CODING, PENSIERO COMPUTAZIONALE E INNOVAZIONE DIDATTICA SIA PER GLI STUDENTI CHE PER I DOCENTI DI OGNI DISCIPLINA

La sfida del coding

Inserire il coding e il pensiero computazionale nella scuole di ogni ordine e grado è una sfida. Una sfida che grazie alle ultime innovazioni nel campo della didattica dell'informatica può essere vinta facilmente. La sfida consiste nell'individuare uno strumento e delle **metodologie** che risultino **interessanti, utili ed efficaci** non soltanto **per gli studenti,** ma anche e **soprattutto per i docenti**. Uno strumento che **unisca studenti e docenti** nella riconversione della scuola in un nuovo ambiente stimolante ed innovativo. E questo obiettivo è oggi alla nostra portata.

Se infatti gli **studenti**, oramai per la maggior parte **nati** in un mondo sostanzialmente orientato al **digitale** sono naturalmente portati ad usare, e ad usare con piacere, gli strumenti tecnologici, lo stesso **non** può essere detto della maggior parte dei **docenti** della scuola. Ma una **strategia efficace** dell'introduzione del digitale non può limitarsi a considerare l'uso del coding e del pensiero computazionale solo per poche ore alla settimana. Deve puntare ad introdurre l'uso di **strumenti digitali** a tutti i livelli e coinvolgendo i docenti di **tutte le materie**. L'applicabilità di una tale strategia è stata finora molto limitata dal senso di inadeguatezza che molti docenti provano nei confronti delle nuove tecnologie, senso di inadeguatezza che può essere non ultimo verificato dall'insufficiente uso delle LIM dove presenti.

Purtroppo gli **ambienti standard di programmazione**, che permettano lo sviluppo di supporti multimediali ed interattivi che risultino utili come supporto per le materie e le lezioni di una qualunque disciplina, sono **al di là della portata** della maggioranza dei docenti della scuola.

Una strategia unificata

La presente proposta, supportando quanto già presentato nella proposta https://labuonascuola.gov.it/area/m/7829/, intende completarla al fine di rendere l'uso del coding e del pensiero computazionale qualcosa di stabile nella scuola.

Oggi esistono infatti strumenti, basati sulla filosofia dei "mattoncini Lego", che sono alla portata di tutti e rendono l'approccio alla **programmazione** qualcosa di estremamente **stimolante** a tutti i livelli. Un esempio è l'ambiente usato dal sito **coding.org**. Questo ambiente, simpatico e coinvolgente per gli studenti a cui permette di risolvere giochi logici o di sviluppare divertenti animazioni interattive, risulta invece molto **limitato** nella sua facilità d'uso per scopi più generali e **non** risulta pertanto **utile** ai **docenti**.

La strategia illustrata in questa proposta è allora quella di usare **ambienti** più evoluti rispetto ad esempio a quello offerto da coding.org, ma allo stesso tempo perfettamente **utilizzabili dagli studenti di ogni età e dai docenti**, e che possano essere usati addirittura prima dell'ingresso nella scuola primaria.

L'ambiente al momento più maturo e più robusto è quello offerto dal tool **Scratch** (http://scratch.mit.edu) sviluppato dal MIT. Scratch permette agli studenti di creare giochi o animazioni multimediali ed interattive usando immagini, musica e suoni. Scratch, grazie al suo design, è disponibile **in italiano** e permette anche ai docenti di imparare in maniera semplice ed intutiva a sviluppare supporti multimediali ed interattivi per le loro lezioni, per qualunque disciplina. Se Scratch è adatto a partire dai 7/8 anni, a Scratch si affianca ora Scratch Junior, un tool che permette addirittura ai bambini in età prescolare di afferrare in maniera giocosa i principi del pensiero computazionale.

Scratch, già di per sé molto evoluto e più che sufficiente, è un prodotto **open source** e può essere modificato ed integrato per adattarsi alle specifiche esigenze dell'ambizioso ma realistico progetto scolastico nazionale che intenda migliorarlo per renderlo disponibile a studenti e docenti di ogni ordine e grado.

Scratch integra già al suo interno tool di disegno vettoriale, l'accesso alla webcam, l'uso di file multimediali permettendo agli studenti di sviluppare le loro **competenze digitali**.

I docenti di qualunque settore possono imparare ad usare Scratch in maniera intuitiva. Sono stati condotti esperimenti che hanno dimostrato come **docenti** di materie principalmente umanistiche delle scuole elementari **non abbiano** avuto **difficoltà** a scoprire da soli come **funziona Scratch**, e abbiamo appreso in maniera velocissima (meno di due ore) come poterlo usare per **realizzare** in maniera molto semplice ed immediata **supporti didattici** interattivi per le loro lezioni.

Nell'ambito di progetti condotti nelle scuole elementari si è visto come gli alunni stessi, naturalmente portati a esplorare e a sperimentare, abbiano coinvolto i loro insegnanti nell'uso di Scratch e siano addirittura giunti a spiegare loro il funzionamento e le potenzialità dello strumento. L'uso di uno strumento comune che permetta allo stesso tempo ai docenti di **innovare la didattica** e agli studenti di essere introdotti al **coding** e al **pensiero computazionale** è un'opportunità eccezionale.

Afferrare i concetti del coding e del pensiero computazionale divertendosi

Scratch permette agli studenti di **usare da subito**, anche senza doverli prima studiare teoricamente, sia tutti **i concetti** più importanti che sono alla base del coding e del pensiero computazionale che altri più avanzati, come ad esempio:

- Gli Algoritmi
- I Cicli
- Le Procedure
- Le Variabili
- Le basi degli Oggetti e delle classi
- La Programmazione concorrente

Una strategia naturalmente portata al BYOD

Esistono oggi versioni di Scratch (http://snap.berkeley.edu, http://www.blocklanguages.org, http//catrobat.org/) basate su Javascript che lo rendendono uno strumento sia accessibile online che **utilizzabile su un qualunque device**, sia esso un tablet o uno smartphone iOS o Android.

Scratch e i suoi derivati sono al momento utilizzati all'interno di corsi *BYOD* (Bring Your Own Device) a partire dalle scuole elementari fino ad arrivare ai corsi universitari, rendendolo quindi un strumento adatto a studenti di ogni ordine e grado.

Il supporto delle risorse

Il sito di Scratch offre una quantità e varietà incredibile di progetti sviluppati da studenti e da docenti di ogni disciplina per le loro lezioni. Al momento il sito ospita più di 6 milioni di **programmi** (chiamati *progetti*) liberamente **scaricabili, modificabili e riutilizzabili**. Al sito di Scratch si affianca inoltre il sito di ScratchEd (http://scratched.gse.harvard.edu/) che offre risorse di **supporto ai docenti** che intendono usare Scratch per sviluppare il pensiero computazionale dei propri studenti.

Raggiungere gli obiettivi unendo gli sforzi

Grazie all'uso di uno strumento unificato utile sia per lo studio che per il divertimento, e che si integra perfettamente come **supporto didattico innovativo** per qualunque tipologia di materia (italiano, grammatica, storia, matematica, geometria, geografia, lingue straniere, disegno, musica, fisica, scienze, ecc in sostanza tutte le materie, senza alcuna limitazione) lo sforzo di **introduzione** del **coding** e del **pensiero computazionale** nella scuola non potrà che avere successo. Un progetto basato su questi presupposti fornirà a studenti e docenti uno strumento di base che si troverà presto a diventare **la nuova penna del XXI secolo**. Uno strumento indispensabile per tutti, ma facile da imparare.

Dalle basi del coding alle applicazioni reali

Sebbene la galassia di applicativi appartenenti alla famiglia di Scratch (Scratch 2.0, Scratc Jr, Snap, Catrobat) permetta di acquisire tutti i concetti alla base del coding e del pensiero computazionale, i docenti delle scuole superiori o i docenti con conoscenze sull'argomento potranno decidere di orientare **successivamente** gli argomenti relativi al coding verso **linguaggi standard** come Javascript, PHP e MySQL nel caso si sia interessati allo sviluppo Web, o ProcessingJS nel caso si sia interessati alla grafica e al rendering o Python nel caso si sia interessati ad applicazioni leggere e multiuso. Tutti questi linguaggi, come i precedenti, sono utilizzabili usando ambienti sia locali che remoti che girano all'interno di una comune **browser web**, garantendo quindi la possibilità di uso di device diversi se dettata dall'approccio BYOD. Inoltre il **passaggio** dagli ambienti Scratch a quelli standard può essere reso più facile grazie all'uso di ambienti di transizione appositamente studiati come, ad esempio, **BloP** (http://www.blocklanguages.org).