



PROGETTO DI TUTORAGGIO
 NELLA FORMAZIONE DEI DOCENTI
 PER L'INTRODUZIONE DEL PENSIERO COMPUTAZIONALE
 ATTRAVERSO ATTIVITA' DI CODING

azione #17 PNSD

a.s.2016-2017

"Non si impara dalla tecnologia, come non si impara
 dall'insegnante. Si impara attraverso il pensiero:
 pensando a cosa si sta facendo o alle cose in cui si crede,
 a cosa altri hanno fatto o sulle cose in cui altri credono,
 pensando al processo che il pensiero svolge. Il pensiero
 media l'apprendimento. L'apprendimento è il risultato
 del pensiero".

(David Jonassen)

TITOLO, RESPONSABILE E BREVE SINTESI DEL PROGETTO

TITOLO: "Coding in your classroom, now!" (CodeMOOC)

RESPONSABILE: Ins. Troia Daniela

SINTESI: La docente Daniela Troia, a seguito della certificazione ottenuta al termine del percorso realizzato sulla piattaforma EMMA con il MOOC (Massive Open Online Course) "Coding in your classroom, now", si rende disponibile al tutoraggio dei colleghi dell'Istituto che vogliano intraprendere lo stesso corso.

Questo progetto vuole accompagnare i docenti nella frequenza di "Coding in your classroom, now!", promosso dall'Università di Urbino sotto il coordinamento del Prof. Alessandro Bogliolo, fruibile on demand sulla piattaforma EMMA dell'Università Federico II di Napoli.

Il corso ha ottenuto il riconoscimento della Comunità Europea grazie alla sua validità nell'aiutare gli insegnanti ad introdurre il pensiero computazionale in classe e per offrire un'ottima possibilità di alfabetizzazione; per questo è stato accreditato anche da Programma il Futuro, l'iniziativa del MIUR, in collaborazione con CINL.

Gli effetti di CodeMOOC sono stati talmente evidenti in termini quantitativi e qualitativi che l'Università di Urbino ha deciso di impegnarsi a tenere aperto CodeMOOC per almeno altri due anni e ad estenderne l'impatto. Questo impegno è stato accolto dalla Commissione Europea come prima azione della nuova **Agenda per le Competenze per l'Europa**, che quindi riparte proprio da qui.

PREMESSA

Quando si affronta un problema o si ha un'idea, spesso si intuisce la soluzione ma non si è in grado di formularla in modo operativo per metterla in pratica.

Il pensiero computazionale è proprio questo, la capacità di immaginare e descrivere un procedimento costruttivo che porti alla soluzione. Come imparare a parlare ci aiuta a formulare pensieri complessi, così il pensiero computazionale ci offre strumenti ulteriori a supporto, non solo della logica, ma anche della fantasia e della creatività.

Il pensiero computazionale è una capacità trasversale che va sviluppata il prima possibile, per questo motivo incontra le esigenze di formazione, non solo dei docenti di scuola primaria e secondaria, ma anche della scuola dell'infanzia.

NORMATIVA, MOTIVAZIONI E INIZIATIVE ISTITUZIONALI

La pervasività dell'informatica e il suo essere indispensabile nella vita quotidiana rende sempre più necessario l'inserimento del pensiero computazionale e del Coding, che ne costituiscono la base, nei processi formativi a tutti i livelli.

Tale esigenza, già esplicitata all'interno della Raccomandazione del Parlamento europeo del 2006 che fra le otto competenze chiave per l'apprendimento permanente annovera la competenza digitale, è stata recepita nelle "Indicazioni Nazionali per il Curricolo della scuola dell'Infanzia e del Primo ciclo dell'Istruzione (2012)", ed è stata rafforzata quando il MIUR in collaborazione con il CINI (Consorzio Interuniversitario Nazionale per l'Informatica) ha lanciato l'iniziativa "Programmare il Futuro"

Essere competenti dal punto di vista digitale non significa solo saper utilizzare oggetti digitali o saper ricavare l'informazione dai social media, ma significa anche conoscere la logica e il funzionamento dei media digitali per favorirne un uso consapevole e responsabile. Il fine ultimo è quello formare nelle aule utenti attivi in grado di decidere autonomamente cosa consumare e come consumarlo. Questa è, peraltro, una delle nuove alfabetizzazioni previste dal documento "La Buona scuola: l'alfabetizzazione digitale", azione promossa dalla legge di riforma n.107/2015, che implica necessariamente il passaggio da consumatori digitali a producer maker.

FINALITA'

- Introdurre il pensiero computazionale in classe attraverso il coding, usando solo attività intuitive e divertenti da proporre direttamente agli alunni.
- Avviare attività volte alla condivisione di buone pratiche didattiche.
- Implementare la comunicazione interna e la condivisione di esperienze attraverso la costituzione di una rete di relazione e rapporti.

OBIETTIVI

- Promuovere lo sviluppo le competenze digitali degli alunni e dei docenti
- Realizzare buone pratiche in grado di innovare la didattica.

STRUTTURA DEL CORSO

Il corso, tenuto dal professor Alessandro Bogliolo, è organizzato in 6 lezioni, suddivise in unit. Ad ogni unit sono associati una video-lezione di un'ora e materiali didattici interni o esterni. Ai corsisti sono inoltre assegnati compiti, la cui verifica può essere affidata al docente o a meccanismi di valutazione tra pari.

Il corso prevede anche il conseguimento del certificato di completamento del "Corso rapido" di Code.org (k-8).

L'impegno complessivo per il completamento del MOOC è di **48 ore**, calcolate come segue:

- videolezioni **23 ore** (considerando solo il tempo necessario alla visione della lezione)
- compiti **5 ore** (considerando un'ora di impegno per la consegna di ciascun compito)
- corso breve di Code.org **20 ore** (considerando il tempo nominale stimato da Code.org)

La frequenza delle videolezioni e la consegna dei compiti sono documentati dalla piattaforma EMMA.

Il completamento del corso breve di Code.org deve essere documentato dal conseguimento del certificato K-8 a nome del corsista.

Al fine del rilascio del badge e del certificato di completamento il corsista deve consegnare anche i tre compiti associati alla Unit 6.5, per i quali non sono riconosciute ore ulteriori in quanto non comportano attività aggiuntive, ma servono solo a documentare l'attività svolta al di fuori della piattaforma.

CONTENUTI

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
definizione e uso di variabili e parametri
verifica e correzione del codice
riuso del codice
programma

TEMPI E LEZIONI

Settembre 2016 – Maggio 2017

Incontro Motivazionale: 22/09/2016

Il corso è suddiviso in 6 lezioni e al termine certifica 48 ore di formazione:

Lezione 1 - Presentazione del corso e del metodo

Lezione 2 - Il linguaggio delle cose

Lezione 3 - L'ora di codice e il labirinto

Lezione 4 - Il corso introduttivo di Code.org

Lezione 5 - Creare e condividere con Scratch

Lezione 6 - Il pensiero computazionale in pratica

ORGANIZZAZIONE:

Un incontro mensile della durata di 2 ore da Settembre a Maggio, per un totale di 18 ore.

Il calendario è da definire in base al calendario degli impegni annuali. Si ipotizza che gli incontri possano svolgersi nel primo pomeriggio, con orario da concordare.

Durante gli incontri in presenza si avvieranno discussioni sui contenuti proposti nel video seguito da casa: per ciascuna unità, infatti, i docenti saranno chiamati a svolgere un COMPITO che potrà richiedere la risoluzione di semplici esercizi di coding o l'ideazione di lezioni in cui utilizzare quelle attività nella loro classe, coinvolgendo gli alunni. Ogni riflessione sarà condivisa all'interno del gruppo così da favorire il confronto e il dialogo. Gli insegnanti acquisiranno, così, strumenti comuni e li adatteranno alle proprie classi innovando la didattica e costruendo buone pratiche educative. Sarà quindi fornito supporto per lo svolgimento del compito che richiederà un impegno a casa in maniera libera nei modi e nei tempi di esecuzione. In questo modo creeremo la nostra comunità di apprendimento fatta di insegnanti, ciascuno responsabile di un'ulteriore comunità di apprendimento formata dalla propria classe.

METODOLOGIE UTILIZZATE

Approccio mediato dal docente formatore

Learning by doing

Cooperative learning

Debugging (didattica dell'errore)

Tutoring

STRUMENTI

Per le attività online si useranno prevalentemente le risorse di Code.org e Scratch.

Per le attività unplugged (senza computer e senza rete) si seguiranno le indicazioni di CodyWay e CodyRoby.

Tra gli strumenti saranno usati come riferimento:

CodyRoby, un metodo di programmazione unplugged fai-da-te, basato su semplici carte da gioco che possono essere utilizzate per creare giochi da tavolo per ogni età o attività motorie.

CodyWay, un metodo di programmazione unplugged fai-da-te che consente di usare i percorsi nel mondo reale per fare esperienze di programmazione.

Code.org, un'organizzazione non-profit che promuove la diffusione del pensiero computazionale proponendo l'ora di codice (Hour of code) e offrendo strumenti didattici online ludici e intuitivi per giocare con la programmazione. In Italia il metodo di Code.org è adottato da Programma il futuro, l'iniziativa del Consorzio CINI e del MIUR.

Scratch, un linguaggio di programmazione visuale sviluppato al MIT Media Lab e reso disponibile online e offline attraverso una piattaforma che consente a chiunque di creare e condividere veri e propri programmi.

SPAZI

Saranno molto importanti per un maggior approfondimento alcuni spazi attrezzati:

Aule con LIM ed accesso a Internet.

Laboratorio di Informatica .

La stabilità della connessione ad Internet risulta significativa per lo svolgimento del corso sia degli insegnanti che degli alunni.

VERIFICA E VALUTAZIONE

E' propedeutica l'iscrizione alla piattaforma EMMA.

La frequenza del corso e il superamento delle prove in itinere saranno certificati.

Si assumeranno come **verifiche** gli ASSIGNMENTS prodotti dai docenti in piattaforma.

Ogni insegnante che frequenterà il corso per almeno l'80% delle attività riceverà un **attestato di completamento dalla piattaforma**; per ricevere l'attestato dall'Università di Urbino bisognerà aver completato il 100% delle attività e aver eseguito tutti gli assignment.

Per la **valutazione della formazione** si farà riferimento ai seguenti indicatori:

- conseguimento attestato di completamento (K8 + certificato di completamento delle 48 ore rilasciato dall'Università di Urbino)
- frequenza e partecipazione alle lezioni in presenza

