*ISTITUTO COMPRENSIVO “PABLO NERUDA”*

**CURRICOLO STEAM**

*SCUOLA SECONDARIA DI PRIMO GRADO*

*a.s. 2022-23*

**Premessa**

STEM è l’acronimo di Scienze, Tecnologia, Ingegneria e Matematica e fa riferimento ad una revisione delle metodologie didattiche finalizzata all’integrazione delle discipline scientifiche con quelle non scientifiche, integrazione necessaria per affrontare e comprendere la complessità che la realtà implica.

STEM pertanto può essere considerata come la tendenza ad integrare le varie discipline in maniera più o meno profonda affrontando gli argomenti da trattare o i problemi da risolvere senza che vi sia un confine stabilito tra gli strumenti delle varie discipline.

Una tale integrazione tra le varie discipline necessita di modalità di apprendimento attive, quali ad esempio:

-il tinkering, una forma di apprendimento informale in cui si “impara facendo” per esprimersi e sperimentare, realizzando oggetti con materiali poveri, puntando più sul processo che sul risultato;

- la tecnologia per l’apprendimento attivo (TEAL - Technology Enabled Active Learning) con simulazioni pratiche al computer.

STEM o STEAM (a cui si aggiunge la componente Arte) non sono una novità, sono semplicemente modi di comprendere e applicare una forma integrata di apprendimento che assomiglia alla vita reale.

Invece di insegnare la matematica separatamente dalla scienza, possono essere insegnate insieme in un modo che le conoscenze di questi due campi si completino e si sostengano a vicenda.

**Riferimenti normativi**

Il curricolo costituisce la descrizione di un percorso che si vuole costruire e delle mete che si intendono raggiungere, data una situazione di partenza.

Nella stesura del curricolo STEAM non si può prescindere dai documenti europei, in primis dalla **Raccomandazione del Parlamento europeo del 2006** che fra le otto competenze chiave per l’apprendimento permanente annovera ***la competenza digitale***.

Dal Documento dell’Unione Europea del 2006:

*“Le competenze chiave sono quelle di cui tutti hanno bisogno per la realizzazione e lo sviluppo personali, la cittadinanza attiva, l’inclusione sociale, l’occupazione.*

*Gli Stati Membri devono sviluppare l’offerta di competenze chiave per tutti nell’ambito delle loro strategie di apprendimento permanente per assicurare che:*

*-L’istruzione e la formazione iniziale offrano a tutti i giovani gli strumenti per sviluppare le competenze chiave ad un  livello tale che li prepari per la vita adulta e costituisca la base*

*per ulteriori occasioni di apprendimento*

*-Gli adulti siano in grado di sviluppare ed aggiornare le loro competenze chiave in tutto l’arco della vita … “*

Tra le ***otto competenze chiave europee di Cittadinanza***, al pari della comunicazione in madrelingua e della competenza matematica, c’è la ***Competenza Digitale:***

*“Utilizzare e produrre strumenti di comunicazione visiva e multimediale, anche con riferimento alle strategie espressive e agli strumenti tecnici della comunicazione in rete.*

*Utilizzare le reti e gli strumenti informatici nelle attività di studio, ricerca e approfondimento disciplinare”*

Nelle **Indicazioni nazionali per il curricolo della scuola dell’infanzia e del primo ciclo d’istruzione del 2012**, si legge che *“La diffusione delle tecnologie...è una grande opportunità e rappresenta la frontiera decisiva per la scuola”.*

E nel paragrafo dedicato alla tecnologia “*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 di progetti e per la comprensione del rapporto tra codice sorgente e risultato visibile.”*

Nel **Piano Nazionale della Scuola Digitale** il MIUR lancia una strategia complessiva per l’innovazione della scuola italiana ed un nuovo posizionamento del suo sistema educativo nell’era digitale.

**Obiettivi generali del percorso**

1. Introdurre il modello del ***“Pensiero Computazionale”*** come metodo per la risoluzione dei problemi che aiuta a riformularli per poterli risolvere efficientemente in modo automatico;
2. Introdurre in modo intuitivo e ludico i concetti base della programmazione per sviluppare il pensiero computazionale.
3. Offrire esperienze di progettazione robotica per sviluppare capacità di problem solving ed estese abilità tecnologiche.
4. Analizzare e organizzare i dati del problema in base a criteri logici.
5. Rappresentare i dati del problema tramite opportune astrazioni.
6. Formulare il problema in un formato che ci permette di usare un “sistema di calcolo” (nel senso più ampio del termine, ovvero una macchina, un essere umano, o una rete di umani e macchine) per risolverlo.
7. Automatizzare la risoluzione del problema definendo una soluzione algoritmica, consistente in una sequenza accuratamente descritta di passi, ognuno dei quali appartenente ad un catalogo ben definito di operazioni di base.
8. Identificare, analizzare, implementare e verificare le possibili soluzioni con un’efficace ed efficiente combinazione di passi e risorse (avendo come obiettivo la ricerca della soluzione migliore secondo tali criteri).
9. Generalizzare il processo di risoluzione del problema per poterlo trasferire ad un ampio spettro di altri problemi.

L’obiettivo è soprattutto quello di sviluppare il ***pensiero computazionale***, attraverso l’utilizzo dei costrutti della programmazione, per la realizzazione di algoritmi ottimizzati ed efficaci alla risoluzione di problemi, utilizzando in modo creativo e transdisciplinare anche la anche robotica educativa.

Si considerano i concetti base dell'informatica, adatti allo specifico livello di maturazione degli studenti, 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 i parametri.

L’esecuzione di attività con ambienti di programmazione visuali quali, per esempio, Scratch, consente agli alunni di ideare e realizzare qualcosa di proprio: uno storytelling, un videogioco, un’App ecc.

L’utilizzo, infine, di supporti legati alla robotica educativa consente di ridurre il livello di astrazione e di applicare gli algoritmi appresi per realizzare la programmazione di oggetti che interagiscono con il mondo reale (intelligenza delle cose).

|  |  |  |  |
| --- | --- | --- | --- |
| **PERIODO** | **METODOLOGIA** | **STRUMENTI** | **VERIFICHE** |
| Intero triennio, iniziando dalle classi prime dall’a.s.2022/23  Tempi: 2 h settimanali a partire dal mese di Ottobre 2022 | Thinkering  Learning by doing;  Cooperative Learning;  Tutoring;  Debugging (didattica dell’errore) | Code.org;  Scratch;  Blockly;  Attività unplugged  CodyRoby, Codyway.  Scheda Arduino  Mbot  Lego Spike  Thinkcard  Adobe Creative Cloud Express  Stampante 3 D  Droni didattici  Telecamera 360 | Osservazioni  I prodotti finali realizzati (attività unplugged, animazioni, simulazioni, videogiochi). |

|  |
| --- |
| **RACCORDI CON LE DISCIPLINE SCOLASTICHE (INTERDISCIPLINARIETÀ E TRANSDISCIPLINARIETÀ)** |
| Nello sviluppo del pensiero computazionale e della didattica STEAM concorrono tutte le discipline.  Nelle attività proposte e nel percorso di coding vengono esplorati concetti ed attività che hanno ricaduta diretta su molte discipline quali:  - **matematica** **e tecnologia** (operazioni, linee, angoli, figure geometriche piane e solide, problem solving),  - **geografia** (orientamento, lateralizzazione, localizzazione)  - **storia** (concetti di ciclicità e sequenza),  - **italiano** (la sequenzialità e la logicità del linguaggio, il saper dare indicazioni precise, leggere e produrre testi regolativi, esplorare le varie forme di narrazione digitale)),  - **arte** (attraverso lo sviluppo di disegni artistici attraverso il pc e non, pixel art),  - **scienze motorie** (attività unplugged svolte in palestra che implicano la messa in gioco di schemi motori).  **- musica** (creazione di suoni, la musica e la matematica)  Tutte le discipline collaborano e interagiscono tra loro per giungere alla soluzione di un problema esplorando le varie forme della narrazione digitale, sino alla creazione di oggetti reali e virtuali. |

|  |  |  |
| --- | --- | --- |
| **COMPETENZE** | **ABILITÀ** | **CONOSCENZE** |
| **PENSIERO RIFLESSIVO E PROBLEM**  **SOLVING**  - Utilizzare le strategie del pensiero razionale negli aspetti dialettici e algoritmici per affrontare situazioni problematiche, elaborando opportune soluzioni.  **COLLABORAZIONE E COMUNICAZIONE**  - Partecipare attivamente alle attività portando il proprio contributo personale  -Collaborare e partecipare comprendendo i diversi punti di vista  **INIZIATIVA E PENSIERO AUTONOMO**  - Agire in modo autonomo e responsabile  - progettare e pianificare  **IMPARARE AD IMPARARE**  - Utilizzare trasversalmente le  conoscenze  **CREATIVITÀ**  - Usare l’immaginazione o idee originali per creare qualcosa | **COSTRUIRE ABILITA’ DI BASE**  - Imparare a scrivere algoritmi, sia legati alla vita quotidiana, sia a qualunque attività didattica, in maniera corretta per risolvere problemi complessi scomponendoli in problemi più semplici.  - Applicare il concetto di variabile nella soluzione di schemi logici  - La rappresentazione dello spazio sia nel mondo digitale con Thinkecard, sia nella realtà, costruendo oggetti immaginati con la stampante 3D  **LEGGERE E SCRIVERE IL CODICE**  - Conoscere le basi del passaggio dall’algoritmo al codice, utilizzando il linguaggio a blocchi.  - Progettare e costruire storie tecnologiche unendole con il linguaggio di programmazione Scratch.  - Remixare programmi esistenti della comunità Scratch.  -Realizzare storie interattive, animazioni, simulazioni, videogiochi, con Scratch:  -Saper creare progetti nuovi con Scratch che facciano uso di tutte le strutture analizzate.  **COSTRUIRE E PROGRAMMARE UN ROBOT**  **-** Basi della programmazione, della codifica e dell'elettronica, inclusa la corrente, la tensione e la logica digitale avvicinando gli alunni al mondo Arduino con Arduino Student Kit e la scheda virtuale di Thinkcard  - Capire cosa sono e come usare sensori e motori per rendere interattivi i modelli  - Stabilire relazioni causa-effetto  - Creare semplici programmi per istruire i modelli Lego e Mbot  **PROBLEM SOLVING**  - Saper lavorare in gruppo.  - Ottimizzare la soluzione di un problema  - Scomporre un problema complesso in problemi di complessità inferiore  - Saper realizzare un algoritmo utile per la soluzione di un problema generico  - Utilizzare le conoscenze disciplinari per riempire di contenuti le attività:  - realizzare storytelling;  - Costruire realtà virtuali attraverso la documentazione del “passato”.  - conoscere le principali regole nella realizzazione di figure geometriche anche di complessità elevata;  - realizzare programmi di musica | - Conoscere il concetto logico di  variabile  - Conoscere il corretto utilizzo delle  funzioni nella realizzazione di un  algoritmo  - Conoscere il concetto di parametro  ed il suo utilizzo all’interno delle  funzioni  - Conoscere le principali funzioni di  un programma ed utilizzarle in  modo congruo  - Conoscere l’ambiente di  programmazione Scratch  - Conoscere l’ambiente di sviluppo  App Inventor  - Conoscere le caratteristiche di un  robot (sensori e motori)  - Conoscere l’ambiente di  progettazione Lego Spike  - Conoscere l’utilizzo di robot mBot e  del suo ambiente di  programmazione  - Conoscere le basi di utilizzo della  scheda Arduino  - Conoscere l’utilizzo e le funzioni di Thinkecard per un approccio digitale al Thinkering  - Conoscere le funzioni di Adobe Creative Cloud Express per realizzare ebook ed animazioni digitali |
|  |

|  |
| --- |
| **CONTENUTI** |
| - L’ora del codice: Minecraft, Frozen, “Ballando con il codice”.  - Il linguaggio delle cose: inventiamo oggetti “smart”  - Code.org sintesi corso rapido 20 ore  - Approfondimento del concetto di Debug  - Creazione di videogiochi didattici con Code.org  - Remix di semplici progetti nati con Scratch  - Basi della programmazione :dall’algoritmo al codice, utilizzando i codici a blocco di Scratch.  - Ambienti di sviluppo Scratch:  - Creazione di uno Storytelling con Scratch  - Creazione di un videogioco didattico con Scratch  - Creazione di “App” didattiche con Scratch e App Inventor  - Basi della programmazione, della codifica e dell'elettronica, inclusa la corrente, la tensione e la logica digitale avvicinando gli alunni al mondo Arduino con Arduino Student Kit  - Introduzione alla programmazione di oggetti smart  - Utilizzo software di progettazione Lego Spike (costruire un robot)  - Utilizzo software di programmazione Lego Spike (programmare un robot):  - Realizzazione di un proprio robot Lego Spike da far interagire con altri robot simili  - Utilizzo di robot mBot basati su schede Arduino e programmazione attraverso ambienti simili a Scratch  - Costrutti principali utilizzati nella programmazione dei robot  - Storytelling con Scratch e Creative Cloud Express  - Modellazione 3 D con Tinkecard ed applicazioni pratiche su stampante 3D |