Form@re
Open Journal per la formazione in rete

esperienze/riflessioni

Start App: una esperienza di coding tra scuola primaria e scuola secondaria

Start App: a coding experience between primary and secondary school

Filippo Bruni[a], Luigi D'Onofrio[b], Michela Nisdeo[c,1]

[a] *Università degli Studi del Molise, filippo.bruni@unimol.it*
[b] *MIUR, ITI Marconi Campobasso, donofriogino@gmail.com*
[c] *MIUR, IC Colozza Campobasso, michelanis@gmail.com*

Abstract

Il contributo presenta una esperienza di coding nella scuola primaria dell'Istituto Comprensivo statale "Colozza" di Campobasso. All'interno della cornice teorica offerta dal pensiero computazionale, utilizzando App Inventor, è stata realizzata una calcolatrice per smartphone in ambiente Android. A guidare gli allievi della primaria sono stati gli studenti dell'Istituto Tecnico Industriale "Marconi" di Campobasso realizzando una interessante forma di collaborazione tra istituti scolastici di ordine diverso.

Parole chiave: coding; pensiero computazionale; competenze digitali; App Inventor; calcolatrice.

Abstract

The paper presents a coding experience in primary school ("Colozza" in Campobasso). Within the theoretical framework offered by computational thinking, using App Inventor, it was created a calculator for smartphone in the Android environment. High school students (from a technical secondary school) guided the pupils in primary school, making an interesting form of cooperation between primary and secondary schools.

Keywords: coding; computational thinking; digital literacy; App Inventor; calculator.

[1] Il testo è frutto della progettazione condivisa e negoziata tra i tre autori, la stesura del paragrafo 1 è opera di Filippo Bruni, la stesura del paragrafo 2 è opera di Michela Nisdeo, la stesura dei paragrafi 3 e 4 è opera di Luigi D'Onofrio.

1. Coding, pensiero computazionale, competenze digitali e didattica della matematica

L'esperienza presentata in queste pagine – chiamata "Start App" dai suoi promotori e nata dalla collaborazione tra un Istituto Tecnico, il "Marconi" di Campobasso ed una scuola primaria dell'Istituto Comprensivo "Colozza", sempre di Campobasso – si inserisce, in prima battuta, all'interno del filone di studi legato al coding, cioè, come noto, all'utilizzo di linguaggi informatici per la realizzazione di software e applicazioni. Le sperimentazioni legate al coding stanno ricevendo una grande attenzione ed il successo di un sito come Code (https://code.org/) è solo un elemento che lo testimonia. Il rischio è però quello di intendere tali attività solo come una anticipazione di una specifica disciplina, l'informatica, nella scuola primaria, perdendo, o almeno allentando, il legame da un lato con il più vasto campo del pensiero computazionale e delle competenze digitali, dall'altro con la matematica e la didattica della matematica.

Il *computational thinking*, come ricorda Jeannette M. Wing che nel 2006 ha coniato questa espressione, è caratterizzato da una serie specifica di elementi che può essere utile ricordare:

- "concettualizzazione non dalla programmazione": il pensiero computazionale va oltre l'abilità di programmare un computer;
- "abilità fondamentale, non legata alla routine": si tratta di andare oltre l'applicazione meccanica per individuare una competenza centrale nella società contemporanea;
- "un modo di pensare umano non del computer": il pensiero computazionale vuole risolvere problemi non ridurre il pensiero umano alle modalità di funzionamento del computer;
- "raccordare e unire il pensiero matematico con quello ingegneristico" nel senso di far interagire il ragionamento logico con la costruzione di sistemi che agiscono nel mondo reale;
- "idee non artefatti" in quanto non si tratta tanto di realizzare software ma "di risolvere problemi, gestire la nostra vita quotidiana, comunicare ed interagire con le altre persone";
- "per tutti, dovunque": più che un esplicito approccio vorrebbe essere trasversale e pienamente integrato nei contesti quotidiani (Wing, 2006, p. 35) e, si potrebbe aggiungere, nei percorsi formali di formazione.

L'approccio richiesto dal pensiero computazionale, inteso in termini generali come la capacità di risolvere problemi facendo appello al pensiero logico matematico e alle risorse digitali, è quindi un primo antidoto a visioni riduzioniste del coding.

In una direzione simile si muove l'evoluzione delle competenze digitali che, da un approccio più legato all'area disciplinare propria dell'informatica, si sono estese a ambiti più vasti (Calvani, Fini & Ranieri, 2010) collegandosi ad una cultura partecipativa che sottolinea il legame delle competenze digitali tanto con le competenze in qualche modo tradizionali (come quelle legate alla produzione e comprensione di testi) quanto alla dimensione sociale (Jenkins, Purushotma, Weigel, Clinton & Robinson, 2010).

Una indicazione di tal genere acquista importanza in relazione alla didattica della matematica nella scuola primaria, dove il coding non sostituisce ma si aggiunge successivamente alla conseguita padronanza dei processi mentali che permettono la soluzione di problemi matematici (Sanford & Naidu, 2016). Del resto, andando all'origine del coding, si ritrovano le proposte e le esperienze di Papert (1980) che, oltre trent'anni fa,

ricordava che "è possibile progettare degli elaboratori in modo che comunicare con loro può essere un processo naturale" e "quando questa comunicazione avviene, i bambini apprendono la matematica come una lingua viva" (p. 12). L'obiettivo indicato da Papert, di legare l'apprendimento della matematica a contesti reali, rimane ancora valido.

Alla luce, e coerentemente, ad un tale quadro teorico, "Start App" ha voluto in primo luogo proporre un approccio inusuale: le attività di coding vengono proposte in una quinta primaria non dai docenti ma dagli studenti di una quinta dell'istituto tecnico. In secondo luogo, essendo ormai acquisite alcune competenze legate all'area matematica, si è puntato sulla realizzazione di applicazioni per smartphone in ambiente Android, legando il percorso ad uno strumento che fa ormai parte della vita quotidiana. Come terzo ed ultimo aspetto si segnala la ricerca di altri ambiti oltre quello matematico, come emerge dalla successiva presentazione dell'applicazione realizzata. In tal senso va anche tenuto presente, per cogliere in modo opportuno il contesto, che nel percorso di formazione quinquennale della classe, sin dalla seconda, è stato fatto un sistematico uso di Moodle come ambiente trasversale alle diverse attività didattiche. A ciò vanno aggiunte attività di coding realizzate con Scratch (https://scratch.mit.edu/) legate alla storia, al digital storytelling, alla creazione di quiz. La presenza di un alunno disabile, pienamente coinvolto nelle attività, è un altro aspetto da tener presente per cogliere la dimensione inclusiva della sperimentazione.

L'esperienza non è ancora conclusa. Qui viene presentato solo una parte del percorso, legata ad una indagine sulle calcolatrici per smartphone destinate alla scuola primaria e alla realizzazione, da parte degli studenti della primaria, di una calcolatrice.

2. App per calcolare

Nella scuola primaria le app sono sempre più utilizzate, sia in contesti scolastici che nello studio individuale, e possono modificare, positivamente, l'insegnamento delle discipline quali ad esempio la matematica. Ognuna di esse facilita processi che sono implicati nell'apprendimento di tale disciplina che riveste un ruolo centrale nei curricoli scolastici.

Nel proliferare di app di matematica, occorre fare una distinzione tra quelle in cui l'elemento ludico è prevalente o esclusivo e quelle in cui la dimensione ludica interagisce in modo adeguato con l'aspetto educativo. Assumendo come principale criterio selettivo la dimensione dell'apprendimento, si propone un elenco (Figura 1) di sette app calcolatrici o appositamente pensate per la fascia d'età della scuola primaria o comunque utilizzabili in tale ordine di scuola. Di ciascuna app si indicano il nome, l'autore, l'indirizzo web da cui reperire maggiori informazioni, la/le lingue di utilizzo e il prezzo. Si aggiunge inoltre una breve descrizione.

Nome	Autore	Indirizzo web	Lingua/e di utilizzo	Prezzo
Calcolatrice	The App Tower Inc.	https://itunes.apple.com/us/app/calculator/id552974550?mt=8	Inglese	gratis
Per ambiente Apple, la calcolatrice è caratterizzata da un bel design, feedback animati e una disposizione tradizionale dei tasti per calcoli facili e veloci. Si segnalano inoltre: animazioni fluide che scorrono alla pressione dei tasti, notazione scientifica, design colorato, supporta fino ad otto cifre e cinque operazioni.				

Moles of Calculator	Yasukazu Umekita	http://umekita-app-en.blogspot.jp/	Inglese, giapponese	gratis
Per ambiente Apple e Android. Di semplice e facile uso con una interfaccia graficamente curata per i bambini.				
Calculator Flower Free	Yasukazu Umekita	http://umekita-app-en.blogspot.it/search?updated-max=2015-12-12T08:06:00%2B09:00&max-results=15	Inglese, giapponese	gratis
Per ambiente Apple e Android. Oltre a essere di facile uso, offre cinque diversi tipi di interfaccia.				
Classicalc	Lukasz Kowalski	https://twitter.com/ClassicalcMusic	Inglese	0,99 €
Per ambiente Apple, utilizzabile anche sull'apposito orologio. La peculiarità è data dal fatto che a ciascun numero è associata una nota abbinando la progressione numerica alla scala musicale.				
Kids Calculator	Nolan Piper	http://www.math3g.com/Site/Welcome.html	Inglese	gratis
Per ambiente Apple, calcolatrice che aggiunge alle funzioni di calcolo un gioco che propone, a molteplici livelli di difficoltà, problemi matematici.				
jCalc Multi Calculator	Jon Ericksen	http://www.jonericksen.com/	Oltre all'inglese e all'italiano sono disponibili una ampia scelta di lingue come norvegese, cinese, francese, tedesco	0,99 €
Calcolatrice per ambiente Apple, con la possibilità di cambiare sfondi.				
Calcolatrice Tasti Grandi	Adalgisa Raniolo	https://itunes.apple.com/it/app/calcolatrice-tasti-grandi/id942770129?mt=8	Italiano, inglese, spagnolo	0,99 €
Per ambiente Apple, con cinque diverse interfacce grafiche.				

Figura 1. Applicazioni per calcolare.

3. App Inventor

È un progetto inizialmente nato in Google e poi passato e perfezionato al Massachusetts Institute of Technology (MIT). App Inventor è un ambiente di sviluppo per applicazioni

Android estremamente semplice da utilizzare. Questa caratteristica lo rende particolarmente adatto nell'insegnamento precoce dell'informatica. La semplicità d'uso nasce fondamentalmente da una interfaccia grafica molto semplice, essenziale, e estremamente amichevole, nonché da un linguaggio di programmazione grafico, strutturato a blocchi colorati, che permette di creare la logica di funzionamento delle app in maniera intuitiva ed in tempi sorprendentemente rapidi. Il linguaggio di programmazione ricorda molto da vicino quello di Scratch del MIT che non di rado viene insegnato e studiato anche nella scuola primaria e tale circostanza rende il processo di apprendimento ancora più semplice.

L'ambiente mette a disposizione degli utili strumenti per il debugging e per il testing. È possibile testare le applicazioni mediante un emulatore che va installato sul pc di sviluppo. Tale emulatore simula le funzioni essenziali di uno smartphone ed è in grado di eseguire l'applicazione che si sta sviluppando riproponendo molte delle funzioni di uno smartphone reale. È inoltre possibile testare direttamente sul proprio dispositivo mobile l'applicazione che si sta sviluppando effettuando modifiche live immediatamente verificabili sul telefono reale. Per quest'ultima funzionalità è necessario che il pc di sviluppo e lo smartphone siano collegati alla stessa rete wireless e che sul telefono sia in esecuzione una app specifica scaricabile liberamente dal Google Play Store.

Dopo le operazioni di test e di perfezionamento dell'app che si sta creando è possibile scaricare l'app in questione sul disco del pc di sviluppo oppure direttamente sul dispositivo di test o ancora inserire l'app nel Play Store. App Inventor non è un'applicazione desktop, non esiste un programma di installazione: è invece una applicazione web che parte direttamente dal browser web che stiamo utilizzando.

Andando all'indirizzo http://appinventor.mit.edu/explore/, per poter lavorare è sufficiente premere il pulsante in alto a destra con la scritta "Create apps!". Dopo aver premuto il pulsante ci viene proposto di collegarci. È sufficiente avere un account Gmail per connetterci. La prima volta che effettuiamo tale operazioni il programma richiede delle autorizzazioni che dobbiamo concedere per poter procedere.

Ci troveremo quindi, nell'ambiente di sviluppo, nella interfaccia di progettazione grafica (Figura 2).

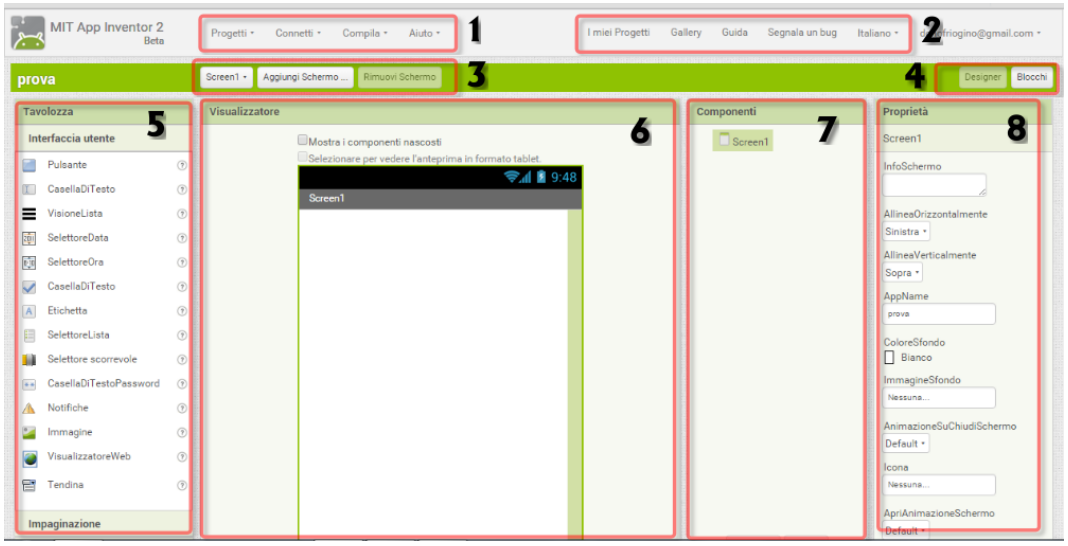

Figura 2. App Inventor: interfaccia grafica.

L'interfaccia dell'ambiente di progettazione grafica è organizzato nelle seguenti parti:

1. menu principale;
2. menu secondario;
3. menu per la gestione schermi;
4. pulsanti per il passaggio dall'ambiente grafico (*Designer*) all'editor del codice (*Blocchi*);
5. la tavolozza degli oggetti grafici;
6. il visualizzatore;
7. la finestra componenti;
8. la finestra proprietà.

Il menu principale contiene una serie di sottomenu tra cui si segnalano: il sottomenu *Progetti*, che contiene tutte le funzioni per la creazione, il salvataggio, l'esportazione e l'importazione dei lavori realizzati. Il sottomenu *Connetti*, che permette di testare le app mediante un dispositivo connesso alla stessa rete wireless del pc di sviluppo (AI Companion). Permette sia di lanciare l'emulatore sul pc di sviluppo con la nostra app in esecuzione simulata sia di inviare la app ad un dispositivo reale connesso ad una porta usb del pc di sviluppo. Il sottomenu *Compila* che ci permette di creare su disco le app.

Il menu secondario porta alla lista delle applicazioni già realizzate. Da questa schermata è possibile creare un nuovo progetto, eliminare un progetto selezionato o pubblicare nella *Gallery* (un archivio di app pubbliche). La voce *Gallery* indirizza all'archivio delle app pubbliche. Le ultime due voci di menu permettono all'utente di segnalare un malfunzionamento e di impostare la lingua dell'interfaccia.

Il menu per la gestione degli schermi ha un primo pulsante per selezionare la schermata di lavoro, mentre gli altri due pulsanti rendono possibile aggiungere/eliminare schermate per la propria app.

I pulsanti per il passaggio dall'ambiente grafico (*Designer*) all'editor del codice (*Blocchi*) permettono di passare dalla modalità di progettazione grafica di tipo visuale della schermata (*Designer*) alla modalità di scrittura del software (*Blocchi*). Nella seconda modalità si entra in un editor del codice che permette di scrivere le istruzioni per l'applicazione.

La tavolozza degli oggetti grafici rende disponibili tutto ciò che occorre per definire l'aspetto grafico dell'app, basta trascinare l'oggetto dalla tavolozza alla schermata. Gli oggetti sono funzionalmente classificati e quindi facilmente individuabili: oggetti per l'interfaccia utente, per l'impaginazione, multimediali, disegno e animazione, sensori, social, etc.

La finestra *Componenti* mostra a colpo d'occhio la struttura e l'organizzazione grafica della schermata selezionata (Figura 3).

Dalla finestra è possibile vedere chiaramente che la schermata della nostra applicazione è composta da quattro contenitori: *Orientamento Orizzontale 1*, *Allineamento Verticale 1*, *Allineamento Verticale 2*, *Allineamento Verticale 3*.

I contenitori disporranno gli oggetti che conterranno e delimiteranno logicamente le seguenti aree: *area operandi*, *area operazioni*, *area risultato*, *area configurazione*. L'*Orientamento Orizzontale 1* delimita l'area operandi e contiene in disposizione orizzontale una etichetta e una casella di testo per il primo e il secondo operando. L'*Allineamento Verticale 1* delimita l'area operazioni e contiene in disposizione verticale una etichetta, un menu a tendina per scegliere l'operazione e un pulsante per eseguire il

calcolo. L'*Allineamento Verticale 2* delimita l'area risultato e contiene in disposizione verticale una etichetta con la scritta risultato e una etichetta il cui contenuto verrà impostato dal codice dell'app e restituisce il risultato del calcolo. L'*Allineamento Verticale 3* delimita l'area configurazione che contiene in disposizione verticale un check box per attivare la sintesi vocale, un'etichetta e un menu a tendina per scegliere la lingua della sintesi.

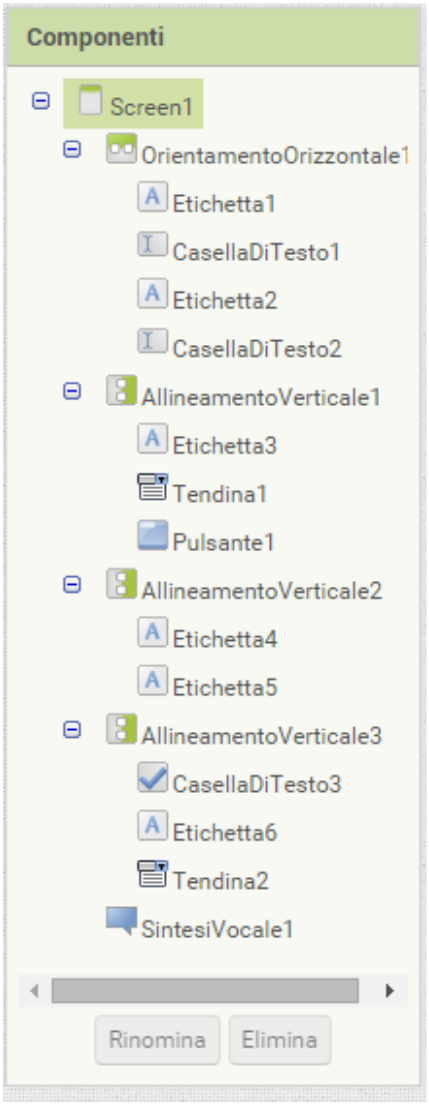

Figura 3. Finestra *Componenti*.

4. Costruire una app calcolatrice

Lo screenshot dell'app che realizzeremo è visibile nella Figura 4.

L'interfaccia è molto semplice: abbiamo due caselle testuali per inserire i valori del primo e secondo operando, un elenco a discesa per scegliere le operazioni, un pulsante per eseguire l'operazione selezionata, un area (in giallo) nella quale restituire il risultato e un'area di configurazione con un check box per attivare la sintesi vocale e un elenco a discesa per selezionare la lingua della sintesi.

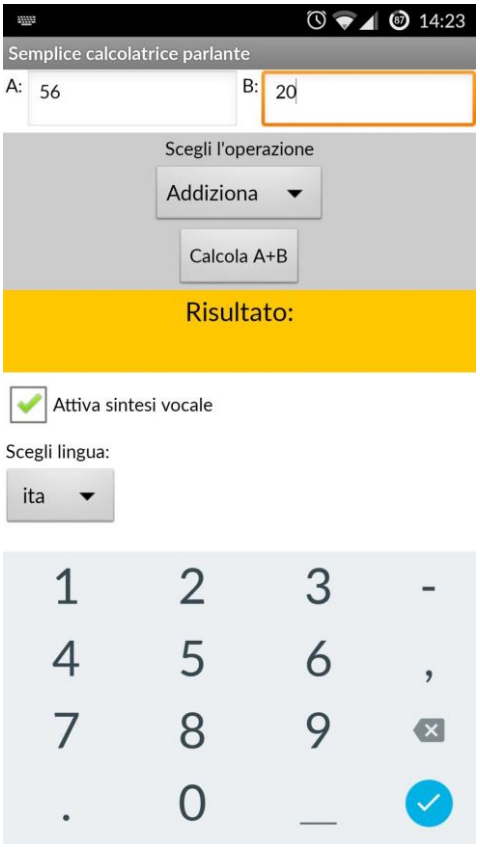

Figura 4. Screenshot dell'applicazione.

4.1. Prima fase (creazione del progetto e costruzione dell'interfaccia dell'app)

Segnaliamo i principali passaggi per la realizzazione dell'app. Creiamo un nuovo progetto dandogli il nome "Calcolatrice", andiamo al menu principale e scegliamo la voce *Progetti* e quindi *Avvio nuovo progetto*. Nella finestra che apparirà, inseriamo il nome "Calcolatrice" e il nostro nuovo progetto verrà aperto nell'editor visuale. Evidenziati vedremo il nome del progetto e la schermata *Screen 1* che rappresenta al momento l'unico componente dell'applicazione. Possiamo cambiare il nome della schermata dalla finestra proprietà cambiando in "Calcolatrice parlante" il campo "Titolo".

Nella sezione *Impaginazione* della *Tavolozza oggetti grafici* va preso un contenitore *Orientamento Orizzontale* e trascinato sulla schermata nell'area di lavoro. Vanno trascinati sulla schermata immediatamente sotto tre contenitori *Allineamento Verticale*. Apparirà un quadrato allineato a sinistra sulla schermata e un oggetto nella finestra componenti nominato automaticamente come *Orientamento Orizzontale 1*. Va selezionato il primo contenitore nella finestra *Componenti* e nella finestra *Proprietà*, cambiando la proprietà "Larghezza" in "Riempi contenitore". L'oggetto *Orientamento Orizzontale* occuperà tutto lo spazio orizzontale della schermata. Ripetere le operazioni anche per i tre contenitori verticali. Dovremmo avere nell'area di lavoro la base sulla quale costruire l'interfaccia della nostra app.

Cominciamo a costruire l'area operandi. Dobbiamo permettere all'utente di inserire due numeri (operandi) etichettati come A e B. Partendo da sinistra in *Orientamento Orizzontale* gli oggetti saranno disposti uno affianco all'altro. Ci sarà un'etichetta con il testo A (una

casella di testo) e un'etichetta con il testo B (un'altra casella di testo). Andiamo alla *Tavolozza oggetti grafici* e apriamo la sezione *Interfaccia utente*. Prendiamo un oggetto *Etichetta* e trasciniamolo sulla schermata avendo l'accortezza di rilasciarlo nel primo dei quattro contenitori. Selezioniamo l'etichetta e scriviamo nella proprietà "Testo A". Trasciniamo poi affianco all'etichetta un oggetto *Casella di testo*. Ripetiamo le operazioni anche per il secondo operando e dovremmo avere l'area operandi pronta.

Nell'*Area operazioni*, essendo un contenitore verticale, gli oggetti verranno impilati dall'alto verso il basso e sono: un'etichetta che ci segnala di scegliere un'operazione, un elenco a discesa che rende possibile tale opzione, un pulsante riportante la scritta "Calcola l'operazione" per eseguire il calcolo. Prima di inserire gli oggetti prepariamo il contenitore verticale in modo da disporre gli oggetti centrati orizzontalmente. Selezioniamo il contenitore *Allineamento Verticale 1* e impostiamo la proprietà "Allinea orizzontalmente" a "Centro". Dalla *Tavolozza* prendiamo nell'ordine una *Etichetta*, una *Tendina* e un *Pulsante* e li rilasciamo nel contenitore *Allineamento Verticale 1*. A questo punto l'*Area operazioni* risulta completa.

Nell'*Area risultato* trasciniamo due *Etichette* dalla *Tavolozza degli oggetti* al secondo contenitore verticale. Nella prima va impostata la proprietà "Testo" a "Risultato", nella seconda inseriremo una stringa vuota, il suo valore verrà impostato dal programma con il risultato del calcolo. Anche per questo contenitore verticale, come per il precedente, va impostata la proprietà "Allinea orizzontalmente" al valore "Centro". Anche l'*Area risultato* a questo punto dovrebbe risultare completa.

Per quanto riguarda l'*Area configurazione*, dalla *Tavolozza*, nella sezione *Interfaccia utente*, bisogna trascinare un *Check box* (tradotto infelicemente in italiano come casella di testo che, di fatto può essere confusa con la vera casella di testo usata per inserire i due operandi) sul terzo contenitore verticale. Va modificata la proprietà "Testo" in "Attiva sintesi vocale", e va aggiunta una *Etichetta* e una *Tendina*. La proprietà "Testo" dell'*Etichetta* va cambiata in "Scegli lingua". E la proprietà della tendina, "Elementi da Stringa", va impostata alle due opzioni possibili "ita, eng". Va aggiunto un ultimo oggetto, questa volta non visuale, scelto dal gruppo *Multimediale* della *Tavolozza*: una *Sintesi vocale*. L'interfaccia grafica dell'applicazione a questo punto dovrebbe essere pronta.

4.2. Seconda fase (scrittura del codice)

Per la scrittura del codice, cioè per l'elaborazione delle istruzioni che permetteranno all'app di funzionare esattamente nel modo voluto, va usato il pulsante in alto a destra con la scritta *Blocchi*. Diventa così disponibile l'editor del codice che ci permetterà di scrivere le istruzioni come blocchi funzionali colorati.

Come prima cosa all'avvio dell'applicazione impostiamo la *Tendina 1* al valore "Addiziona" e la *Tendina 2* al valore "ita".

Selezionando *Screen 1* (la nostra schermata) apparirà a destra una finestra con le istruzioni associate. Scegliamo il blocco *Inizializza* e trasciniamolo nel visualizzatore. Scegliamo *Tendina 1* e trasciniamo all'interno del blocco precedente l'istruzione "imposta selezione". Selezioniamo quindi il *Gruppo istruzioni testo* e trasciniamo la riga di testo collegandola alla precedente. Nella riga di testo scriveremo il primo valore di quelli possibili nella *Tendina 1*: "Addiziona". Inizializziamo anche la *Tendina 1* ripetendo le istruzioni precedenti.

Scegliamo dai blocchi *Pulsante 1* e dalla finestra che appare il *blocco click* che conterrà le istruzioni da eseguire al click del pulsante. Trasciniamo il blocco sul *Visualizzatore* sotto il blocco di istruzioni che abbiamo appena creato. Il visualizzatore apparirà più o meno come nella Figura 5.

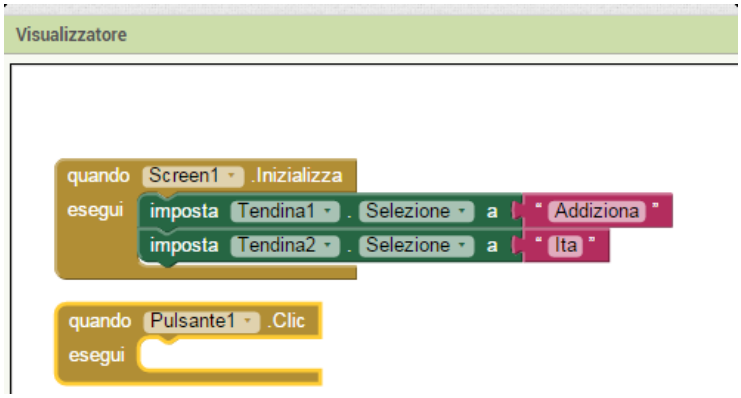

Figura 5. Visualizzatore.

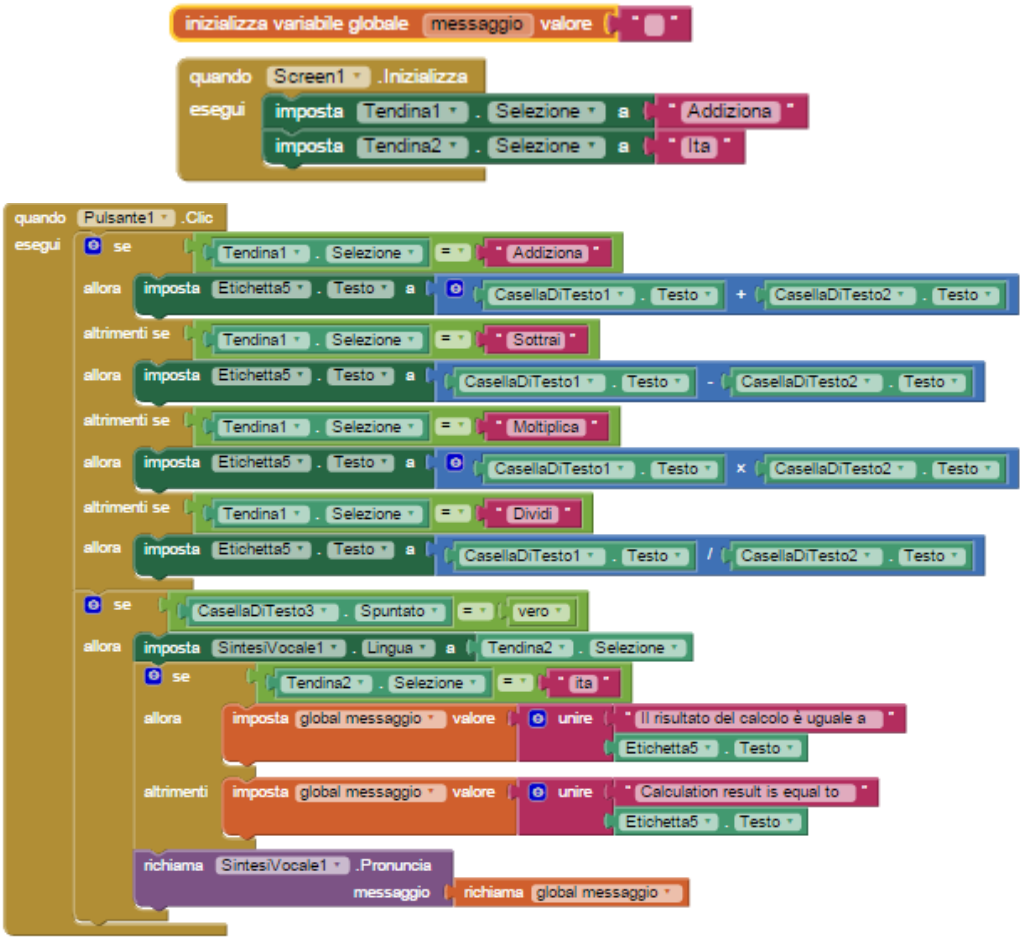

Figura 6. Il codice completo.

In Figura 6 riportiamo il codice completo commentato in tutte le sue istruzioni. Le istruzioni che compongono il codice vanno prese dalla finestra *Blocchi* e sono classificate sia funzionalmente e sia in base agli oggetti inseriti nell'interfaccia grafica. Nella classificazioni gioca un ruolo importate il colore dei blocchi: questa caratteristica assegna un vantaggio strategico alle nuove generazioni nella velocità con la quale sono capaci di individuare il blocco funzionale che cercano.

Queste sono le istruzioni da inserire e, come si diceva, può essere d'aiuto nella scrittura del codice notare il colore dei blocchi (Figura 7). Tale informazione ci permette di scegliere la classe di blocchi che contiene la nostra istruzione.

Dal codice vediamo chiaramente la logica dell'applicazione: quando viene premuto il pulsante *Calcola* viene determinata l'operazione selezionata nel menu a tendina e i contenuti delle due caselle di testo degli operandi vengono processati con l'operazione scelta e il valore ottenuto è assegnato all'etichetta risultato. Successivamente se il *check box* della sintesi è spuntato, si imposta la lingua scelta nel menu a *Tendina 2* e quindi, se la lingua selezionata è l'italiano, si costruisce un messaggio di risposta in italiano altrimenti lo si costruisce in inglese.

L'ultima istruzione chiama la sintesi vocale che rende possibile la pronuncia del messaggio precedentemente costruito.

A questo punto possiamo testare la nostra app con l'emulatore: se la procedura è stata seguita correttamente, l'emulatore ci mostrerà il risultato e la sintesi lo pronuncerà.

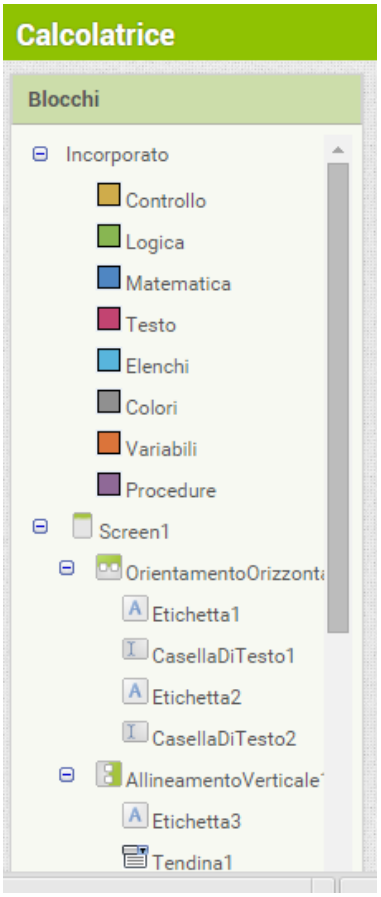

Figura 7. I blocchi.

4.3. Terza fase (generazione dell'eseguibile dell'app e installazione su un dispositivo reale)

Dopo aver controllato sull'emulatore che tutto funzioni correttamente, l'applicazione è completa e può essere tranquillamente scaricata su un dispositivo Android reale. La generazione dell'eseguibile (file.apk) si ottiene selezionando dal sottomenu *Compila* la voce "Salva il file apk sul mio computer".

Si possono aggiungere migliorie: ad esempio è possibile cambiare il testo del pulsante di calcolo ogni volta che viene selezionata una diversa operazione dalla *Tendina 1*. Si possono cambiare i colori dello sfondo delle aree dell'applicazione e le dimensioni dei font per renderla più chiara ed accessibile.

5. Conclusioni

Nel riportare in maniera parziale l'esperienza in corso ci si è limitati a sottolineare solo una parte, quella legata alle calcolatrici, che costituisce un nucleo dotato di una relativa autonomia. Come già sinteticamente segnalato, l'attività si inserisce in un più ampio percorso che ha visto l'attenzione verso le competenze digitali come elemento trasversale a partire sin dai primi anni della scuola primaria. Per una valutazione adeguata, sia di quanto realizzato sia di quanto è ancora in corso d'opera, andrebbero prese in considerazione e documentate una serie di elementi come l'integrazione tra coding e le altre dimensioni dell'apprendimento, l'articolazione della progettazione didattica complessiva, gli effettivi livelli di apprendimento raggiunti. Fermandosi però su un aspetto fondamentale del percorso – quello dell'interazione tra gli studenti delle scuole secondarie di secondo grado, che hanno dovuto acquisire livelli di competenza legati non solo all'aspetto digitale ma anche alla pratiche di insegnamento, e gli studenti della scuola primaria, che hanno realizzato l'app – emerge una percezione positiva legata ad un compito autentico che rimette dinamicamente in gioco motivazione, apprendimento e competenze.

Bibliografia

App Inventor. http://appinventor.mit.edu/explore/ (ver. 30.03.2016).

Calvani, A., Fini, A., & Ranieri, M. (2010). *La competenza digitale nella scuola. Modelli e strumenti per valutarla e svilupparla*. Trento: Erickson.

Code. https://code.org/ (ver. 30.03.2016).

Jenkins, H., Purushotma, R., Weigel, M., Clinton, K., & Robinson, A. (2010). *Culture partecipative e competenze digitali. Media education per il XXI secolo*. Milano: Guerini.

Gmail. https://mail.google.com (ver. 30.03.2016).

Google Play Store. https://play.google.com/store/apps (ver. 30.03.2016).

Papert, S. (1984). *Mindstorms. Bambini, computers e creatività*. Milano: Emme edizioni.

Scratch. https://scratch.mit.edu/ (ver. 30.03.2016).

Standford, J.F., & Naidu, J.T. (2016). Computational thinking concepts for grade school. *Contemporary Issues in Educational Research, 9*(1), 23–32.

http://www.cluteinstitute.com/ojs/index.php/CIER/article/view/9547/9616 (ver. 30.03.2016).

Wing, J.M. (2006). Computational thinking. *Communication of the ACM*, *49*(3), 33–35. https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf (ver. 30.03.2016).