

Computational Thinking and Humanities

Francesco Maiorana^{1,2,3,4}

¹ Università degli Studi di Catania, Dipartimento di Scienze della Formazione

² Istituto d'Istruzione Superiore Statale G.B. Vaccarini di Catania

³ Scientix Ambassador

⁴ CoderDojo champion

fmaioran@gmail.com

Abstract. Un profondo cambiamento a livello mondiale sta portando a considerare l'introduzione del pensiero computazionale fin dalla scuola primaria come competenza di base a fianco della lettura, della scrittura e dell'acquisizione di competenze in aritmetica. Alla base di questo profondo cambiamento c'è la consapevolezza che queste abilità e competenze devono diventare patrimonio culturale di tutti gli studenti ed accompagnarli lungo tutto l'arco della vita. Il lavoro presenta l'esperienza biennale di un corso universitario che introduce il pensiero computazionale a studenti dell'ultimo anno di un corso triennale in discipline umanistiche. Partendo dalla descrizione dei criteri di progettazione del corso, dei suoi obiettivi e degli approcci pedagogici, il lavoro prosegue descrivendo contenuti, abilità, competenze, micro e macro progetti assegnati. Una valutazione qualitativa del corso, basata su questionari somministrati agli studenti e una riflessione sulla necessità di nuovi curricula dalla scuola primaria ai corsi universitari conclude il lavoro.

Keywords: Computational Thinking, STEAM, Pensiero algoritmico, Coding.

1 Introduzione

Un profondo rinnovamento a livello mondiale vede impegnate le maggiori organizzazioni per progettare nuovi curricula e definire frameworks di competenze che devono supportare gli studenti del 21° secolo durante il loro processo di apprendimento lungo tutto l'arco della vita. Tra i frameworks ricordiamo a livello Europeo i frameworks centrati sul pensiero computazionale [1] e i frameworks che coprono le competenze chiave a tutti i livelli: dalla cittadinanza (DigComp) [2], agli educatori (DigCompEdu) [3], dalle organizzazioni educative (DigCompOrg) [4] ai consumatori (DigCompConsumers) [5], dalle istituzioni educative superiori (OpenEdu) [6], all'imprenditorialità (EntreComp) [7]. Ricordiamo anche negli Stati Uniti il framework for 21st century learning [8] realizzato dalla Partnership for 21st century learning e dall'OECD [9].

Sul fronte dei curricula il profondo rinnovamento è partito da nazioni come l'Inghilterra, seguita in Europa da altre nazioni, che ha introdotto obbligatoriamente l'insegna-

mento dell'Informatica fin dalla scuola elementare [10]. Negli stati Uniti, un vasto movimento sta spingendo per l'introduzione della Computer Science come disciplina a sé stante fin dalla scuola elementare con curricula come l'Advanced Placement Computer Science Principles [11] o iniziative come [12] che hanno avuto una diffusione a livello mondiale anche in Europa [13] e in Italia [14] [15].

In Italia le nuove linee guida per la scuola elementare [16] prevedono il “Pensiero Computazionale” come un aspetto di apprendimento basato sulla lingua e sulla matematica con cui s'intende: “un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia”. Le linee guida sono predisposte in accordo con le iniziative a livello europeo, sia istituzionale [17] sia associativo [18]. Seppur non esistono ancora curricula e linee guida che prevedano l'Informatica come disciplina a sé stante in tutti gli ordini di scuola e di studi universitari, linee guida auspiccate da tempo e richieste a più voci, il fervore a livello italiano è indubbio. Questo fervore trova nei docenti italiani di tutte le scuole di ogni ordine e grado il volano principale [19]. Il fervore è tangibile anche a livello europeo dove l'Italia è impegnata in prima linea, come dimostrato dai progetti presentati nell'Open Book of Educational Innovation [20]. Numerosi progetti europei vedono l'Italia impegnata in prima linea anche a supporto delle fasce più deboli [21] in progetti che si collocano all'interno di iniziative analoghe rivolta ad un pubblico più ampio [22].

In questo lavoro presenteremo un'esperienza biennale relativa ad un curriculum “formale” d'Informatica per gli studenti del terzo anno di un corso di laurea in “Formazione di Operatori Turistici”, curriculum che può essere adattato a equivalenti programmi nella scuola secondaria superiore [23-24] e trasversalmente lungo tutto il percorso educativo [25].

Il resto del lavoro è organizzato come segue: la sezione 2 presenta i criteri di progettazione del corso, dagli obiettivi agli approcci pedagogici; la sezione 3 presenta i dettagli del corso, dalle informazioni di contesto ai suoi contenuti, abilità, competenze, micro e macro progetti realizzati anche in forma collaborativa; la sezione 4 presenta una valutazione qualitativa del corso sulla base di un questionario di fine corso somministrato agli studenti e presenta delle riflessioni critiche sulla necessità di predisporre materiali didattici, indicazioni nazionali e curricula; la sezione 5 presenta le conclusioni e suggerisce alcuni sviluppi futuri.

2 Progettazione del corso

Il corso d'informatica è rivolto a studenti del terzo anno del Corso di Laurea in “Formazione di Operatori Turistici”. Il corso è il primo e l'unico corso di Informatica durante l'intero corso di Laurea. Nell'ottica dello sviluppo delle competenze del XXI secolo si è incoraggiato il lavoro di gruppo per la realizzazione di progetti utilizzando quanti più strumenti digitali possibili: da strumenti e metodi per la progettazione e creazione di applicazioni per dispositivi mobili come App Inventor, a strumenti e metodi per la progettazione e sviluppo di siti web, da strumenti e metodi per le ricerche bibliografiche a strumenti e metodi per la memorizzazione, analisi, e visualizzazione di dati aperti. L'approccio pedagogico ha seguito le moderne teorie costruttiviste e si è basato

sul learning by doing [26] e il project based learning [27]. La pedagogia Inquiry based learning [28-29], utilizzata anche in primari progetti europei [30], ha visto nell'uso di verifiche formative e sommative [31-32] la realizzazione ottimale dell'approccio. Nell'ottica di una metodologia Bring Your Own Device [33], si è incoraggiato l'uso di dispositivi (Laptop, Tablet, Smartphone, con alimentazione e connessione ad Internet autonoma) durante le lezioni.

I principi ispiratori della prima edizione del corso sono stati:

- 1) Sviluppo di competenze trasversali come imparare ad imparare, problem-solving, decision-making e computational thinking
- 2) Acquisire e favorire la progettazione e la realizzazione di applicazioni software oltre all'uso delle stesse.
- 3) Sviluppo di "app" per dispositivi mobili che risolvono problemi reali e che trovano applicazioni nella vita quotidiana
- 4) Sviluppo di competenze ed abilità per la ricerca, selezione, analisi e sintesi di informazioni necessarie alla risoluzione di problemi
- 5) Risoluzione autonoma di problemi piuttosto che seguire istruzioni precise
- 6) Una verifica e valutazione basata su competenze attraverso la progettazione, realizzazione e presentazione di diversi progetti
- 7) Un approccio basato sul laboratorio e sessioni pratiche di lavoro con lo scopo di guidare gli studenti nello sviluppo dei loro progetti, come lo sviluppo di "app" per promuovere e valorizzare siti turistici o la ricerca, selezione, analisi e visualizzazioni di dati aperti sui siti dichiarati dall'UNESCO patrimonio mondiale dell'umanità. Articoli sull'uso delle tecnologie dell'Informazione e della Comunicazione [34-35] sono stati suggeriti come approfondimento a questo riguardo.
- 8) Sviluppo di competenze collaborative attraverso lavori di gruppo.
- 9) Sviluppo di competenze comunicative anche attraverso l'uso costante di strumenti di comunicazione on-line e real-time.

La prima edizione del corso, nell'anno accademico 2016/17, è stata seguita da 45 studenti, la seconda, nell'anno accademico 2017/18, da 70 studenti, in questo secondo caso anche da studenti provenienti da altri corsi di laurea che hanno scelto il corso come materia opzionale. Le donne, con una percentuale dell'85%, hanno, in entrambi i corsi, rappresentato la netta maggioranza dei corsisti. I corsi si sono tenuti durante il primo semestre del terzo anno, tra ottobre e gennaio, con 24 incontri di due ore ciascuno per un totale di quarantotto ore in due incontri settimanali. Gli studenti arrivano al primo semestre del terzo anno con studi principalmente umanistici con due corsi di Archeologia, uno di museologia, due di economia, due corsi per tre lingue straniere (inglese, tedesco e spagnolo), preistoria, sette corsi di storia, due di diritto ed uno di sociologia. Lo studio dell'Informatica al primo semestre del terzo anno è affiancato da un corso di statistica, con il quale condivide l'uso di strumenti per la ricerca, analisi e visualizzazione di dati, altri due corsi di storia e un corso di geografia.

Le tabelle 1 e 2 riportano le conoscenze, competenze ed abilità principali sviluppate all'interno del corso e forniscono un esempio di progetti assegnati con cadenza media settimanale. In particolare, la tabella 1 contiene informazioni sulla ricerca di informazioni e sullo sviluppo di applicazioni per dispositivi mobili. Come strumento per questa

seconda attività è stato utilizzato l'ambiente di sviluppo, sia su cloud sia stand alone, App Inventor [36-38]. La tabella 2 si focalizza sulla ricerca, selezione, memorizzazione, analisi e visualizzazione di dati e sugli strumenti di presentazione.

Nella seconda edizione del corso si è dato più spazio alla progettazione e realizzazione di dispositivi mobili anche perché, grazie all'esperienza dell'anno precedente, esperienza condivisa tra pari dagli studenti stessi, è stato un po' più semplice superare la paura degli studenti e delle studentesse a cimentarsi con questi argomenti. A questo riguardo risulta molto utile presentare e condividere con gli studenti i progetti realizzati da loro stessi negli anni precedenti contribuendo a formare una "base di conoscenza" che si incrementa con il tempo.

Tabella 1. Progettazione del corso: ricerca di informazioni e sviluppo di applicazioni

N°	Conoscenze	Abilità/Competenze	Micro-Macro progetti
1	Gogle Sites Google Forms	Criteri di progettazione di un sito Criteri di progettazione di un questionario	Progettare un sito e un questionario
2	Motori di ricerca	Metodologie e tecniche per la ricerca, analisi, selezione e sintesi di informazioni	Ricerca bibliografica e suo uso in un articolo di review
3	Programmazione ad eventi: interfaccia	Criteri di progettazione dell'interfaccia interfaccia utente	Creare l'interfaccia di un'app con input e output di dati
4	Programmazione di un'app	Progettazione dei gestori degli eventi	Creare un'app con input e output di dati
5	Istruzioni di scelta	Progettazione e realizzazione di applicazioni che richiedono delle scelte	Creare un'app per gestire scelte dell'utente
6	Cicli	Uso di cicli con un numero finito di ripetizione e con condizioni	Algoritmi di base: conversioni tra sistemi di numerazione
7	Variabili	Variabili e loro uso.	Creazione di app con memorizzazione di stato. Storie interattive
8	Liste	Uso delle liste	Algoritmi di ricerca e ordinamento.
9	Procedure e funzioni	Progettazione di un software top-down	Creazione di un'app relativa al turismo
10	Database NoSQL	Progettazione e realizzazione di database NoSQL.	Uso di un database NoSQL in un'app relativa al turismo
11	Progettazione di un'applicazione	Progettazione di applicazioni	Creazione di un'app relativa al turismo

La figura 1 rappresenta due esempi dell'interfaccia grafica di due applicazioni realizzate da due gruppi di studenti.

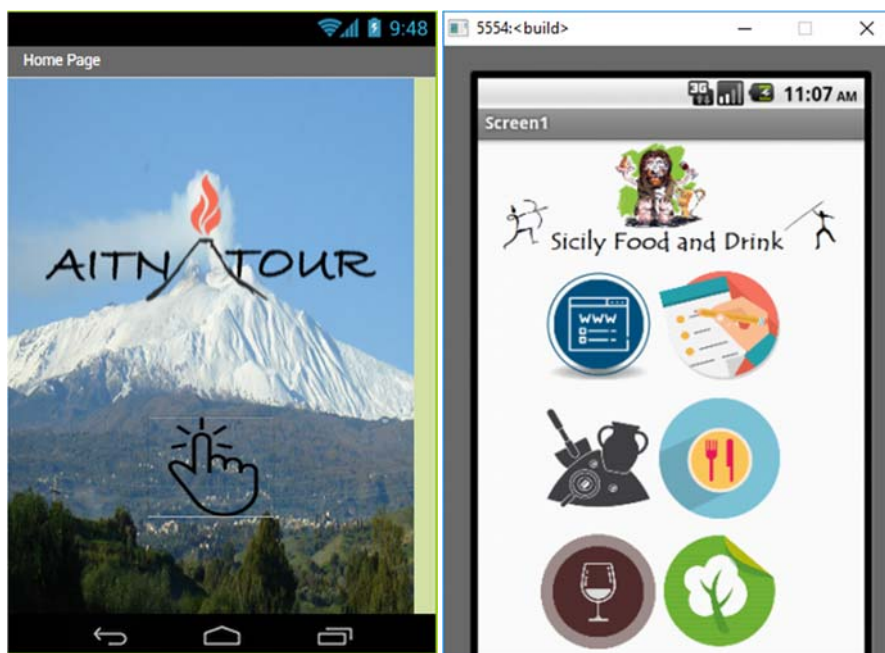


Fig. 1. Due esempi di interfacce grafiche di applicazioni realizzate dagli studenti del corso.

Le applicazioni sono state principalmente rivolte alla valorizzazione, in forma multimediale, di siti turistici presenti nel territorio. I siti sia naturalistici sia culturali, sono stati valorizzati oltre che da un'applicazione in App Inventor e da un video dimostrativo dell'applicazione stessa, anche attraverso un sito web, realizzato in genere con Google Sites, una presentazione, una ricerca bibliografica sui siti studiati e sulle problematiche tecniche riscontrate. I lavori sono stati completati da una ricerca ed analisi dei dati relativi al sito d'interesse, anche recuperandoli attraverso contatti con le agenzie sparse nel territorio, da un questionario somministrato ad utenti campione, e da un database relazionale finalizzato alla memorizzazione e gestione di dati relativi al sito turistico.

Nella seconda edizione del corso i progetti relativi ad App Inventor sono stati più corposi, anche se ci sono ancora spazi di miglioramento sull'uso di blocchi e sul coding. Nella prima edizione del corso le relazioni bibliografiche hanno avuto mediamente un taglio leggermente più orientato alla ricerca sull'uso delle tecnologie in campo umanistico con particolare riferimento all'archeologia e ai sistemi informativi territoriali; nella seconda sono state più finalizzate alla valorizzazione del sito turistico.

Nella seconda edizione del corso relativamente ad App Inventor è stato approfondito l'uso di Firebase. Altre applicazioni tipiche hanno spaziato dall'uso di mappe interattive, consultazione di siti di approfondimento, realizzazione di bussole e sistemi di guida lungo percorsi tipicamente naturalistici, prenotazione e contatti con mezzi telefonici e sociali, creazione di animazioni.

Tabella 2. Progettazione del corso: analisi dei dati e presentazione delle informazioni

N°	Conoscenze	Abilità/Competenze	Micro-Macro progetti
12	Database relazionali: modello concettuale	Modello Entità/Associazione	Progettazione di un database
13	Database relazionali: modello logico	Modello Logico dei dati	Realizzazione di un Database in Access
14	Query: proiezione, selezione e giunzione	Operazioni di base dell'algebra relazionale	Progettazione e realizzazione di query
15	Query: giunzioni su tre tabelle. Raggruppamenti.	Giunzioni tra tabelle relative a entità in associazione N-N	Progettazione e realizzazione di query
16	Query: Raggruppamenti e condizioni sui raggruppamenti	Individuazione ed uso di raggruppamenti e condizioni sui raggruppamenti	Progettazione e realizzazione di query
17	Presentazione dei dati. Progetto di un database	Progettazione e realizzazione di un database inerente al turismo	Realizzazione di un database in Access
18	Foglio di calcolo: riferimenti, intervalli, formule e funzioni	Uso di riferimenti, intervalli e funzioni in formule	Progettazione e realizzazione di un foglio di calcolo
19	Grafici	Progettazione e realizzazione di grafici	Visualizzazione di dati
20	Fusion tables e tabelle Pivot	Progettazione e realizzazione di tabelle di dati per rispondere a delle domande	Uso di fusion tables e delle tabelle Pivot
21	Open data. Progetto sull'uso degli open data	Ricerca, selezione, analisi e visualizzazione di open data	Ottenere dati a supporto di conclusioni e loro elaborazione
22	Strumenti di presentazione	Criteri di progettazione e realizzazione di presentazioni	Realizzazione di una presentazione
23	Strumenti di videoscrittura	Uso di strumenti di videoscrittura	Realizzazione di un manuale utente e un manuale tecnico
24	Ricerche bibliografiche	Metodi per effettuare ricerche bibliografiche	Realizzazione di un articolo di review

3 Valutazione del corso e considerazioni

Il corso è stato valutato attraverso un questionario somministrato agli studenti dopo la chiusura del corso, ma prima della prima sessione d'esame, al quale gli studenti hanno risposto su base volontaria ed anonima. In particolare, il 73,9% degli studenti ha ritenuto che lavorare per progetti richiede più lavoro, ma ben il 90,9 % degli studenti ritiene che lavorando per progetti si impara di più. Questi risultati supportano le decisioni che

hanno portato a realizzare sia il corso sia la sua valutazione enfatizzando l'aspetto progettuale. Il 69,9% degli studenti ritiene che sia meglio creare un'applicazione piuttosto che usarla, a supporto della scelta progettuale di potenziare aspetti legati alla progettazione e realizzazione di applicazioni e al coding.

La figura 2 mostra l'auto-valutazione degli studenti relativamente alla loro capacità di creare applicazioni all'inizio del corso a) e alla fine del corso b). Gli studenti hanno valutato le loro capacità usando una scala Likert a 7 livelli (1 scarso, 7 ottimo).

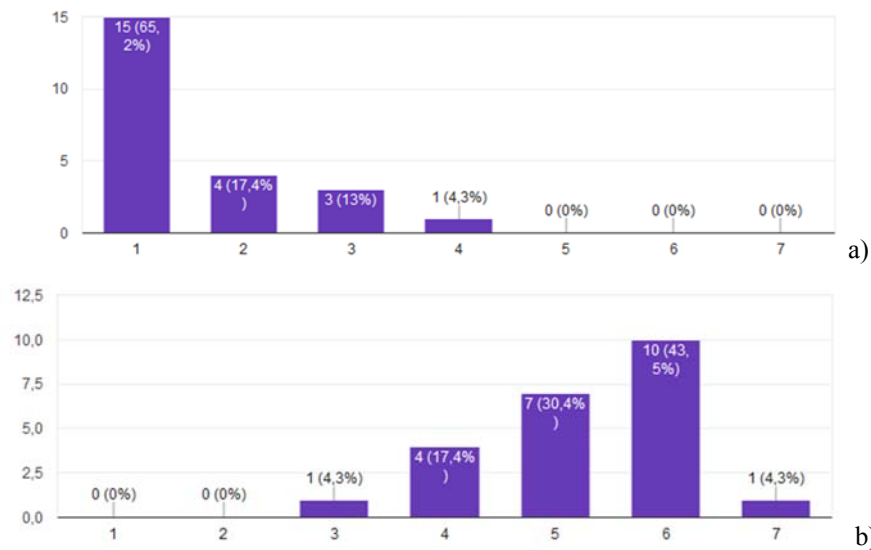


Fig. 2: autovalutazione degli studenti nella capacità di creare applicazioni all'inizio a) e alla fine b) del corso.

Dall'analisi delle risposte, oltre che dall'andamento di entrambi i corsi nelle due annualità, e dai risultati conseguiti nelle prove in itinere e dalla valutazione dei progetti di fine corso, si evince chiaramente che la bassa confidenza degli studenti è il maggior ostacolo da superare nel presentare un rigoroso approccio computazionale. La quasi totalità dei corsisti ha avuto un percorso di apprendimento che ha visto la totale assenza di contenuti legati all'informatica fino al terzo anno universitario. Questo è ancora presente nel percorso italiano: l'informatica ed il pensiero computazionale non è presente "formalmente" nella scuola secondaria di primo grado. Nella scuola secondaria superiore in alcuni percorsi di studi, come il liceo classico, è completamente assente, o ha una presenza marginale a supporto di altre discipline scientifiche come la matematica, lungo tutto il percorso di studi. Gli studenti, in alcuni casi, hanno una prima, ed unica, esposizione ad un curriculum ufficiale in Informatica, solo all'interno del loro percorso di studi universitari. Nel contesto internazionale sopra descritto, le considerazioni precedenti richiedono una profonda riflessione [39]. Le stesse risposte incoraggianti in termini di percezione degli studenti, percezione che ha trovato riscontro negli elaborati di fine corso, presentati di fronte ai colleghi, e nella valutazione orale, si sono ottenute anche per la ricerca e selezione d'informazioni e sull'uso di software per la creazione

di contenuti digitali (documenti, siti, presentazioni, etc...) in forma collaborativa. La figura 3 sintetizza l'autovalutazione degli studenti.

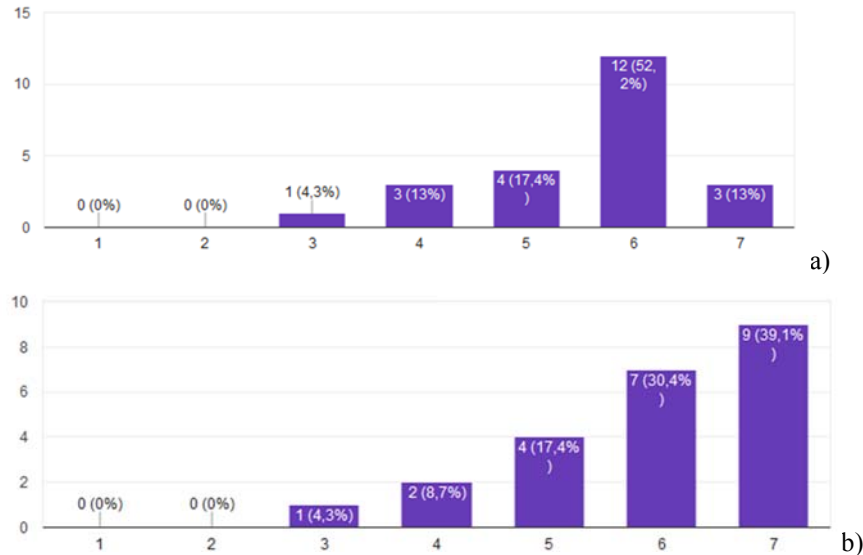


Fig. 3: autovalutazione degli studenti delle loro capacità di cercare e selezionare software e, in generale informazioni a), e creare contenuti in forma collaborativa b).

Infine, come mostrato dalla figura 4, è fortemente sentita dagli studenti la necessità di approfondire i contenuti in corsi successivi.

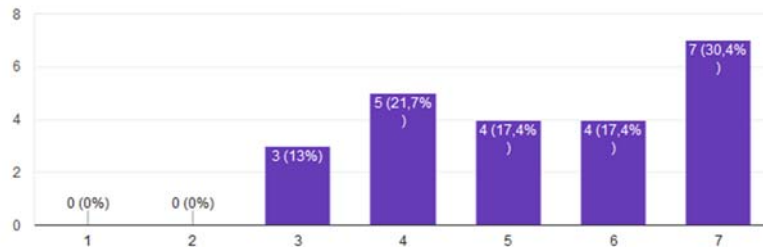


Fig. 4: Risposte degli studenti alla domanda: "Quanto ritieni utile un corso di approfondimento".

La percezione degli studenti trova riscontro dal confronto, effettuato dal docente, delle verifiche di metà corso e degli elaborati finali. In particolare, gli elaborati progettuali relativi al coding e all'applicazione in App Inventor, hanno visto, nelle due edizioni del corso, miglioramenti nella complessità e ricchezza delle soluzioni proposte, anche se, come già detto in precedenza, l'aspetto del coding necessita di ulteriori approfondimenti e sviluppi. Per la realizzazione di siti web risulta importante l'approfondimento di tecniche di progettazione dell'interfaccia e dell'interazione con l'utente. Si auspica inoltre l'ampliamento del ventaglio di progetti internazionali rilevanti nel settore d'interesse, come [40], da presentare agli studenti per approfondimenti.

4 Conclusioni e sviluppi futuri

In questo lavoro, a partire da un'analisi della letteratura e dello stato dell'arte sui nuovi framework e curricula relativi al pensiero computazionale, sono stati presentati criteri di progettazione, contenuti, abilità, competenze e progetti relativi ad un corso di Informatica per studenti di discipline umanistiche tenutosi negli anni accademici 2016/17 e 2017/18, corso valutato attraverso un'analisi qualitativa. A partire dalle considerazioni ed esperienze svolte si ritiene opportuno esporre tutti gli studenti di tutte le discipline, non solo quelle scientifiche e tecniche, a corsi d'informatica consentendo loro di ampliare ulteriormente il bagaglio culturale e si auspicano materiali didattici, curricula e linee guida a supporto di queste tematiche. A fronte di curricula e linee guida sviluppati da gruppi di lavoro, sia a livello ministeriale sia a livello internazionale, dovrebbero essere sviluppati rigorosi materiali didattici a supporto di questi curricula in modo da consentire sia ad organizzazioni operanti nel territorio nazionale sia a singoli docenti di realizzare un percorso didattico trasversale.

Riferimenti bibliografici

1. Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing Computational Thinking in Compulsory Education
2. DigComp. Being digitally competent – a task for the 21st century citizen. <https://ec.europa.eu/jrc/en/digcomp>
3. DigCompEdu. Digital Competence Framework for Educators. <https://ec.europa.eu/jrc/en/digcompedu>
4. DigCompOrg. European Framework for Digitally Competent Educational Organisations. <https://ec.europa.eu/jrc/en/digcomporg>
5. DigCompConsumers. The Digital Competence Framework for Consumers <https://ec.europa.eu/jrc/en/digcompconsumers>
6. OpenEdu. Policy Recommendations for Opening Up Education. <https://ec.europa.eu/jrc/en/open-education>
7. EntreComp. Entrepreneurship Competence. <https://ec.europa.eu/jrc/en/entrecomp>
8. Framework for 21st century learning – P21. <http://www.p21.org/our-work/p21-framework>
9. OECD. The Future of Education and Skills. Education 2030. <http://www.oecd.org/education/2030/oecd-education-2030-position-paper.pdf>
10. Department of Education. Statutory guidance. National curriculum in England: computing programmes of study. September 2013.
11. Advanced Placement. AP Computer Science Principles Including the Curriculum Framework. Fall 2017
12. <https://code.org/>
13. <http://codeweek.eu/>
14. <https://www.programmailfuturo.it/>
15. <http://codeweek.it/>
16. Ministero dell'Istruzione dell'Università e della Ricerca (2017). Indicazioni Nazionali e Nuovi Scenari. <http://www.miur.gov.it/-/indicazioni-nazionali-di-infanzia-e-primo-ciclo-piu-attenzione-alle-competenze-di-cittadinanza>

17. Balanskat, A., & Engelhardt, K. (2014). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. European Schoolnet.
18. ACM Europe & Informatics Europe. (February 2018). Informatics for All. The strategy.
19. A. Torrebruno, A. Torrebruno, S. Casola (2017). Polireading: alzare il livello della discussione pedagogica. Didamatica 2017.
20. A.H, Licht, E. Tasiopoulou, P. Wastiau, P. (2017). Open Book of Educational Innovation. European Schoolnet, Brussels.
21. DIS-CODE project. <http://www.allyouneediscode.eu/it/dis-code>
22. All you need is code. <http://www.allyouneediscode.eu/home>
23. Giordano, D., & Maiorana, F. (2014, April). Use of cutting edge educational tools for an initial programming course. In Global Engineering Education Conference (EDUCON), 2014 IEEE (pp. 556-563). IEEE.
24. Giordano, D., & Maiorana, F. (2015, March). Teaching algorithms: Visual language vs flowchart vs textual language. In Global Engineering Education Conference (EDUCON), 2015 IEEE (pp. 499-504). IEEE. ECDL computing
25. ECDL Computing. <http://www.aicanet.it/ecdl-computing>
26. Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. Instructional-design theories and models: A new paradigm of instructional theory, 2, 161-181.
27. Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist*, 26(3-4), 369-398.
28. E Hazelkorn, R Charly. (2015) Science education for responsible citizenship. Report to the European Commission of the Expert Group on Science Education
29. European Commission (EC), & High Level Group on Science Education. (2007). Science Education NOW: A Renewed Pedagogy for the Future of Europe (EUR 22845). Brussels: DG Research.
30. Scientix project. <http://www.scientix.eu/>
31. Giordano, D., Maiorana, F., Csizmadia, A. P., Marsden, S., Riedesel, C., Mishra, S., & Viničienė, L. (2015, July). New horizons in the assessment of computer science at school and beyond: Leveraging on the viva platform. In Proceedings of the 2015 ITiCSE on Working Group Reports (pp. 117-147). ACM.
32. Maiorana, F., Giordano, D., & Morelli, R. (2015, October). Quizly: A live coding assessment platform for App Inventor. In Blocks and Beyond Workshop (Blocks and Beyond), 2015 IEEE (pp. 25-30). IEEE. L.
33. Bennett, L., & Tucker, H. (2012). Bring your own device. *ITNow*, 54(1), 24-25.
34. Schieder, T. K., Adukaite, A., & Cantoni, L. (2013). mobile apps devoted to UNESCO world heritage sites: A map. In *Information and Communication Technologies in Tourism 2014* (pp. 17-29). Springer, Cham.
35. Werthner, H., Alzua-Sorzabal, A., Cantoni, L., Dickinger, A., Gretzel, U., Jannach, D., ... & Stangl, B. (2015). Future research issues in IT and tourism. *Information Technology & Tourism*, 15(1), 1-15.
36. Wolber, D. (2011, March). App inventor and real-world motivation. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 601-606). ACM.
37. Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2011). *App Inventor*. " O'Reilly Media, Inc."
38. App Inventor. <http://appinventor.mit.edu/explore/>
39. https://labuonascuola.gov.it/area/m/7829/?_gmln_m=m14275
40. Europeana Foundation (2015). Transforming the world with culture.