



**CODING FOR
INCLUSION**

www.codinc.fun



Co-funded by the
Erasmus+ Programme
of the European Union

INTRODUZIONE	3	RISULTATI DELL'APPRENDIMENTO	6	RISULTATI DELL'APPRENDIMENTO PER GLI INSEGNANTI DELL'ISTRUZIONE SECONDARIA E DELLE SCUOLE PRIMARIE	6	RISULTATI DELL'APPRENDIMENTO PER GLI STUDENTI																																									
7	PENSIERO COMPUTAZIONALE	7	ABILITÀ COLLABORATIVE E SOCIALI	8	ABILITÀ CREATIVE	9	RISOLUZIONE DEI PROBLEMI E DI PROGETTAZIONE	9	OBIETTIVI PEDAGOGICI E DI ANIMAZIONE	9	MATRICE DEGLI OBIETTIVI DI APPRENDIMENTO PER LA FORMAZIONE DEGLI ALUNNI DELLA PRIMARIA	9	FORMAZIONE PRIMARIA OBIETTIVI DI APPRENDIMENTO	10	OBIETTIVI DI APPRENDIMENTO PER GLI ALUNNI (<12 ANNI)	10	PENSIERO COMPUTAZIONALE	10	COMPETENZE COLLABORATIVE E SOCIALI	11	ABILITÀ CREATIVE	11	PROBLEM SOLVING E CAPACITÀ PROGETTUALE	12	OBIETTIVI DI APPRENDIMENTO PER LA FORMAZIONE PRIMARIA	12	BACKGROUND	13	ABILITÀ DEL 21° SECOLO	13	PENSIERO COMPUTAZIONALE	14	PERCHÉ IL PENSIERO COMPUTAZIONALE?	14	PERCHÉ IL PENSIERO COMPUTAZIONALE È COSÌ IMPORTANTE?	14	PENSIERO CREATIVO	15	PERCHÉ IMPARARE A PROGRAMMARE?	15	PRINCIPI PEDAGOGICI FONDAMENTALI E FACILITAZIONE	18	FOGLI DI VALUTAZIONE	21	LAVORO	23	COLLEGAMENTI
23	MATERIALI	23	APPENDICE I: RIFERIMENTI	23																																											

INTRODUZIONE

Questa metodologia si rivolge agli insegnanti delle scuole secondarie e primarie o agli e-facilitatori.

Il documento spiegherà all'utente come coinvolgere gli studenti attraverso una metodologia pedagogica stimolante che vedrà gli studenti delle scuole secondarie (dai 15 anni in su) insegnare i principi del coding e l'educazione STEAM ai loro coetanei più giovani - alunni di età compresa tra gli 8 ei 12 anni.

Durante il progetto CODINC, insegniamo ai giovani e ai loro insegnanti le basi della programmazione. I nostri istruttori forniranno 10 ore di formazione riguardo il coding in classe.

Dopo, la formazione, gli studenti delle scuole secondarie istruiranno gli alunni nelle scuole primarie accompagnati dai loro insegnanti e dai nostri facilitatori.

Per gli insegnanti delle scuole secondarie, la conduzione di questo progetto li aiuterà ad aumentare il capitale scientifico, sviluppare un atteggiamento pedagogico e capacità di coaching e trasmettere le nozioni di base del coding agli alunni di età compresa tra 8 e 12 anni.

Il progetto CODINC si concentra sulla promozione dell'inclusione attraverso il peer to peer e l'educazione alla programmazione. I giovani e gli alunni coinvolti nel progetto CODINC provengono in particolare da quartieri a più alto rischio di esclusione sociale e più alti tassi di svantaggio rispetto ai loro coetanei. Il progetto CODINC mira ad aumentare la fiducia degli studenti di età compresa tra 15 e 18 anni provenienti da ambienti svantaggiati mostrando loro di aiutarli a scoprire il loro potenziale creativo, innovativo e di supporto. Questo aiuta gli studenti che potrebbero essere in difficoltà con l'identità e stanno cercando di trovare un posto in un mondo adulto. I giovani provenienti da aree svantaggiate hanno la sensazione di essere considerati cittadini di secondo grado e hanno bisogno della fiducia degli adulti per scoprire il loro potenziale creativo e innovativo.

CODINC adatterà, diffonderà e amplierà la pratica di apprendimento inclusivo di "Capital Digital", che è stata sviluppata e implementata dal partner del progetto Maks nei quartieri di Bruxelles, in Belgio, classificati come svantaggiati.

Capital Digital è un programma che impiega una metodologia di apprendimento peer-to-peer innovativo, formando i giovani provenienti da ambienti svantaggiati in STEAM e codifica per diventare e-facilitator dei loro coetanei più giovani. In pratica, Capital Digital ha addestrato

con successo giovani di 15-18 anni provenienti da ambienti svantaggiati per insegnare la programmazione ai loro coetanei di 10-12 anni. I giovani "e-facilitatori" hanno imparato a coinvolgere i bambini nelle STEAM e attività di coding in modo giocoso. Questo programma è diventato la prima esperienza lavorativa per gli e-facilitator aumentando la loro fiducia. Ha allargato i loro orizzonti al momento di decidere un percorso educativo e di carriera adeguato. Il programma gli ha consentito di connettersi gli uni con gli altri in modo costruttivo e di godere del ruolo di educare i loro coetanei più giovani. Inoltre, il progetto ha supportato efficacemente i giovani sviluppando il loro pensiero critico, la creatività, abilità digitali e collaborative. Ancora più importante, il metodo pedagogico peer-learning di Capital Digital ha un forte potenziale inclusivo e promuove l'educazione STEAM e l'inclusione di studenti svantaggiati sia all'interno che all'esterno della classe.

Il progetto CODINC adatterà e amplierà la metodologia Capital Digital (e gli strumenti di formazione) in diversi contesti educativi (scuole - Educazione formale) e lo diffonderà in 5 paesi europei, ovvero Belgio, Cipro, Germania, Italia e Spagna. Gli obiettivi specifici del progetto CODINC sono i seguenti:

1. Aumentare e migliorare la capacità degli insegnanti nel processo di insegnamento/apprendimento in giovani svantaggiati con un approccio educativo inclusivo basato sull'apprendimento tra pari.
2. Sostenere giovani svantaggiati nell'acquisizione e nello sviluppo di competenze collaborative, problem solving, fiducia in se stessi e creatività attraverso un programma di formazione peer-learning basato sul Coding
3. Promuovere lo sviluppo di una comunità di apprendimento europea "Coding for Inclusion" tra diversi attori e in diversi settori (educazione formale e non formale) in grado di sostenere i risultati del progetto e amplificarne l'impatto.

C'è bisogno in tutta Europa di sviluppare le competenze digitali di tutti i cittadini, in molti paesi il sistema educativo non è ancora completamente adattato all'uso di tecnologie sempre nuove. Nei quartieri dove ci sono più alti tassi di esclusione, c'è ancora più bisogno di usare gli strumenti digitali per promuovere pratiche inclusive, mentre i bambini e i giovani sono utenti attivi di Internet e dispositivi mobili, gli insegnanti hanno bisogno di sviluppare le loro abilità per supportare adeguatamente gli studenti con le nuove tecnologie. La tecnologia e il coding applicato alle STEAM offrono nuovi percorsi per l'insegnamento, basati sull'interesse di bambini e giovani, che gli consentono di essere produttori, e non solo consumatori, di contenuti digitali.

CODINC si rivolgerà direttamente a: Alunni della

-scuola primaria (da 8 a 12 anni) e studenti delle scuole secondarie (dai 15 ai 18 anni) provenienti da zone svantaggiate. Denominiamo "alunni" il primo gruppo e "studenti" il secondo gruppo.

- Insegnanti di scuola primaria e secondaria

- Formatori che lavorano con i giovani in contesti formali e non formali (scuole, centri giovanili, ONG, centri di formazione ecc.)

- Soggetti chiave: scuole, famiglie, organizzazioni della società civile, comunità locali, ONG, istituzioni educative e fornitori di apprendimento, autorità pubbliche, responsabili politici, ecc.

In questa metodologia, descriveremo un programma per l'implementazione di un corso di formazione di 10 ore riguardo il coding e le competenze pedagogiche per i giovani e un programma di formazione per gli studenti e gli insegnanti. Ci sono molte somiglianze tra il programma proposto agli studenti e agli alunni. Questo perché gli studenti dovrebbero essere formati sugli stessi moduli che useranno a turno per insegnare agli alunni. Il programma di formazione per studenti e insegnanti si concentra anche sulla metodologia pedagogica, tecniche di valutazione e motivazionali. Attraverso la formazione degli studenti è facile cogliere concetti, la loro fiducia in sé stessi nell'insegnare e credere nelle loro capacità insieme alla loro autostima crescerà e la paura di dover insegnare qualcosa che non capiscono davvero scomparirà.

Spero ti piaccia la lettura!

RISULTATI DELL'APPRENDIMENTO

Usando questa metodologia, i tuoi studenti completeranno una vasta gamma di compiti di apprendimento, dal pensiero computazionale alle abilità creative e sociali.

I risultati dell'apprendimento nella metodologia CODINC sono divisi in due sezioni, una per l'istruzione primaria e una per l'istruzione secondaria. Ogni sezione contiene risultati di apprendimento per gli alunni (8 fino a 12) o studenti (+15) che partecipano al programma e obiettivi separati per te come insegnante.

Puoi consultare le matrici degli obiettivi di formazione per un rapido riferimento su quale attività ti aiuta a completare ogni obiettivo di apprendimento.

Risultati dell'apprendimento per gli insegnanti dell'istruzione secondaria e delle scuole primarie

Come insegnante, in primo luogo, acquisirai preziose informazioni sul pensiero computazionale e sul coding, rafforzerai secondariamente le tue abilità pedagogiche.

Dopo aver completato questa metodologia, avrai una migliore comprensione di:

- Che cos'è il pensiero computazionale e perché è importante nella società?
- Quali sono le competenze chiave del XXI secolo per i tuoi studenti?
- In che modo le attività di coding favoriscono il pensiero computazionale, la creatività e le capacità di problem solving nei bambini?
- Quali competenze pedagogiche e sociali sono essenziali per i giovani a condurre attività di codifica con i loro coetanei più giovani?

Come professionista dell'educazione, imparerai ad applicare e condurre attività di formazione in riguardo il coding.

- Informatica offline.
- Esercizi collaborativi e creativi.
- Apprendimento attraverso la progettazione
- Attività pedagogiche e di animazione.

Gli insegnanti della scuola secondaria vengono formati insieme ai loro studenti; gli insegnanti della scuola primaria hanno una formazione separata.

Risultati dell'apprendimento per gli studenti

Gli studenti che partecipano al programma completeranno una vasta gamma di risultati di apprendimento, nei seguenti campi:

- Pensiero computazionale
- Abilità collaborative e sociali
- Creatività
- Problem solving e abilità progettazione
- Abilità pedagogiche e di motivazione di gruppo.

Naturalmente questi non sono gli unici risultati che puoi ottenere con questa metodologia, poiché i tuoi studenti useranno anche linguaggi verbali e visivi in modi creativi. Potrai anche applicare la matematica e decidere di collegare il progetto finale a un argomento a tua scelta, come la storia, la geografia, l'arte e così via.

Pensiero computazionale

I risultati dell'apprendimento per il pensiero computazionale sono divisi in due sezioni: i concetti che devono essere controllati dagli studenti e le loro pratiche cognitive corrispondenti.

Ci sono nove **concetti** chiave per il pensiero computazionale in questa metodologia, con la quale i giovani possono padroneggiare con successo la programmazione e creare i propri giochi o programmi.

Concetto	Descrizione del
Algoritmi	Una serie scritta di istruzioni per un dispositivo computazionale; ad esempio, per un gioco, un'app o anche una ricetta di cucina.
Sequenze	Istruzioni fornite in ordine discreto, l'una dopo l'altra. Il computer li esegue da cima a fondo.
Ripetizione e cicli	Ripetere un sottoinsieme di istruzioni più volte (o infinitamente) viene chiamato "ciclo".
Eventi e selezione	Indicazione di quando un evento dovrebbe aver luogo; per esempio, il gatto inizia a muoversi quando viene premuto il pulsante "start".
Condizionatori e operatori logici	Lasciare che il computer prenda una decisione. Se succede qualcosa, allora un evento dovrebbe aver luogo, o non dovrebbe aver luogo.
Operatori matematici	Algoritmi che richiedono matematica, come moltiplicazione, addizione, ... ad esempio per ridurre la velocità di una palla nel tempo.
Variabili e gestione dati	Le Variabili sono caselle in cui è possibile memorizzare numeri (o testi). Queste caselle vengono quindi utilizzate nel codice, ad esempio per mostrare il punteggio corrente del giocatore.
Funzioni	Riutilizzare un sottoinsieme di codice, ad esempio per andare avanti, con un'etichetta nome.

Di conseguenza, ci sono sei **pratiche di pensiero**, che contribuiscono a diventare un abile programmatore e pensatore computazionale. Queste pratiche di pensiero non sono solo utili nel campo della tecnologia, ma si trasferiscono molto bene in altri campi del pensiero e della vita, come la risoluzione di problemi creativi.

pratica di pensiero	Descrizione della
Strategie di lavoro	Gli studenti suddividono il loro lavoro in piccoli passi e tornano ai passaggi precedenti nel loro modo di pensare per migliorare il loro programma. La codifica non è un processo lineare, ma una spirale che continua a ripetersi.
Test e debug	Gli studenti possono testare un gioco o un progetto contro le loro aspettative e trarne insegnamenti per migliorarlo. Possono identificare problemi (bug) e usare strategie per risolverli (debugging).
Riutilizzare e remixare	Gli studenti possono imparare dai progetti fatti da altri; riutilizzare pezzi di codice o processi di pensiero; e remixare soluzioni o progetti esistenti in qualcosa di nuovo.
Astrazione	Gli studenti possono trasferire le lezioni apprese da un progetto in schemi astratti.
Modularizzazione	Gli studenti possono rompere i progetti in parti più piccole (movimento, velocità, punteggio ...) e riutilizzare quelle parti nel loro lavoro.
Raccolta e gestione delle informazioni	Gli studenti possono identificare le fonti di informazioni e cercare soluzioni per i loro problemi in vari luoghi, ad esempio con colleghi o online.

Competenze collaborative e sociali

La metodologia CODINC sceglie esplicitamente un approccio collaborativo, peer-to-peer, in cui gli studenti più grandi (istruzione secondaria) lavorano insieme per insegnare la codifica agli alunni più piccoli (istruzione primaria).

I tuoi studenti lavoreranno sulle loro capacità di collaborazione, in quanto sapranno (1) lavorare come una squadra e imparare insieme. Impareranno a (2) negoziare tra loro, così come con gli alunni con cui lavoreranno. Le attività di insegnamento richiedono non solo una conoscenza dell'argomento, ma forniscono anche grandi spunti pedagogici sul modo di pensare dei più giovani. Successivamente, impareranno a (3) descrivere i propri processi mentali, in una lingua appropriata per il proprio pubblico di riferimento (che si tratti del gruppo di alunni più giovani o dei propri coetanei). Infine, impareranno gli uni dagli altri e dal gruppo più giovane, attraverso (4) esperienza e problem solving collaborativo.

Abilità creative

Durante tutta la formazione, le abilità creative vengono attivamente incoraggiate. I tre aspetti principali della creatività (secondo il quadro della creatività sistemica di LEGO) sono messi in atto: (1) esplorare possibilità, idee e progetti di altre persone; (2) combinare competenze, idee, progetti esistenti ... in nuovi sforzi creativi; e (3) trasformare le proprietà esistenti in qualcosa di nuovo.

Problem Solving and Abilità di progettazione

Infine, la metodologia affronta anche la risoluzione dei problemi e le capacità di progettazione. Gli studenti impareranno come (1) identificare un nuovo problema e i suoi componenti. Acquisiranno anche strategie per (2) generare nuove idee attraverso il brainstorming e (3) come implementare queste idee. Ovviamente, (4) valuteranno la loro soluzione e rifletteranno sull'impatto. Infine, miglioreranno la loro soluzione e rivaluteranno il problema in una successiva (5) iterazione del processo.

Obiettivi pedagogici e di animazione

In questo programma, gli studenti acquisiranno una grande quantità di abilità pedagogiche e approfondimenti nell'animazione di gruppo. Apprenderanno il programma di formazione di dieci ore, ma ancor più lavorando attivamente con un gruppo di alunni.

Acquisiranno conoscenze e abilità riguardanti (1) la conduzione di attività di gruppo; (2) gestire la motivazione individuale e di gruppo negli alunni e usare gli energizzanti per aumentare il morale; (3) valutare attività, processi e le proprie prestazioni come animatori; (4) stabilire regole di condotta di gruppo; e (5) infine organizzazione e intuizioni pratiche.

Matrice degli obiettivi di apprendimento per gli alunni della scuola primaria

Le diverse attività della metodologia CODINC affronteranno ovviamente una selezione variabile di questi obiettivi di apprendimento. Durante tutto il programma, tutti gli obiettivi di apprendimento saranno soddisfatti. Puoi trovare una comoda panoramica di tutti i diversi obiettivi di apprendimento e di quali attività sono acquisite nella **matrice per l'istruzione secondaria**.

Matrice degli obiettivi di apprendimento in allegato Cfr.

Obiettivi di apprendimento per l'istruzione primaria: alunni (<12 anni)

Gli alunni che partecipano al programma completeranno una grande varietà di obiettivi di apprendimento, nei campi del (1) pensiero computazionale; (2) capacità collaborative e sociali; (3) capacità creative; (4) e capacità di problem solving e progettazione.

Naturalmente questi non sono gli unici obiettivi che puoi ottenere con questa metodologia, poiché i tuoi studenti useranno anche linguaggi verbali e visivi in modi creativi. Potrai anche applicare la matematica e decidere di collegare il progetto finale a un argomento di tua scelta, come la storia, la geografia, l'arte e così via.

Pensiero computazionale

Gli obiettivi di apprendimento per il pensiero computazionale sono divisi in due sezioni: i concetti che devono essere controllati dagli alunni e le pratiche di pensiero corrispondenti.

Ci sono nove concetti chiave per il pensiero computazionale in questa metodologia, con la quale i bambini possono padroneggiare con successo la programmazione e creare i propri giochi o programmi.

concetto	Descrizione del
Algoritmi	Una serie scritta di istruzioni per un dispositivo computazionale; ad esempio, per un gioco, un'app o anche una ricetta di cucina.
Sequenze	Istruzioni fornite in ordine discreto, l'una dopo l'altra. Il computer li esegue da cima a fondo.
Ripetizione e cicli	Ripetendo un sottoinsieme di istruzioni più volte (o infinitamente) viene chiamato un ciclo.
Eventi e selezione	Indicazione di quando un evento dovrebbe aver luogo; per esempio, il gatto inizia a muoversi quando viene premuto il pulsante "start" in una partita.
Condizionatori e operatori logici	Lasciare che il computer prenda una decisione. Se succede qualcosa, allora un evento dovrebbe aver luogo, o non dovrebbe aver luogo.
Operatori matematici	Algoritmi che richiedono matematica, come moltiplicazione, addizione, ... ad esempio per ridurre la velocità di una palla nel tempo.
Variabili e gestione dati	Le variabili sono caselle in cui è possibile memorizzare numeri (o testi). Queste caselle vengono quindi utilizzate nel codice, ad esempio per mostrare il punteggio corrente del giocatore.
Funzioni	Riutilizzare un sottoinsieme di codice, ad esempio per andare avanti, con un'etichetta nome.

Di conseguenza, ci sono sei **pratiche di pensiero**, che contribuiscono a diventare un abile programmatore e pensatore computazionale. Queste pratiche di pensiero non sono solo utili

nel campo della tecnologia, ma si trasferiscono tremendamente bene in altri campi del pensiero e della vita, come la risoluzione di problemi creativi.

pratica di pensiero	Descrizione della
Strategie di lavoro incrementali e iterative Gli	alunni suddividono il loro lavoro in piccoli passi e ritornano ai passaggi precedenti nel loro modo di pensare per migliorare il loro programma. La codifica non è un processo lineare, ma una spirale che continua a ripetersi.
Test e debug	Gli studenti possono testare un gioco o un progetto in base alle loro aspettative e trarne insegnamenti per migliorarlo. Possono identificare problemi (bug) e usare strategie per risolverli (debugging).
Riutilizzare e remixare	Gli studenti possono imparare dai progetti fatti da altri; riutilizzare pezzi di codice o processi di pensiero; e remixare soluzioni o progetti esistenti in qualcosa di nuovo.
Astrazione	Gli alunni possono trasferire le lezioni apprese da un progetto in schemi astratti.
Modularizzazione	Gli studenti possono rompere i progetti in parti più piccole (movimento, velocità, punteggio ...) e riutilizzare quelle parti nel loro lavoro.
Raccolta e gestione delle informazioni	Gli alunni possono identificare le fonti di informazioni e cercare soluzioni per i loro problemi in vari luoghi, ad esempio con colleghi o online.

Competenze collaborative e sociali

La metodologia CODINC sceglie esplicitamente un approccio collaborativo, in cui gli alunni esplorano, sperimentano e creano insieme un progetto finale. Per questo motivo viene acquisita e praticata una grande varietà di abilità sociali e collaborative.

I bambini imparano a (1) lavorare insieme su un gioco nel progetto finale, che gli insegna a (2) negoziare tra loro, così come (3) descrivere i propri processi mentali.

Imparano anche a (4) presentare i propri progetti di fronte ai gruppi, e (5) dare e ricevere feedback reciproci. Infine, imparano non solo da soli, ma ancor più da (6) esperienze condivise.

Abilità creative

Durante tutto il percorso formativo, le abilità creative vengono attivamente incoraggiate. I tre aspetti principali della creatività (secondo il quadro della creatività sistemica di LEGO) sono messi in atto: (1) esplorare possibilità, idee e progetti di altre persone; (2) combinare competenze, idee, progetti esistenti ... in nuovi sforzi creativi; e (3) trasformare le proprietà esistenti in qualcosa di fresco.

Problem Solving e Abilità di Progettazione

Infine, la metodologia affronta anche la risoluzione dei problemi e le capacità di progettazione. Gli studenti impareranno come (1) identificare un nuovo problema e i suoi componenti. Acquisiranno anche strategie per (2) generare nuove idee attraverso il brainstorming e come (3) implementare queste idee. Ovviamente, (4) valuteranno la loro soluzione e rifletteranno sull'impatto. Infine, miglioreranno la loro soluzione e rivaluteranno il problema in una successiva (5) iterazione del processo.

Matrice degli obiettivi di apprendimento per la matrice di istruzione primaria

Le diverse attività della metodologia CODINC affronteranno ovviamente una selezione variabile di questi obiettivi di apprendimento. Durante tutto il programma, tutti gli obiettivi di apprendimento saranno soddisfatti. Puoi trovare una comoda panoramica di tutti i diversi obiettivi di apprendimento e di quali attività sono acquisite nella **matrice per l'istruzione primaria**.

Gli obiettivi di apprendimento cf. allegato

BACKGROUND

Competenze del 21° secolo

Il World Economic Forum ha svolto un'indagine sulle competenze più necessari nel 2020 rispetto al 2015.

La creatività diventerà uno delle prime tre competenze di cui i lavoratori avranno bisogno. Con la valanga di nuovi prodotti, nuove tecnologie e nuovi modi di lavorare, i lavoratori dovranno diventare più creativi per beneficiare di questi cambiamenti.

Allo stesso modo, l'ascolto attivo, considerato oggi un'abilità di base, scomparirà completamente. L'intelligenza emotiva, che non è presente nella top 10 di oggi, diventerà una delle migliori competenze necessarie a tutti. (<https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution/>)

In Europa, l'OCSE ha definito una serie di competenze che bambini, giovani e adulti avranno bisogno per prosperare nella società moderna: le abilità del 21 ° secolo. Si tratta di competenze interconnesse, come la risoluzione di problemi, la collaborazione, il pensiero creativo e il pensiero computazionale, che dovrebbero far parte di tutta l'istruzione, sostenere i cittadini del futuro e organizzarli per l'apprendimento permanente.

Queste capacità sono:

1. Pensiero critico
2. Pensiero creativo
3. Risoluzione dei problemi
4. Pensiero computazionale
5. Abilità informatiche
6. IT di base
7. Alfabetizzazione mediatica
8. Comunicazione
9. Collaborazione
10. Competenze sociali e culturali
11. Autoregolamentazione

Nella metodologia CODINC, miriamo esplicitamente a migliorare e sostenere queste capacità del XXI secolo, concentrandosi in particolare sul pensiero computazione, problem solving, pensiero creativo, collaborazione e comunicazione. Anche le altre abilità sono coinvolte nel progetto, poiché sono tutte interconnesse.

Pensiero computazionale

Perché il pensiero computazionale?

Quando vedi il termine "Pensiero computazionale", potresti pensare che sia una parte dell'informatica. Tuttavia, è molto più ampio. Il termine è stato coniato da Jeanette Wing nel 2006, alla Carnegie Mellon University.

Puoi trovare molti video e risorse sull'argomento, ma ti consigliamo di consultare questa breve spiegazione di Google: <http://bit.ly/CTbyGoogle>.

Il pensiero computazionale non è solo una competenza, ma una gamma di concetti, applicazioni, strumenti e strategie di pensiero utilizzate per risolvere i problemi. Questi possono essere usati per scoprire il genoma del DNA umano o per analizzare gli scritti di Shakespeare. Puoi praticare il Pensiero computazionale senza nemmeno toccare un computer. Ma cos'è allora?

*"Il pensiero computazionale è un approccio al problem solving. Quindi sta smontando un problema e immaginando come attaccarlo, usando ciò che sappiamo del calcolo."
(Diane Main, Direttore dell'apprendimento, dell'innovazione e del design, The Harker School, Upper School)*

Google definisce quattro aspetti principali del pensiero computazionale:

1. Decomposizione: rompere un problema in parti più piccole;
2. Riconoscimento del modello: trovare somiglianze e differenze tra le diverse parti, essere in grado di fare previsioni;
3. Astrazione: la capacità di trovare i principi generali dietro le parti e gli schemi nei problemi;
4. Progettazione di algoritmi: sviluppo delle istruzioni passo passo per risolvere diversi problemi.

Perché il pensiero computazionale è così importante?

In una società che diventa sempre più complessa e focalizzata sulla tecnologia, è fondamentale per i bambini e i giovani imparare a pensare in modo critico e in grado di controllare e creare la propria esperienza digitale. Piuttosto che i consumatori della tecnologia digitale, vogliamo che i bambini diventino i produttori e insegnino loro una comprensione critica. Il pensiero computazionale diventa quindi parte di una più ampia cultura dei media.

È qui che entra in gioco il pensiero computazionale. Non solo aumenterà l'attuale divario digitale se ignoriamo questa necessità; passerà dall'uso della tecnologia alla capacità di creare tecnologia. Gli "abbienti e non abbienti" diventeranno i "crea e non crea". Ci sarebbe una

divisione tra coloro che possono controllare il proprio uso della tecnologia e generare soluzioni ai loro bisogni; e quelli che semplicemente consumano senza pensiero critico.

Un vantaggio è che il contatto precoce con le sfide del pensiero computazionale ispirerà i bambini a scegliere più campi di studio correlati a STEAM (Scienza-Tecnologia-Ingegneria-Arti-Matematica), in particolare le ragazze, che hanno un feedback positivo e una maggiore autostima in questi campi.

Certo, c'è anche il bisogno professionale. Il mercato del lavoro avrà bisogno di un numero sempre maggiore di dipendenti formati nelle TIC e nella codifica. Anche ora c'è già una carenza. Ma non solo i lavori di programmazione o di tecnologia necessiteranno di un pensiero computazionale. Con l'Intelligenza Artificiale e la Robotica che assumono compiti "semplici", saranno necessari sempre più impiegati con capacità creative e di risoluzione dei problemi.

Pensiero creativo

L'abilità del pensiero creativo è una sfida da definire: cos'è la creatività? Nel saggio su "Systematic Creativity in the Digital Realm" (Ackermann et al., 2009), il gioco creativo è suddiviso in tre attività principali: esplorazione, combinazione e trasformazione.

Combinare - inventare idee e manufatti nuovi, sorprendenti e preziosi attraverso la combinazione di idee e oggetti esistenti.

Esplorare - ampliare la nostra comprensione di un'area o di un dominio creativo presentando idee e manufatti nuovi, sorprendenti e di valore.

Trasformare - trasformare il modo in cui vediamo o comprendiamo il mondo attraverso la creazione di idee e manufatti nuovi, sorprendenti e preziosi.

Nell'insieme delle abilità del 21° secolo, l'abilità del pensiero creativo è suddivisa in diverse abilità più piccole, come ad esempio (1) conoscere e utilizzare tecniche creative; (2) lasciando il sentiero battuto; (3) vedere nuove connessioni e combinazioni; (4) il coraggio di prendere rischi (calcolati); (5) vedere gli errori come possibilità di apprendimento; e (6) avere un atteggiamento aperto, investigativo.

Perché imparare a codificare?

Ci sono molti modi per sviluppare le competenze del 21° secolo; in particolare, il pensiero computazionale e la creatività. Tuttavia, riteniamo che insegnare ai bambini e ai giovani a codificare sia uno dei migliori percorsi da seguire. Questo perché stimola le loro capacità di identificare, estrapolare e creare modelli. I bambini che imparano a programmare hanno una maggiore comprensione dei sistemi e di come sono progettati. Analizzano continuamente i

problemi e escogitano nuove soluzioni per loro. La codifica offre anche ai bambini gli strumenti per iniziare a creare da soli.

Comprendere il codice e i principi alla base dei linguaggi di programmazione offre un vantaggio significativo per i futuri datori di lavoro e coloro che cercano lavoro. Ma ciò non significa che il vantaggio della codifica sia limitato agli scopi professionali.

Imparare a programmare permette ai bambini di diventare creatori e produttori in un mondo digitale. Invece di consumare la tecnologia su base giornaliera, con un atteggiamento acritico, ora possono iniziare a costruire il proprio futuro digitale. Invece di consumatori di app, diventano sviluppatori di app. Che sia professionale o solo per divertimento, l'apprendimento del codice offre un grande vantaggio.

Perché la codifica di giovani e bambini svantaggiati è importante?

L'accesso ai computer è sufficiente per ridurre il divario digitale tra classi sociali più elevate e gruppi svantaggiati? Molte relazioni evidenziano la necessità di imparare a usare il computer o altri dispositivi.

L'accesso alle informazioni e il possesso di adeguate competenze informative sono diventati risorse chiave nella vita economica e sociale e possono dare origine a una nuova fonte di stratificazione nella società. Il "gap di conoscenza" è ben lungi dall'essere un nuovo concetto (Tichenor, Donohue e Olien 1970), ma potrebbe raggiungere una maggiore importanza nell'era digitale (Bonfadelli 2002). La pervasività della tecnologia digitale e delle TIC ha raggiunto livelli senza precedenti nella storia della vita umana. Le informazioni sono sempre più accessibili su Internet e dispositivi digitali. Per la maggior parte degli studi, il cosiddetto primo divario digitale (cioè il divario socioeconomico nell'accesso alla tecnologia) sta diminuendo e si ritiene che scompaia a fianco della maturazione del processo di digitalizzazione. Tuttavia, nonostante l'ipotesi popolare secondo cui la digitalizzazione e Internet porterebbero ad un aumento generale e uguale della conoscenza della popolazione, recenti evidenze empiriche evidenziano notevoli disparità sociali nel modo in cui 4 analisi aggiuntive includono un termine quadratico dell'indice di possesso delle TIC per catturare possibili le non linearità nella relazione con le abilità di navigazione confermano in parte questa interpretazione, in quanto l'associazione negativa viene rilevata solo a valori alti dell'indice. Troppa tecnologia in giro per l'ambiente domestico non si traduce in un uso migliore e in competenze più elevate. Le classi sociali più elevate possono impiegare le loro risorse più elevate e il capitale culturale per trarre maggiore vantaggio dalle nuove tecnologie e quindi ottenere un accesso più rapido e migliore alle informazioni.

Gli studenti delle classi sociali più elevate traggono più vantaggio dalle tecnologie rispetto ai loro coetanei svantaggiati che utilizzano il computer principalmente come giochi di consumo, utilizzando Facebook e acquistando prodotti su Internet.

Aumentare le competenze digitali dei gruppi svantaggiati è estremamente urgente per ridurre questo divario digitale e favorire l'integrazione sociale e la mobilità. La produzione di prodotti multimediali con dispositivi mobili o computer cambia la relazione tra lo studente e i dispositivi. Quando produci contenuti TIC, diventi un prosumer e impari a considerare il computer uno strumento per l'apprendimento permanente. Questo è anche uno degli obiettivi indiretti di questo progetto.

Usare la codifica per elevare la posizione dei gruppi svantaggiati e aumentare l'autostima e la fiducia.

Dare ai giovani svantaggiati un ruolo da svolgere nell'ambiente scolastico è dare loro voce e senso di appartenenza alla comunità scolastica. La metodologia peer-to-peer è una scelta appropriata per aumentare il loro benessere e responsabilizzare i giovani. I giovani di origine e di provenienza da rifugiati di paesi terzi non possono offrire lo stesso capitale scientifico o capitale sociale dei loro pari, lasciandoli a rischio di tassi di abbandono più elevati e di una minore autostima. Il progetto CODINC offre ai giovani provenienti da ambienti svantaggiati quali il capitale scientifico e sociale di provenienza di rifugiati o di paesi terzi la formazione per il ruolo di mediatore / facilitatore di STEAM per i loro coetanei. Ciò consente di aumentare la fiducia e l'autostima come attori nel mondo digitale e leader nella propria comunità. L'esperienza CODINC può anche migliorare la relazione insegnante-studente in classe, in quanto gli insegnanti hanno la possibilità di vedere gli studenti impegnati nell'educazione in un modo diverso e di sperimentare un ruolo diverso. Ciò può anche contribuire ad aumentare il livello di cooperazione e coesione sociale in classe.

Principi Pedagogici e Facilitazioni

Nella metodologia CODINC, I principi pedagogici sono basati sulle quattro P inviate da Mitch Resnick (MIT, LifeLong Kindergarten e il creatore del linguaggio di programmazione visuale di Scratch): Projects, Passion, Peers and Play.

Puoi trovare una breve introduzione video riguardo l'Apprendimento Creativo e le quattro P qui: <http://bit.ly/resnickfourp>

Progetti

I bambini e i ragazzi sono più motivati (e imparano meglio) quando stanno lavorando a progetti che amano e che hanno un significato per loro. Pertanto, il programma non si limita ai tutorial o ai seguenti esempi; ma conduce alla costruzione di un progetto personale. I bambini creano il loro gioco, uno con cui sono venuti in mente e di cui possono essere orgogliosi.

Passione

L'apprendimento è divertente, o almeno dovrebbe esserlo. Nel programma di 10 ore, uno degli obiettivi principali è sia per i bambini che per i giovani divertirsi e amare quello che stanno facendo. Essere appassionato di una materia o di un progetto porta a una maggiore motivazione, a un maggiore coinvolgimento e ad un apprendimento più profondo.

Peers

Non è un caso che il programma applichi un approccio peer-to-peer all'apprendimento. Lavorare, giocare e lavorare insieme è uno dei processi di apprendimento più forti che ci siano. Non solo gli studenti secondari insegneranno ai loro coetanei più giovani, ma impareranno anche da loro. L'apprendimento solitario non è solo più lento e restrittivo; è noioso.

Gioco

Questo programma non è composto da lezioni tradizionali. L'obiettivo è giocare con la tecnologia, con i concetti e gli uni con gli altri. Attraverso sperimentazioni attive e giochi divertenti, i bambini impareranno più di quanto vorrebbe seduti e ascoltando. Pertanto, l'atmosfera ideale dovrebbe essere giocosa e divertente; non limitativo o tradizionale.

Il coding e il movimento per la formazione dei maker si basano sui principi pedagogici di pedagoghi come Jean Piaget e Paolo Freire e partono dal seguente punto di vista:

I bambini apprendono naturalmente

Il loro desiderio di comprendere il mondo che li circonda, di acquisire abilità e competenza in quel mondo e di svolgere un ruolo significativo in esso, è forte quanto il loro desiderio di cibo, calore, conforto e amore. Come lo fanno? Imparano attraverso l'immersione in una cultura umana e attraverso acuta osservazione, gioco, imitazione e partecipazione all'interno di quella cultura. Vedono ciò che fanno le persone più anziane e più abili e hanno una forte spinta a fare le stesse cose. Infatti, i bambini piccoli esplodono di rabbia quando non sono autorizzati a fare cose che vedono fare gli altri. Vogliono unirsi a quel ballo chiamato "vita", non solo sedersi in disparte. I bambini sono come scienziati; sviluppano teorie, formulano ipotesi, li mettono alla prova e rivedono o abbandonano le loro teorie, se necessario. L'insegnamento

può aiutare a imparare quando supporta e consente alle persone di fare ciò che vogliono, quando aiuta a capire ciò che cercano di capire - ma solo quando tale intervento è richiesto, richiesto, invitato o in qualche modo accettato dallo studente. L'insegnamento non invitato, indesiderato e non richiesto non aiuta l'apprendimento. Lo ostacola.

Facilitare il coding e attività offline

Fare e creare può essere difficile. Per sostenere la creatività e l'innovazione, dobbiamo aiutare gli studenti a considerare queste sfide come una parte normale del processo e a imparare a navigare con successo con la pratica. Per insegnare la codifica, aiuta a capire le fasi chiave del processo, le reazioni degli studenti in ogni fase, le risposte utili degli insegnanti e le abilità insegnate.

Questo approccio può trasformare la mentalità tradizionale della classe in cui gli insegnanti passano la conoscenza in un modo meno tradizionale di lavorare dove l'insegnante diventa un facilitatore che allena i propri alunni per imparare.

Una singola valutazione sommativa delle conoscenze acquisite può trasformarsi in una serie di valutazioni minori utilizzando prodotti creati dagli studenti che si basano l'uno sull'altro e culminano in un progetto finale più sostanziale.

Durante le attività di scollegamento e codifica, bambini e giovani sono stimolati a provare alcune cose e vedere se funzionano. La prova e l'errore possono portare al fallimento ma anche all'iterazione e all'innovazione.

Tutti gli studenti possono essere sfidati e possono crescere, e nel perseguimento di questo obiettivo costruiamo uno spazio sicuro in cui il fallimento è una possibilità fondamentale. Un esperimento fallito non è una fine, come in una valutazione standardizzata, ma piuttosto una sfida da provare di nuovo.

Quando gli studenti sono liberi di inventare e creare, iniziano a vedere la tecnologia come un mezzo per risolvere problemi del mondo reale e portare il loro apprendimento al livello successivo.

Le nostre risorse sono progettate specificamente come strumenti di apprendimento pratico per aiutare gli studenti di oggi a sviluppare competenze per l'economia creativa e digitale: pensiero critico, collaborazione, comunicazione, curiosità, problem solving e invenzione.

Devo, come insegnante, conoscere il coding prima dell'inizio delle attività?

In primo luogo, durante questo progetto, i formatori addestreranno i giovani e i loro insegnanti nella base della codifica. I formatori verranno per 10 ore di codice in classe. Successivamente, gli studenti della scuola secondaria frequentano le scuole elementari. Gli insegnanti delle scuole primarie sono formati per facilitare gli studenti insieme agli alunni. Quindi, non è necessario sapere come codificare prima dell'avvio del progetto. Dovresti comunque essere disposto a provare e sperimentare un nuovo approccio di apprendimento in classe. Se uno studente finisce presto, invitali a migliorare il progetto che ha realizzato

(aggiungendo un punteggio, aumentando la difficoltà, cambiando il tema del gioco ...) o creando un secondo, più difficile.

Alla fine, concedi 10 minuti in modo che gli studenti possano testare i rispettivi progetti e fare una rapida discussione su ciò che hanno imparato finora.

SOGLIE

Durante il progetto Capital Digital, le scuole erano favorevoli per consentire attività di coding gratuite se venivano forniti i materiali e l'hardware per insegnarlo. È importante che i formatori che realizzano il progetto CODINC abbiano una preparazione buona e strutturata con gli insegnanti e le scuole per allineare al meglio i risultati dell'apprendimento e gli orari per gli studenti prima di iniziare il progetto.

Insegnanti

Gli insegnanti hanno supportato l'utilizzo del progetto Capital Digital una volta che hanno visto quali sono le attività e una volta che vedono che gli alunni sono entusiasti delle attività.

Anche i Dirigenti Scolastici erano favorevoli a fornire attività di coding in classe: il coding è richiesto e ai genitori piace il fatto che il coding sia introdotto nella scuola. L'introduzione del coding e del pensiero computazionale ha contribuito ad aumentare il prestigio della scuola. Questo è eccellente per le scuole che sono state identificate nelle aree escluse. È importante che i formatori parlino con il direttore prima di iniziare il progetto CODINC nella scuola per ottenere il loro supporto in modo scritto, facendo un accordo.

Dire no al caos in classe, creare qualche struttura

Gli insegnanti hanno paura del caos in classe, perché ai loro occhi la non organizzazione è un ostacolo per l'insegnamento. Ecco perché gli esercizi devono essere dati in modo molto strutturato. È importante pensare al modo in cui organizzarsi l'aula e come saranno programmati i diversi esercizi. Nel nostro programma di formazione con i giovani dovremmo discuterne. La struttura della classe può essere cambiata per questo giorno o no? Come posso far lavorare il gruppo con gli alunni senza modificare la struttura della classe? C'è un'altra stanza che può essere utilizzata per le attività? Questa è una cosa importante da discutere con gli insegnanti prima dell'inizio del progetto.

Discutere i tempi prima di iniziare il progetto

Per le attività di codifica è interessante avere alcuni blocchi di due-tre ore di lavoro, invece di corsi da 50 minuti come da abitudine nelle scuole secondarie (in Belgio? - è così anche in Spagna, a Cipro e in Germania).

5 blocchi di 2 ore sono ideali per pianificare le attività con i giovani.

Alunni

Fai una prova, prima di iniziare il progetto

Introdurre l'attività di coding in classe, raccontando agli studenti cosa possono aspettarsi dalle attività in questo progetto e in che modo verrà sviluppato. Lascia che provino qualcosa di pratico, così vedono che è divertente. La maggior parte degli alunni è felice quando ci sono progetti in classe, cambia la routine quotidiana e dai corsi che ricevono. Chiedi quali sono le

loro paure per lavorare con gli alunni e discutetene. Se hai la possibilità, prova a cercare con loro che tipo di scuola primaria vogliono lavorare.

Modelli di ruoli per piccolo Fratelli e sorelle

Nella maggior parte delle famiglie svantaggiate, i legami tra fratelli sono molto forti e i "grandi" fratelli e sorelle amano aiutare i più piccoli.

Allegato 1: Matrice dei risultati dell'apprendimento

WORKSHEETS

https://docs.google.com/document/d/1wEyC7jfkImP7ks15qr_cx7pFo7kqc5Y0X0lhqYg5U6c/e/dit#

LINKS

Lightbot (online en apps) - <http://lightbot.com/hour-of-code.html>

Hour of Code: Star Wars - <https://code.org/starwars>

Hour of Code: Minecraft - <https://code.org/minecraft>

MATERIALI

Item	Estimated price / item	Number	Estimated price total €
Makey Makey Kit	Circa 55 euro per kit	10	550
Computers / laptops		10	
Tablets (opzionale)	100	10	1000
Crafting materials			
Writing materials			

APPENDIX I: REFERENCES

Ananiadou, K. (2009). *21st Century Skills and Competences for New Millenium Learners in OECD Countries*.

Brennan, K., & Resnick, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Raspberry Pi Foundation. (2017). *CodeClub Curriculum*. Retrieved March 6, 2018, from <https://codeclubprojects.org/en-GB/curriculum/>

<http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden>

The second digital divide in Europe. A crossnational study on students' digital reading and navigation skills* Davide Azzolinia[†] , Antonio Schizzerottoa,b

<https://irvapp.fbk.eu/wp-content/uploads/2017/09/FBK-IRVAPP-Working-Paper-No.-2017-02.pdf>

