

Premessa

Il progetto ha lo scopo di introdurre una didattica innovativa della cultura scientifica e tecnologica .

Partendo da un'alfabetizzazione digitale, si arriva allo sviluppo del pensiero computazionale, essenziale affinché le nuove generazioni siano in grado di affrontare la società e le tecnologie del futuro, non come consumatori passivi, ma come utenti attivi. Il pensiero computazionale, che è alla base del coding, è un processo mentale per la risoluzione di problemi costituito da strumenti concettuali utili per affrontare molti tipi di problemi in tutte le discipline e non solo perché direttamente applicati nei calcolatori, nelle reti di comunicazione. I benefici del “pensiero computazionale” si estendono a tutti gli ambiti disciplinari per affrontare problemi complessi, ipotizzare soluzioni che prevedono più fasi, immaginare una descrizione chiara di cosa fare e quando farlo.

Perché il coding nella scuola?

- ✓ Crea motivazione nei ragazzi
- ✓ Utilizza strategie didattiche legate al gioco (game based learning).
- ✓ Favorisce il lavoro in team e l'apprendimento cooperativo-collaborativo.
- ✓ Fornisce strumenti per la realizzazione di una didattica inclusiva attraverso il:
 - ✓ Lavoro di gruppo o progettuale.
 - ✓ Learning by doing: apprendimento per prove ed errori.
 - ✓ Peer tutoring.
 - ✓ Pair programming.

Lo scopo principale di questo percorso è avviare gli alunni al pensiero computazionale, ovvero ad un approccio inedito ai problemi e alla loro soluzione. Infatti i nostri alunni con il coding svilupperanno il pensiero computazionale e l'attitudine a risolvere problemi più o meno complessi, non impareranno solo a programmare ma programmeranno per apprendere.

In sintesi si troveranno davanti a quello che più li diverte: un tablet, un monitor di un pc, un piccolo robot, e saranno loro ad animare, far prendere vita, imparare a fare muovere i loro personaggi in un certo modo, siano essi virtuali o meno (come nel caso della robotica educativa). In conclusione impareranno a raggiungere un obiettivo divertendosi.

Titolo unità didattica

“VERSO UN’OTTICA DI APPRENDIMENTO: IDENTITÀ, PENSIERO, CULTURA”

Periodo di svolgimento

Ottobre – novembre – dicembre – Gennaio

Classi 5 A – 5 B – 5 C - 5 D

A.S.2019/2020

Disciplina

TECNOLOGIA

Competenze chiave europee

- ✓ COMUNICAZIONE NELLA MADRELINGUA O LINGUA DI ISTRUZIONE
- ✓ COMUNICAZIONE NELLE LINGUE STRANIERE
- ✓ COMPETENZA MATEMATICA E COMPETENZE DI BASE IN SCIENZA E TECNOLOGIA
- ✓ COMPETENZA DIGITALE
- ✓ IMPARARE A IMPARARE
- ✓ COMPETENZE SOCIALI E CIVICHE
- ✓ SPIRITO DI INIZIATIVA
- ✓ CONSAPEVOLEZZA ED ESPRESSIONE CULTURALE

Traguardi per lo sviluppo delle competenze

(Indicare solo quelli pertinenti con l’UD e fedeli alle Indicazioni Nazionali)

- Avviare gli alunni all’acquisizione della logica della programmazione.

- Acquisizione della terminologia specifica attraverso la dimostrazione e l'elaborazione di mini attività di gioco sempre più complesse.
- Sviluppare attenzione, concentrazione e motivazione.
- Sviluppare la capacità di analizzare e risolvere problemi.
- Sviluppare autonomia operativa.
- Favorire lo sviluppo della creatività attraverso la molteplicità di modi che l'informatica offre per affrontare e risolvere un problema.
- Avviare alla progettazione di algoritmi, producendo risultati visibili (anche se nel mondo virtuale).
- Aiutare a padroneggiare la complessità (imparare a risolvere problemi informatici aiuta a risolvere problemi complessi in altre aree).
- Sviluppare il ragionamento accurato e preciso (la scrittura di programmi che funzionano bene richiede l'esattezza in ogni dettaglio).

Obiettivi di Apprendimento

(Indicare solo quelli pertinenti con l'UD e fedeli alle Indicazioni Nazionali)

- ♥ Sviluppare percorsi laboratoriali in tutte le aree del sapere
- ♥ Coinvolgere attivamente gli studenti nel loro processo di apprendimento e di costruzione delle conoscenze, promuovendo il pensiero creativo.
- ♥ Stimolare la capacità di analisi, la capacità organizzativa e la capacità di comunicare, utilizzando l'operatività

Obiettivi di Apprendimento Specifici

(A discrezione del docente)

- ♥ Saper interpretare codici segreti
- ♥ Sviluppare la capacità di osservazione.
- ♥ Stimolare l'arte e la creatività.
- ♥ Sapersi orientare e muoversi su una scacchiera.
- ♥ Saper interpretare (leggere e decodificare) e creare (scrivere) istruzioni.

- ♥ Tecnologia digitale: utilizzare software per disegnare.
- ♥ Avviare alla progettazione di algoritmi, producendo risultati visibili (anche se nel mondo virtuale).
- ♥ Sviluppare il ragionamento accurato e preciso (la scrittura di programmi che funzionano bene richiede l'esattezza in ogni dettaglio).

Contenuti

(A discrezione del docente)

- ♥ **Il coding prima dell'informatica:** Codici e segreti
- ♥ **Il coding a quadretti:** La pixel art
- ♥ **La strada del coding:** Giochiamo con i percorsi
- ♥ **A una condizione:** Se, allora, altrimenti.
- ♥ **Primi passi nell'informatica:** iniziamo a programmare.

Attività

(A discrezione del docente)

Introduzione in modo intuitivo e ludico ai concetti base della programmazione per sviluppare il pensiero computazionale.

In particolare, le attività con gli alunni introdurranno i seguenti concetti:

- ✓ oggetti programmabili
- ✓ algoritmo
- ✓ programmazione visuale a blocchi
- ✓ esecuzione di sequenze di istruzioni elementari
- ✓ esecuzione ripetuta di istruzioni
- ✓ esecuzione condizionata di istruzioni
- ✓ definizione e uso di procedure
- ✓ verifica e correzione del codice
- ✓ riuso del codice

- ✓ crittografia

Strategie metodologico – didattiche

Metodo induttivo

Metodo sperimentale

Procedure di ricerca

Scoperta guidata

Roleplaying

Peer tutoring

Cooperative learning

Problem solving

- ✓ Lezione frontale
- ✓ E-learning (Apprendere attraverso internet)
- ✓ Cooperative learning (gli studenti lavorano insieme in piccoli gruppi per raggiungere obiettivi comuni)
- ✓ Learning by doing (imparare facendo, imparare attraverso il fare)
- ✓ Brainstorming (Tempesta di cervelli)

Sussidi e mediatori didattici

- ✓ Libri di testo
- ✓ Testi didattici di supporto
- ✓ Stampa specialistica
- ✓ Documenti di supporto
- ✓ Schede predisposte dall'insegnante
- ✓ Sussidi visivi (diapositive, fotografie, cartelloni, ecc.)
- ✓ Sussidi audiovisivi (VHS, film, ecc.)
- ✓ Sussidi sonori (registratori, CD, musicassette, ecc)
- ✓ Sussidi informatici

- ✓ Attrezzature in dotazione alla palestra
- ✓ Supporti (lavagna a gesso, lavagna luminosa)
- ✓ Lim
- ✓ Zaplycode (strumento didattico per il coding)

Il lato scientifico-culturale dell'informatica, definito anche pensiero computazionale, aiuta a sviluppare competenze logiche e capacità di risolvere problemi in modo creativo ed efficiente, qualità che sono importanti per tutti i futuri cittadini. Il modo più semplice e divertente di sviluppare il pensiero computazionale è attraverso la programmazione (coding) in un contesto di gioco. E' possibile utilizzare tecnologie non particolarmente complicate e dai costi accessibili, come i Robottini educativi parlanti DOC

Raccordi con altre discipline

Geometria
Arte
Geografia
Scienze
Ed. fisica

Competenze da verificare al termine della UD

(Esse saranno oggetto della prova di verifica bimestrale)

L'alunno:

1. Interpreta codici segreti
2. Sviluppa la capacità di osservazione.
3. Stimola l'arte e la creatività. Sa orientarsi e muoversi su una scacchiera.
4. Sa interpretare (leggere e decodificare) e creare (scrivere) istruzioni.

Tecnologia digitale: utilizza software per disegnare

L'alunno si

1. Avvia alla progettazione di algoritmi.
2. produce risultati visibili (anche se nel mondo virtuale).
3. Sviluppa il ragionamento accurato e preciso (la scrittura di programmi che funzionano bene richiede l'esattezza in ogni dettaglio).

Titolo unità didattica

COMPETENZE IN AZIONE: VERSO NUOVE METE

Periodo di svolgimento

Febbraio, Marzo, Aprile, Maggio

Classi 5 A – 5 B – 5 C - 5 D

A.S.2019/2020

Disciplina

Tecnologia

Competenze chiave europee

- ✓ COMUNICAZIONE NELLA MADRELINGUA O LINGUA DI ISTRUZIONE
- ✓ COMUNICAZIONE NELLE LINGUE STRANIERE
- ✓ COMPETENZA MATEMATICA E COMPETENZE DI BASE IN SCIENZA E TECNOLOGIA
- ✓ COMPETENZA DIGITALE
- ✓ IMPARARE A IMPARARE
- ✓ COMPETENZE SOCIALI E CIVICHE
- ✓ SPIRITO DI INIZIATIVA
- ✓ CONSAPEVOLEZZA ED ESPRESSIONE CULTURALE

Traguardi per lo sviluppo delle competenze

(Indicare solo quelli pertinenti con l'UD e fedeli alle Indicazioni Nazionali)

- Avviare gli alunni all'acquisizione della logica della programmazione.
- Acquisizione della terminologia specifica attraverso la dimostrazione e l'elaborazione di mini attività di gioco sempre più complesse.
- Sviluppare attenzione, concentrazione e motivazione.
- Sviluppare la capacità di analizzare e risolvere problemi.
- Sviluppare autonomia operativa.
- Favorire lo sviluppo della creatività attraverso la molteplicità di modi che l'informatica offre per affrontare e risolvere un problema.
- Avviare alla progettazione di algoritmi, producendo risultati visibili (anche se nel mondo virtuale).
- Aiutare a padroneggiare la complessità (imparare a risolvere problemi informatici aiuta a risolvere problemi complessi in altre aree).
- Sviluppare il ragionamento accurato e preciso (la scrittura di programmi che funzionano bene richiede l'esattezza in ogni dettaglio).

Obiettivi di Apprendimento

(Indicare solo quelli pertinenti con l'UD e fedeli alle Indicazioni Nazionali)

- ♥ Utilizzare il programma Scratch per la programmazione informatica attraverso i blocchi visivi.
- ♥ Acquisire la terminologia specifica attraverso la dimostrazione e l'elaborazione di mini attività di gioco sempre più complesse.

Obiettivi di Apprendimento Specifici

(A discrezione del docente)

- ♥ Confrontare valori usando gli operatori di uguaglianza (=), di minoranza (<) o di maggioranza (>).
- ♥ Tradurre frasi condizionali espresse nel linguaggio naturale in istruzioni di un programma.
- ♥ Identificare quando è necessario usare un'istruzione condizionale per trattare valori non noti.
- ♥ Eseguire (passo passo) un algoritmo che contiene istruzioni condizionali.
- ♥ Utilizzare istruzioni condizionali per fare scelte basate sulla logica.
- ♥ Annidare condizionali per analizzare condizioni che presentano più di due alternative, usando cioè la logica se / altrimenti-se / altrimenti.
- ♥ Scrivere funzioni che contengono condizionali nidificati.

- ♥ Favorire lo sviluppo della creatività attraverso la molteplicità di modi che l'informatica offre per affrontare e risolvere un problema.
- ♥ Avviare alla progettazione di algoritmi, producendo risultati visibili (anche se nel mondo virtuale).
- ♥ Conoscere il programma Scratch e creare semplici lavori

Contenuti

(A discrezione del docente)

1. **Introduzione a Scratch:** un gattino per il coding.
2. **Introduzione a Scratch:** Si va in scena
3. **Primi passi con Scratch:** Un disegno pasticciato
4. **La geometria con Scratch:** Disegnare un quadrato;
5. **La musica con Scratch:** Il pianoforte buffo;
6. **Suoni e immagini:** Storytelling con Scratch;
7. **Domande e risposte:** Il mio maestro Scratch;
8. **Le probabilità con Scratch:** Lanciamo i dadi;
9. **Creare un videogioco:** Scratch nel labirinto,

Attività

(A discrezione del docente)

Le attività con gli alunni introdurranno i seguenti concetti:

- ✓ oggetti programmabili
- ✓ algoritmo
- ✓ programmazione visuale a blocchi
- ✓ esecuzione di sequenze di istruzioni elementari
- ✓ esecuzione ripetuta di istruzioni
- ✓ esecuzione condizionata di istruzioni
- ✓ definizione e uso di procedure

- ✓ verifica e correzione del codice
- ✓ riuso del codice
- ✓ crittografia

Nell'ambito dell'attività laboratoriale i ragazzi utilizzeranno Code.org, una piattaforma internazionale che, insegna a muovere i primi passi nel mondo della programmazione partendo dall'attività "L'Ora del Codice".

Prodotti finali sezione 1 e sezione 2

Per ciascuna sezione del Progetto si prevede la realizzazione di prodotti pubblicizzabili all'esterno:

allestimento di mostre, organizzazione di “tavoli” di ragionamento, sfide di logica, presentazione di giochi informatici realizzati dagli alunni.

Strategie metodologico – didattiche

Metodo induttivo

Metodo sperimentale

Procedure di ricerca

Scoperta guidata

Roleplaying

Peer tutoring

Cooperative learning

Problem solving

- ✓ Lezione frontale
- ✓ E-learning (Apprendere attraverso internet)
- ✓ Cooperative learning (gli studenti lavorano insieme in piccoli gruppi per raggiungere obiettivi comuni)
- ✓ Learning by doing (imparare facendo, imparare attraverso il fare)
- ✓ Brainstorming (Tempesta di cervelli)

Sussidi e mediatori didattici

- ✓ Libri di testo
- ✓ Testi didattici di supporto

- ✓ Stampa specialistica
- ✓ Documenti di supporto
- ✓ Schede predisposte dall'insegnante
- ✓ Sussidi visivi (diapositive, fotografie, cartelloni, ecc.)
- ✓ Sussidi audiovisivi (VHS, film, ecc.)
- ✓ Sussidi sonori (registratori, CD, musicassette, ecc)
- ✓ Sussidi informatici
- ✓ Attrezzature in dotazione alla palestra
- ✓ Supporti (lavagna a gesso, lavagna luminosa)
- ✓ Lim
- ✓ Zaplycode (strumento didattico per il coding)

Il lato scientifico-culturale dell'informatica, definito anche pensiero computazionale, aiuta a sviluppare competenze logiche e capacità di risolvere problemi in modo creativo ed efficiente, qualità che sono importanti per tutti i futuri cittadini. Il modo più semplice e divertente di sviluppare il pensiero computazionale è attraverso la programmazione (coding) in un contesto di gioco. E' possibile utilizzare tecnologie non particolarmente complicate e dai costi accessibili, come i Robottini educativi parlanti DOC

Raccordi con altre discipline

Geometria
Arte
Geografia
Scienze
Ed. fisica
Italiano

Competenze da verificare al termine della UD

(Esse saranno oggetto della prova di verifica bimestrale)

L'alunno:

1. Confronta valori usando gli operatori di uguaglianza (=), di minoranza (<) o di maggioranza (>).
2. Traduce frasi condizionali espresse nel linguaggio naturale in istruzioni di un programma.
3. Identifica, quando è necessario usa un'istruzione condizionale per trattare valori non noti.
4. Esegue (passo passo) un algoritmo che contiene istruzioni condizionali.
5. Utilizza istruzioni condizionali per fare scelte basate sulla logica.

6. Annida condizionali per analizzare condizioni che presentano più di due alternative, usando cioè la logica se / altrimenti-se / altrimenti.
7. Scrive funzioni che contengono condizionali nidificati.
8. Favorisce lo sviluppo della creatività attraverso la molteplicità di modi che l'informatica offre per affrontare e risolvere un problema.
9. Si avvia alla progettazione di algoritmi, producendo risultati visibili (anche se nel mondo virtuale).
10. Conosce il programma Scratch e creare semplici lavori

Castrovillari, 23/10/2019

Le insegnanti:

Rosamaria Lucia Rubino

Maria Burgio