematica", e poi le competenze linguistiche e matematiche vengono naturalmente travasate e messe in gioco quando si parla – ad esempio – di arte o di storia.

Perché allora parliamo di "pensiero computazionale" nella scuola?

Quando parliamo di pensiero computazionale, vogliamo esplicitamente parlare degli aspetti culturali e scientifici dell'informatica, a prescindere da qualunque aspetto strumentale o tecnologico. In secondo luogo, riteniamo che l'informatica abbia una caratteristica, nuova e sua specifica, che la rende ancora più interessante in un'ottica interdisciplinare rispetto alla lingua naturale e alla matematica.

La capacità, unica dell'informatica, di concretizzare mondi virtuali, grazie alle sue capacità apparentemente illimitate di elaborare simboli di qualunque tipo ed in qualunque modo, costituisce la sua forza in un contesto educativo. Perché niente è utile per apprendere un concetto come farne esperienza concreta. E con l'informatica questo è possibile, in qualunque disciplina. Questo è il motivo per cui parliamo di pensiero computazionale nella scuola. Ed è questo il motivo per cui uno degli strumenti per sviluppare al meglio questa competenza è certamente il coding.

Il coding però viene spesso associato solo ed esclusivamente alla tecnologia e più strettamente all'informatica. Andrebbe invece riscoperta proprio la sua radice terminologica, codificare (to code). Il coding non dovrebbe essere visto solo come attività laboratoriale dell'ora di tecnologia, ma come approccio all'analisi, alla logica e alla codifica di tutte quelle informazioni che fanno parte di un processo formativo. D'altra parte il coding non dovrebbe essere il tutto, bensì uno strumento che accompagna l'allievo nel maturare il proprio pensiero logico nel percorso di crescita quotidiano.

Un approccio metodologico che aiuti lo studente ad apprendere e sviluppare le capacità di analisi, di scomposizione di un dato problema, di comprendere le ragioni di un determinato fattore e di poter elaborare delle soluzioni.

Pensare di relegare il coding al solo ambito informatico è riduttivo e semplicistico. Strumenti come Scratch, CodyRoby, PixelArt possono essere utilizzati in vari percorsi formativi, con contenuti didattici non prettamente legati all'informatica. Non solo, possono essere di esempio per sviluppare nuovi progetti come ad esempio CodyWay, un interessante strumento che prende spunto dal linguaggio a blocchi applicandolo all'ambiente quotidiano che ci circonda.

Un altro strumento molto efficace allo sviluppo del pensiero computazionale è certamente la robotica educativa. Essa è la concretizzazione di quanto gli alunni "progettano" con il coding. Attraverso infatti strumenti pensati appositamente per la didattica (Bee Bot, Blue Bot, Lego Mind, Arduino ecc) si può completare il percorso formativo iniziato con il coding, utilizzando gli algoritmi per "programmare" robot o altre entità tecnologiche, così da realizzare una vera e propria intelligenza artificiale.

La robotica educativa spesso produce risultati ancora migliori rispetto al coding, poiché risulta essere per gli alunni meno astratta e dunque più intuitiva per l'ideazione e l'applicazione degli algoritmi e il conseguente sviluppo del pensiero computazionale.

## Scuola dell'infanzia

Nel mondo attuale, dove ci si collega a internet da ovunque vogliamo e con diversi strumenti, è importante la cultura della fruizione non passiva; in questo ci viene in aiuto il pensiero computazionale.

Il pensiero computazionale è il riuscire a mettere in pratica, in una serie di azioni (singole, precise e consequenziali), un'idea che abbiamo avuto per risolvere un determinato problema. Attraverso il *coding* si cerca di trasmettere ai cosiddetti "nativi digitali" sia il linguaggio della programmazione e degli algoritmi (mettendo in primo piano la logica che sta dietro alla tecnologia) sia la potenza della segmentazione di un problema e il loro affidamento a piccoli gruppi. Così si dà forza al lavoro collaborativo, al rispetto delle diverse idee.

La costruzione di questa competenze può contribuire a favorire lo sviluppo di elementi di progettualità anche in ambiti disciplinari diversi da quello informatico. *Coding* e pensiero computazionale rendono possibile apprendere strategie di risoluzione di problemi, progettazione e comunicazione anche a chi non programmerà mai. Grazie alle attività che si possono svolgere nel progetto "Programma Il Futuro", consistenti in esercizi "guidati" sul pensiero computazionale (svolti sia in un ambiente tecnologico, con l'utilizzo del computer e della rete internet sia in modalità "unplugged"), alle attività proposte in Code Week e la conoscenza di altri ambienti di programmazione visuale (non è necessario scrivere i comandi, le "stringhe" di codice, ma bisogna mettere in sequenza logica le raffigurazioni grafiche che li rappresentano) come Blockly è possibile acquisire alcune capacità come:

- capire cos'è un algoritmo: facendo scoprire ai bambini che sono algoritmi alcuni dei modi di operare, nella vita di tutti i giorni o a scuola, che realizziamo (quasi) automaticamente;
- usare il ragionamento logico per spiegare il funzionamento di alcuni semplici algoritmi;
- capire i principi alla base del funzionamento di un computer;
- selezionare, trasportare e lasciare: attività di *drag and drop*.

Completa l'offerta formativa l'utilizzo di Bee-Bot e Blue-Bot.

Anno	Competenze	Contenuti	Strumenti	Motivazione	Metodologia
Dai 4 ai 5 anni	Giocare con gli strumenti tecnologici e non, con un approccio aperto alla curiosità e all'esplorazione	<ul style="list-style-type: none"><li>• Giochi di esplorazione dell'ambiente</li><li>• Giochi di movimento su grandi scacchiere</li><li>• Muovere giocattoli /oggetti sulle scacchiere</li><li>• Gli strumenti digitali (pc, tablet, Lim, Smart Toys)</li></ul>	<ul style="list-style-type: none"><li>• Scacchiere di piccole e grandi dimensioni</li><li>• Cody &amp; Roby;</li><li>• Kodable</li><li>• Bits and Bricks</li><li>• Bee Bot e Blue bot</li></ul>	<p>La curiosità, l'azione diretta, il gioco.</p> <p>Utilizzare le TIC come strumento di sviluppo dell'apprendimento, per permettere ai bambini di imparare attraverso una pluralità di canali.</p>	<p>Favorire la curiosità, la scoperta, l'esplorazione concreta, il gioco, il procedere per tentativi, la collaborazione, la riflessione sulle esperienze</p> <p>Learning by doing</p> <p>Cooperative Learning</p>

## Scuola Primaria

Il Coding è una prima forma di approccio interdisciplinare alle TIC: consente l'avvio all'uso consapevole del computer per comprendere che le dotazioni tecnologiche sono strumenti attraverso i quali realizzare dei progetti; sviluppa il pensiero riflessivo e procedurale; stimola la riflessione sull'errore come nuovo spunto di lavoro; sviluppa delle capacità di analisi sul proprio operato; incrementa delle capacità di espressione linguistica sia orale che scritta per comunicare il proprio operato agli altri o come memoria personale (relazione fasi attività, documento di sintesi del lavoro, etc); garantisce un utilizzo diretto di conoscenze matematiche, linguistiche, antropologiche e scientifiche per sostanziare di contenuti gli elaborati prodotti; sviluppa il lavoro cooperativo e delle abilità individuali.

Nella scuola primaria l'obiettivo è di sviluppare tutte queste competenze e capacità attraverso lo studio dei principali costrutti del coding (iterazione, istruzioni condizionali, funzioni).

L'utilizzo di strumenti didattici a difficoltà progressiva, come ad esempio il portale code.org è fortemente consigliato, sia per le attività al pc sia per quelle unplugged. Oltre all'utilizzo di Bee-Bot e Blue-Bot, completa l'offerta formativa l'utilizzo di Lego WeDo.

PERIODO	METODOLOGIA	STRUMENTI	VERIFICHE
intero corso di studi	Attività di approccio mediato dal docente, Learning by doing; Cooperative Learning  Didattica dell'errore (debug); tutoring	Code.org  ScratchJr (eventualmente abbinato con Lego WeDo) Blockly  Strumenti unplugged: CodyRoby, Codyway, PixelArt  Bee-Bot e Blue-Bot	Gli stessi prodotti realizzati
<b>Raccordi con la Scuola dell'Infanzia</b>			
<ul style="list-style-type: none"> <li>• Conoscenza dei blocchi "Vai avanti", "Gira a destra"; "Gira a sinistra";</li> <li>• Usare il ragionamento logico per spiegare il funzionamento di alcuni semplici algoritmi;</li> <li>• Attività unplugged con CodyRoby e Codyway</li> <li>• Code org. Corso 1</li> <li>• Bee bot e Blue bot</li> </ul>			
<b>Raccordi con la Scuola Secondaria di I Grado</b>			
<ul style="list-style-type: none"> <li>• Conoscenza dei blocchi "Vai avanti", "Gira a destra"; "Gira a sinistra"; "Ripeti"; "Se – Allora – Altrimenti"</li> <li>• Code.org Corso 2 e 3</li> <li>• Ambienti editor Scratch: sprite, stage, costume e script</li> <li>• modifica sprite con editor grafico, inizializzazione di un progetto</li> </ul>			
<b>Raccordi con le discipline scolastiche (interdisciplinarietà)</b>			
Capacità di comunicazione orale e scritta, applicazione delle conoscenze curriculari apprese nelle varie classi negli ambiti linguistici, matematici, scientifici, tecnologici, antropologici, etc, per la pianificazione e elaborazione dei prodotti			
Sviluppare capacità di Problem Solving. Collaborare e interagire con gli altri per giungere alla soluzione di un			

problema; esplorare varie forme di narrazione digitale, animazioni e creazioni di videogiochi.

Competenze	Abilità	Conoscenze
<p>Sapersi esprimere e comunicare utilizzando codici e linguaggi diversi</p> <p>L'alunno utilizza con consapevolezza i concetti di ripetizione e condizione</p> <p>Analizzare e rappresentare processi utilizzando modelli logici</p> <p>Approcciarsi ad un uso consapevole delle TIC</p> <p>Utilizzare trasversalmente le conoscenze</p>	<p><b>COSTRUIRE ABILITA' DI BASE</b></p> <ul style="list-style-type: none"> <li>● Indicare la destra e la sinistra rispetto al personaggio da muovere sullo schermo</li> <li>● Imparare a scrivere i comandi in ordine e risolvere i problemi utilizzando una serie di situazioni.</li> <li>● Utilizzare il blocco di ripetizione.</li> <li>● Riconoscere in una dichiarazione logica sia l'ipotesi che la conclusione. Utilizzare le espressioni condizionali se/allora</li> <li>● Applicare il concetto di funzione nella soluzione di schemi logici</li> </ul> <p><b>LEGGERE E SCRIVERE IL CODICE</b></p> <ul style="list-style-type: none"> <li>● Identificare istruzioni condizionali.</li> <li>● Creare istruzioni condizionali.</li> <li>● Utilizzare espressioni condizionali, cicli, funzione, ripetizione</li> <li>● Scrivere codice conciso</li> <li>● Apportare modifiche al codice</li> <li>● Identificare gli errori nel codice e dedurre soluzioni corrette.</li> <li>● Progettare e costruire storie tecnologiche unendo le potenzialità del kit LEGO WeDo</li> <li>● Stabilire relazioni causa-effetto</li> <li>● Creare semplici programmi per istruire i modelli LEGO</li> </ul> <p><b>PROBLEM SOLVING</b></p> <ul style="list-style-type: none"> <li>● Logicamente applicare la soluzione più efficace in problemi che possono essere risolti in molteplici modi.</li> <li>● Saper esprimere i propri punti di vista e le proprie proposte</li> <li>● Saper aprire un programma, utilizzarlo, salvare il lavoro, riaprirlo</li> <li>● Comprendere che la macchina è un mezzo e non un fine del lavoro</li> <li>● Utilizzare le conoscenze disciplinari per riempire di contenuti le attività: realizzare stringhe di testo di vario tipo; creare ambientazioni coerenti negli elementi</li> </ul>	<p>Imparare cos'è un ciclo, quando usarlo.</p> <p>Conoscere le espressioni condizionali se/allora</p> <p>Conoscere il concetto logico di funzione</p> <p>Conoscere il costrutto di ripetizione</p> <p>conoscere le parti di un device</p> <p>Conoscere l'ambiente di programmazione visuale Scratch Jr</p> <p>Conoscere un editor grafico (come ad es. quello per editare gli sprite in Scratch)</p> <p>Conoscere le componenti del kit LEGO WeDo e maneggiarle con disinvoltura, sensori e motori</p>

	caratterizzanti; conoscere le principali regole grafiche nella realizzazione di opere ( figura in primo piano, sfondo, orizzonte, cromia etc)	
--	---	--

<b>Contenuti</b>		
------------------	--	--

- |   |  |  |
|---|--|--|
| <ul style="list-style-type: none"><li>- Concetto di programmazione (programmatore, linguaggio di programmazione, sequenza)</li><li>- L'ora del codice</li><li>- Code.org Corso 1 e 2, per le prime tre classi della primaria</li><li>- Code.org Corso 3 per le ultime classi della primaria</li><li>- Scratch Jr, per le prime tre classi della primaria</li><li>- Scratch (avvio al suo utilizzo): inizializzazione di un progetto (posizione di partenza del gattino), sprite, stage, costumi e sfondi</li><li>- Modifica sprite con editor grafico</li><li>- Codi-Amo</li><li>- Code.org PlayLab costruire un gioco, una storia.</li><li>- Sequenza e Algoritmi, espressioni condizionali, cicli</li><li>- La funzione</li><li>- La ripetizione</li><li>- PixelArt</li><li>- Bee Bot e Blue Bot , per le prime tre classi della primaria</li><li>- Lego WeDo, per le ultime classi della primaria</li><li>- Gioco, storytelling, grafica computerizzata: dialoghi e interazione tra personaggi per le animazioni, creazione dei propri personaggi con editor grafico; uso della interattività e del movimento per la realizzazione di giochi multimediali; modalità per creare disegni geometrici con il codice, uso del suono.</li><li>- Altre esperienze di programmazione orientata sugli oggetti</li></ul> |  |  |
|---|--|--|

SCUOLA PRIMARIA		
	COMPETENZE	ATTIVITA'
CLASSE PRIMA		
CLASSE SECONDA		
CLASSE TERZA		
CLASSE QUARTA		
CLASSE QUINTA		
SCUOLA SECONDARIA		
	COMPETENZE	ATTIVITA'
CLASSE PRIMA		
CLASSE SECONDA		
CLASSE TERZA		



# Scuola Secondaria di Primo Grado

Il **pensiero computazionale** è un processo mentale per far risolvere problemi ad un agente, sia esso persona o macchina, fornendogli una serie di istruzioni che deve eseguire in autonomia.

Nella scuola secondaria di primo grado l'obiettivo è di sviluppare il pensiero computazionale, attraverso l'utilizzo dei costrutti più complessi della programmazione, per la realizzazione di algoritmi ottimizzati ed efficaci alla risoluzione di problemi, e attraverso la robotica educativa.

Ovviamente, come per la matematica alle medie non si insegna il calcolo differenziale (e non lo si fa neanche alle superiori) così per l'informatica bisogna insegnare i concetti di base in modo adatto allo specifico livello di maturazione degli studenti, ed in modo indipendente dalla tecnologia.

L'utilizzo di strumenti didattici a difficoltà progressiva, come ad esempio il portale code.org è indispensabile ad introdurre alcuni concetti piuttosto astratti, come ad esempio le funzioni con parametri. La prosecuzione delle attività con ambienti di programmazione visuali quali scratch o App Inventor consentono agli alunni di ideare e realizzare qualcosa di proprio, come ad esempio un videogioco.

L'utilizzo in fine di supporti legati alla robotica educativa, consentono di ridurre il livello di astrazione e di applicare gli algoritmi realizzati alla programmazione di oggetti che interagiscono con il mondo reale (intelligenza delle cose).

PERIODO	METODOLOGIA	VERIFICHE
L'intero triennio	Learning by doing; Cooperative Learning	I prodotti finali
<b>Raccordi con la Scuola Primaria</b>		
<ul style="list-style-type: none"><li>● Conoscenza dei blocchi “Vai avanti”, “Gira a destra”; “Gira a sinistra”; “Ripeti”; “Se – Allora – Altrimenti”</li><li>● L’Ora del Codice di Programma il Futuro;</li><li>● Code.org Corso 2 e 3</li><li>● Ambiente editor Scratch Jr</li><li>● Ambiente sviluppo Scratch: sprite, stage, costume e script modifica sprite con editor grafico inizializzazione di un progetto</li></ul>		
<b>Raccordi con le discipline scolastiche (interdisciplinarietà)</b>		
Matematica e Geometria, Tecnologia (disegno tecnico)		
Problem solving, logica e la capacità di trasformare le proprie idee in App e Videogiochi come focus della programmazione. Utilizzando i più conosciuti linguaggi di programmazione per bambini e ragazzi – quali Scratch e App Inventor) – i docenti aiuteranno i giovani sviluppatori a muovere i primi passi nel coding.		
Italiano e Lingue straniere		
Realizzazione di Storytelling con ambienti di sviluppo quali Scratch, realizzazione di app in lingua straniera		
Musica		
Realizzazione di programmi multimediali		

Competenze	Abilità	Conoscenze
<p>Conoscere il concetto di algoritmo nelle sue varie forme ed applicazioni</p> <p>Approfondire i concetti di variabile e di funzione con parametri</p> <p>Approcciarsi ad una scrittura ottimizzata del codice</p> <p>Conoscere le caratteristiche di un robot</p> <p>Utilizzare trasversalmente le conoscenze</p>	<p><b>COSTRUIRE ABILITA' DI BASE</b></p> <ul style="list-style-type: none"> <li>● Imparare a scrivere algoritmi in maniera corretta e a risolvere problemi complessi scomponendoli in problemi più semplici.</li> <li>● Applicare il concetto di variabile nella soluzione di schemi logici</li> <li>● Utilizzare correttamente le funzioni con e senza parametri</li> </ul> <p><b>LEGGERE E SCRIVERE IL CODICE</b></p> <ul style="list-style-type: none"> <li>● Debugging</li> <li>● Scrivere codice ottimizzato</li> <li>● Apportare modifiche al codice in modo da ottimizzare lo stesso pur conservandone la funzionalità</li> <li>● Identificare i bug nel codice</li> <li>● Progettare e costruire storie tecnologiche unendo con il linguaggio di programmazione Scratch.</li> <li>● Remixare programmi esistenti</li> <li>● Realizzare un Videogioco</li> </ul> <p><b>COSTRUIRE E PROGRAMMARE UN ROBOT</b></p> <ul style="list-style-type: none"> <li>● Capire cosa sono e come usare sensori e motori per rendere interattivi i modelli</li> <li>● Stabilire relazioni causa-effetto</li> <li>● Creare semplici programmi per istruire i modelli LEGO</li> </ul> <p><b>PROBLEM SOLVING</b></p> <ul style="list-style-type: none"> <li>● Saper lavorare in gruppo.</li> <li>● Ottimizzare la soluzione di un problema</li> <li>● Scomporre un problema complesso in problemi di complessità inferiore</li> <li>● Utilizzare registri linguistici adatti agli interlocutori</li> <li>● Saper realizzare un algoritmo utile per la soluzione di un problema generico</li> <li>● Utilizzare le conoscenze disciplinari per riempire di contenuti le attività: realizzare storytelling; conoscere le principali regole nella realizzazione di figure geometriche</li> </ul>	<p>Conoscere il concetto logico di variabile</p> <p>Conosce il corretto utilizzo delle funzioni nella realizzazione di un algoritmo</p> <p>Conosce il concetto di parametro ed il suo utilizzo all'interno delle funzioni</p> <p>Conoscere le principali funzioni di un programma ed utilizzarle in modo congruo</p> <p>Conoscere l'ambiente di programmazione Scratch</p> <p>Conoscere l'ambiente di sviluppo App Inventor</p> <p>Conoscere le caratteristiche di un robot (sensori e motori)</p> <p>Conoscere l'ambiente di progettazione Lego Mindstorm</p>

	anche di complessità elevata (es. circonferenza); realizzare programmi di musica ecc.	
--	---	--

### Contenuti

- L'ora del codice 'minecraft'
- Il linguaggio delle cose: inventiamo oggetti "smart"
- Code.org corso 4
- Approfondimento del concetto di Debug
- Creazione di un videogioco con Code.org
- Remix di semplici progetti nati con Scratch
- Ambienti di sviluppo Scratch:
  - Pixel e coordinate dello schermo
  - Aspetto e utilizzo dei costumi
  - Controllo
  - Movimento
  - Sensori (attendi, sta toccando)
  - Stage ed utilizzo degli sfondi
  - Utilizzo di Variabili e Funzioni
  - Messaggi (comunicazione tra gli sprite)
- Creazione di uno Storytelling con Scratch
- Creazione di un videogioco con Scratch
- Introduzione alla programmazioni ad oggetti
- Ambiente di sviluppo App Inventor: Interfaccia grafica
  - Oggetti e loro metodi
  - Sensori e Timer
- Remix di un videogioco esistente in App Inventor
- Creazione di un App per SmartPhone Android con App Inventor
- Utilizzo software di progettazione Lego Mindstorm (costruire un robot)
- Utilizzo software di programmazione Lego Mindstorm (programmare un robot):
  - Motori e sensori
  - Costrutti principali utilizzati nella programmazione dei robot
- Realizzazione di un proprio robot Lego Mindstorm da far interagire con altri robot simili