

Università degli Studi dell'Insubria

Facoltà di Scienze MM.FF.NN.

Corso di Laurea Triennale in Informatica



Sviluppo di un'applicazione Android per l'apprendimento della lingua inglese

Relatore: Dott. Mauro Ferrari

Tesi di Laurea di: Francesco Belloni

Matricola 725676

Anno Accademico 2016-2017

Indice

1. Introduzione	2
1.2 E-learning.....	3
2. Analisi dei requisiti	5
3. Progettazione	7
3.1 Descrizione generale.....	7
3.2 Specifica dei requisiti	8
3.2.1 Interfaccia utente	8
3.2.2 Come realizzare il layout.....	9
3.3 Database.....	11
3.4 Web server Java	13
3.5 Web Server C#	14
4. Sviluppo.....	16
4.1 Creazione della base di dati	17
4.2 IntentService.....	18
4.3 Activity	19
4.4 Registrazione e login	22
4.5 Gamification	23
4.6 Lettore Mp3.....	23
4.7 Sicurezza	25
4.9 Test API.....	27
5. Conclusioni	28
Allegato A – Specifica dei requisiti.....	30
Citazioni	36

1. Introduzione

Il progetto di tesi si propone di sviluppare una applicazione mobile Android (da adesso in poi riferita come App) che permetta il rafforzamento della conoscenza dell'inglese attraverso una vasta gamma di attività, che vanno dalle tradizionali lezioni grammaticali agli esercizi di traduzione da e verso l'inglese, dalle prove sulla valutazione della comprensione orale a quelle che si concentrano sulle competenze della lingua scritta.

La App deve essere un supporto per l'utente del sito esercizinglese.com, che avrà la possibilità di fruire dei contenuti sul proprio device, anche in modalità offline. Tutto lo sviluppo del progetto è stato eseguito con un processo a cascata: la prima e storica metodologia di sviluppo del software. Secondo questo modello, il processo di realizzazione del software è strutturato in una sequenza lineare di fasi o passi, che comprende:

1. analisi dei requisiti
2. progetto
3. sviluppo
4. collaudo
5. manutenzione

Ogni fase produce uno specifico documento progettuale, che viene usato per iniziare la fase successiva.

Si è deciso di usare il modello a cascata (*waterfall model* in inglese) poiché offre ancora diversi vantaggi, nonostante questa storica metodologia sia stata superata nell'industria del software.

Il principale beneficio del modello a cascata è la sua struttura lineare che rende ideale la stesura della tesi, mostrando tutti i passaggi che si incontrano nello sviluppo di un software.

Un altro vantaggio è prettamente economico. Il grafico che si occupa della realizzazione della App è una persona esterna al team e per contenere i costi è conveniente far realizzare in un unico preventivo le circa 30 schermate che la App necessita.

La rigidità del waterfall, apprezzabile dal punto di vista organizzativo, diventa tuttavia il suo principale svantaggio quando si sviluppa un progetto di grandi dimensioni. Infatti, un errore nella fase progettuale si ripercuote a cascata in tutto il progetto.

La nostra App è però una diversa implementazione su un diverso device di contenuti già sviluppati, il che ne limita errori progettuali.

1.2 E-learning

Per apprendimento online (noto anche con il termine inglese *E-learning*) s'intende l'uso delle tecnologie multimediali e di Internet per migliorare la

qualità dell'apprendimento facilitando l'accesso alle risorse e ai servizi, così come anche agli scambi in remoto e alla collaborazione a distanza [1].

Il mercato globale della formazione e-learning continua a crescere e ad evolversi. Lo dimostrano l'incremento del budget per i programmi di formazione online, la crescente diffusione dell'e-learning nei vari mercati in tutto il mondo, i nuovi trend nel settore delle tecnologie e degli strumenti formativi, nonché il ruolo sempre più centrale della formazione sociale nelle priorità dei leader L&D (Learning & Development) [2].

Di seguito la pubblicazione della App su Google Play.



**Esercizi Di Inglese e Grammatica
Gratis Offline**

esercizinglese.com Istruzione

PEGI 3

Contiene annunci

- Si tratta di un'app non ancora pubblicata, pertanto potrebbe essere instabile.
- L'app è compatibile con il tuo dispositivo.

Installata



ESERCIZINGLESE.COM

**Spiegazioni
chiare con
tanti esempi**

LEZIONE 1

"To be or not to be" is a quotation from Hamlet

COME SI FORMA L'INFINITE

l'infinito presente in inglese si rende con la parola to be con la caduta della forma base del

2. Analisi dei requisiti

In ingegneria del software, l'analisi dei requisiti è la prima fase del processo di sviluppo ed è un'attività fondamentale per la buona riuscita del prodotto. Lo scopo principale è quello di definire tutte le funzionalità offerte. L'analisi dei requisiti si conclude con la stesura di una dettagliata documentazione chiamata *specificazione dei requisiti* che descrive le funzionalità del nuovo software nella loro interezza.

La specifica dei requisiti è il documento principale e guida tutte le fasi successive dello sviluppo del software. La App vuole offrire gli identici contenuti del sito web ma fruibili anche offline, ovvero:

- Fase di registrazione (*e-mail e password*)
- Fase di login
- Lezioni di grammatica inglese
- Esercizi di grammatica (*fill the gaps, crocette*)
- Gamefication dell'apprendimento
- Esercizi di traduzione
- Esercizi audio
- Letture di brani
- Verbi irregolari

- Phrasal verbs
- News del giorno
- Possibilità di scaricare i nuovi contenuti prodotti dal sito

Oltre a dei requisiti funzionali, la App deve anche rispondere di requisiti non funzionali quali:

- Tempi di risposta reattivi
- Sicurezza
- Grafica accattivante
- Facilità di utilizzo dell'App

Il sito web esercizinglese.com fa parte di un network di siti di e-learning (*progettospartaco.it – libera la cultura*). Tutti i siti del network condividono in effetti la stessa struttura dati e quindi un altro requisito fondamentale deve essere il massimo riutilizzo del codice per le prossime App.

L'output dell'analisi dei requisiti è la *specifica dei requisiti* che viene allegata come appendice A, essendo risultata essere un documento sostanzioso.

3. Progettazione

Partendo dalla *specifica dei requisiti*, la fase di progettazione definisce come tali requisiti saranno soddisfatti, entrando nel merito della struttura del sistema software, pur tuttavia rimanendo una parte distinta della programmazione [11].

3.1 Descrizione generale

Questo paragrafo fornisce una panoramica di tutto il sistema e spiega come esso interagisce con tutti i suoi sottosistemi: l'applicazione mobile e le API del web server `progettospartaco.it`, di cui `esercizinglese.com` fa parte.

Nella figura 1 si possono vedere tutte le tecnologie interessate.

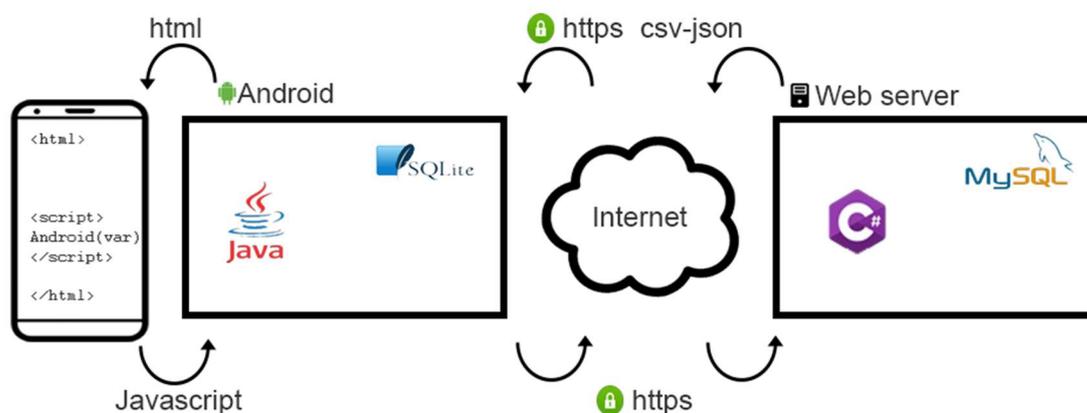


Figura 1 Sistema nel suo complesso

L'applicazione, come da specifiche, deve essere fruibile offline ma contemporaneamente deve offrire la possibilità di accedere alla rete per poter scaricare gli ultimi contenuti prodotti per il sito web.

Per visualizzare i contenuti si è ricorsi ad una *webView* dove il browser nativo di Android (*Chromium*) renderizza il codice HTML in una grafica gradevole e familiare al sito web. L'HTML viene prodotto localmente dal codice Java, ricreando la simulazione di una navigazione tra le varie pagine dell'App. La comunicazione tra la *webView* e il cuore Java dell'applicazione è affidata ad una speciale interfaccia che Android fornisce.

Il server web invece implementa delle API pubbliche a cui l'App, grazie a semplici chiamate https, accede quando vuole comunicare con il server web o deve scaricare i nuovi esercizi.

3.2 Specifica dei requisiti

Questa sezione contiene tutti i requisiti funzionali e di qualità del sistema. Dà una descrizione dettagliata del sistema e tutte le sue caratteristiche.

3.2.1 Interfaccia utente

Contemporaneamente alla stesura dell'analisi dei requisiti si procede anche con la schematizzazione di un possibile layout.

Infatti, avendo bene in mente i requisiti di ogni singola pagina, si riesce a disegnare a mano una prima bozza: si inseriscono i vari pulsanti, le textbox e tutti gli elementi grafici necessari.

Riportiamo come esempio l'homepage della applicazione. La figura 2, ovvero bozza disegnata a mano, insieme ad una breve descrizione è stata utilizzata per creare la homepage della nostra App (figura 3).



Figura 2 bozza a mano del design della home page

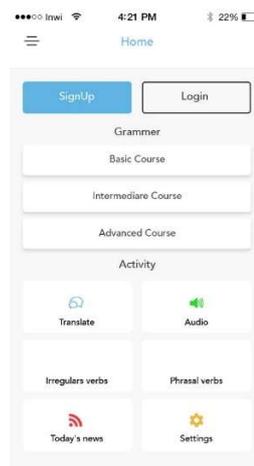


Figura 3 idea della home page prodotta dal grafico

Per massimizzare lo spazio dedicato alla lettura e agli esercizi, si è scelto di avere una *navbar* che possiede solamente due pulsanti: “menù” e “torna indietro”.

3.2.2 Come realizzare il layout

Nei requisiti del progetto è espressamente richiesto:

- 1 la App deve essere riusabile per altri siti di e-learning con il minimo lavoro di integrazione.
- 2 l'intero progetto deve essere concluso in tre mesi.

Per rispettare questi vincoli e strutturare le varie pagine con una grafica accattivante si è quindi deciso di sviluppare un'applicazione ibrida.

Le app ibride sono applicazioni scritte in un linguaggio cross-platform (in genere Javascript + HTML5), che però possono essere incapsulate tramite Web View nel linguaggio nativo di una certa piattaforma. Si possono quindi scaricare dagli store di riferimento ed installare sui rispettivi dispositivi: hanno un discreto accesso al sistema operativo anche se non sono state scritte interamente nel linguaggio specifico della piattaforma [3].

Il mondo Android offre delle soluzioni per rendere il processo di sviluppo di App native più rapido, un esempio può essere il software Cordoba. Nel nostro progetto si è deciso però di sviluppare interamente tutto il codice, evitando di utilizzare componenti terze.

Il grande vantaggio delle App ibride, ossia la possibilità di poter utilizzare HTML combinato con Javascript, è l'abbattimento dei tempi di realizzazione del layout. Per poter però ricreare la sensazione di navigazione tra le diverse pagine, si deve però sviluppare un servizio che simuli il comportamento web server.

Per avere lo stesso feeling tra le diverse schermate si è deciso di usare un *framework*, in particolare materializecss.com.

Un framework nello sviluppo software, è un'architettura logica di supporto su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore [4]. Materializecss è dunque una raccolta di strumenti per la creazione di siti e applicazioni. Esso contiene

modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.

La caratteristica principale di `materializecss` consiste nell'essere basato sul Material Design di Google, la grafica suggerita dalla stessa google per sviluppare le App Android.

3.3 Database

Per poter offrire all'utente un'esperienza completamente offline l'App avrà bisogno che tutti i dati necessari siano salvati sul dispositivo, ovvero necessita di un database sql.

Per introdurre l'argomento database in Andorid è necessario effettuare una piccola premessa sul processo di installazione.

Per installare un nuovo programma in Android si inizia con l'utente che effettua il download, solitamente dallo store di GooglePlay, di un file *'apk'*.

Questo file è un archivio compresso che contiene tutti i file necessari alla App. Completato il download, il programma viene decompresso in una nuova cartella [5], nel nostro caso in `"/data/esercizinglese"`.

Se il database si trovasse direttamente nel file apk, una volta effettuata la decompressione, lo si ritroverebbe nella cartella `"/asset"`, quindi con l'impossibilità di essere modificato e nemmeno cancellato [6]. Questa impostazione di Android è stata la prima vera scelta progettuale:

- opzione 1: al primo avvio dell'applicazione i dati che risultano salvati in un database sqlite compresso nel nostro file apk, vengono copiati in un nuovo database. Questo nuovo database avrà la possibilità di poter essere anche usato in modalità scrittura.
- opzione 2: al primo avvio dell'applicazione si creano le tabelle del nostro database e si scarica da Internet il contenuto necessario.

Si riassumono i pro e i contro dei due approcci con due tabelle:

Opzione 1

Vantaggi	Svantaggi
L'operazione di copiatura del database è molto veloce [meno di 10 secondi].	Due database e quindi doppio spazio occupato.

Opzione 2

Vantaggi	Svantaggi
Minor spazio richiesto dalla Applicazione	L'operazione per scaricare e inserire i dati nel database è molto lenta [circa 1 minuto].
L'utente scarica direttamente la App con tutte le ultime risorse del sito	Serve un servizio che si occupi di gestire tutte le richieste di installazione, fornendo i dati in csv o json per popolare il database
Per le opzioni di aggiornamento ed inserimento di nuovo materiale si dispone già di un servizio di API, che andrebbe comunque sviluppato	

La scelta di offrire un servizio per l'installazione risulta molto più onerosa sia in termini di tempo di sviluppo sia come esperienza utente. Tuttavia, questa scelta è stata preferita poiché permetterà alle successive App del network una maggiore velocità di sviluppo.

Una volta recuperati i dati, Android facilita l'uso di un database grazie alle classi *Helper*.

Al primo avvio vengono create e popolate 7 tabelle. Ogni tabella è indipendente dalle altre.

Ogni pagina che viene visualizzata dalla App, accede alle informazioni salvate nel database ma grazie alla velocità di SQLite le query restituiscono risposte in tempo di pochi millisecondi.

3.4 Web server Java

Per una maggiore rapidità di sviluppo e per velocizzare le future App di elearnig, come specificato nell'analisi dei requisiti, l'utilizzo di una webview è stata quasi obbligata. La spiegazione di come si è implementato un piccolo web server locale sarà esposta in dettaglio nel capitolo 4 dedicato alla programmazione.

In questa fase del progetto si deve tener conto che il web server deve essere in grado di poter restituire velocemente l'html prodotto della specifica pagina richiesta alla webview e poter chiamare dei servizi che invece devono agire in background per operazioni più lente. Ovviamente il web server deve utilizzare solamente le risorse installate sul dispositivo e non vi

sarà nessuna richiesta http. Come schematizzato in figura 4, il codice Javascript richiede una nuova pagina al web server, una volta generata, è caricata direttamente nella webview dell'applicazione.

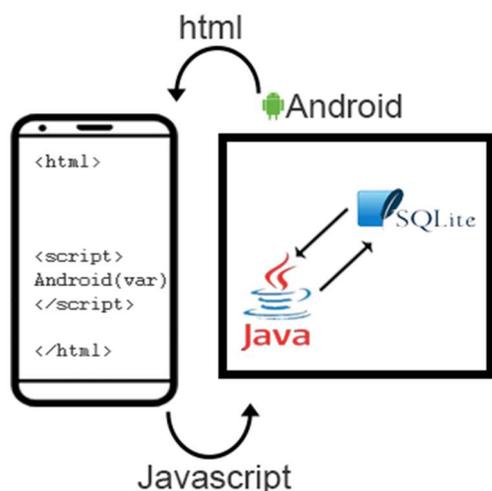


Figura 4 simulazione di un web server locale in Java

Per permettere l'interazione tra il codice Javascript della pagina e il codice Java, Android mette a disposizione una speciale interfaccia.

3.5 Web Server C#

Per minimizzare la quantità di spazio necessario della App e per fornire in un futuro delle API pubbliche dei vari contenuti del network, è stato creato un servizio API.

Quello che si vuole sviluppare è un servizio REST. Il termine REST è spesso usato per descrivere ogni semplice interfaccia che trasmetta dati su HTTP senza un livello opzionale come SOAP o la gestione della sessione tramite i cookie [12].

Questo servizio serve per accedere direttamente ai dati immagazzinati nel database Mysql dei vari siti internet. La risposta del server è in formato csv: le varie colonne sono separate dalla virgola mentre i dati saranno racchiusi tra virgolette.

Per poter accedere alle api di progetto Spartaco, bisogna navigare su un indirizzo speciale e scegliere opportunamente i parametri querystring.

Traduzioni

- **&c=21** ritorno la lista di tutti gli argomenti da tradurre

<http://www.progettospartaco.it/mobile.api?s=6&c=21>

Return
CSV list

- **&c=22** ritorno la lista di tutti le frasi da tradurre (database completo)
 - &p1= id dell'argomento da visualizzare (se passo il parametro -1 scarico tutte le traduzioni)

<http://www.progettospartaco.it/mobile.api?s=6&c=22>

Return
CSV list

- **&c=23** ritorno la lista di tutti le frasi da tradurre selezionate per l'ID dell'argomento
 - &ps1= id dell'argomento da visualizzare
 - oppure
 - &ps1=1-80 id degli argomenti da visualizzare, separati dal carattere -

<http://www.progettospartaco.it/mobile.api?s=6&c=23&ps1=1-80>

Return
CSV list

Figura 5 Pagina web di riepilogo di tutte le API sviluppate

4. Sviluppo

La programmazione è un processo che porta da una formulazione originale di un problema informatico a programmi per computer eseguibili. La programmazione comprende attività come l'analisi, la generazione di algoritmi, la verifica dei requisiti degli algoritmi (inclusa la loro correttezza e consumo di risorse) e l'implementazione in un linguaggio di programmazione target. Lo scopo della programmazione è di trovare una sequenza di istruzioni che automatizza l'esecuzione di un compito specifico o la risoluzione di un determinato problema. Il processo di programmazione spesso richiede esperienza in molte materie diverse, compresa la conoscenza del dominio dell'applicazione, algoritmi specializzati e logica formale [9].

Per lo sviluppo del codice si è partiti da tutti i documenti prodotti durante la fase di progettazione, andando incrementalmente ad implementare tutte le caratteristiche richieste.

Inizialmente si è andati a scrivere il codice per la creazione, il popolamento e l'accesso alla base di dati. Successivamente si sono create tutte le pagine in html. Infine, si è creato il web server locale in Java e tutti i servizi necessari.

4.1 Creazione della base di dati

In base alle specifiche della App, viene ricreata solo una parte delle strutture dati presenti nel database del sito. Come mostrato in figura 6, la App necessita solamente di dieci tabelle. Nelle tabelle sono salvati tutti gli esercizi e lezioni che l'utente può svolgere.

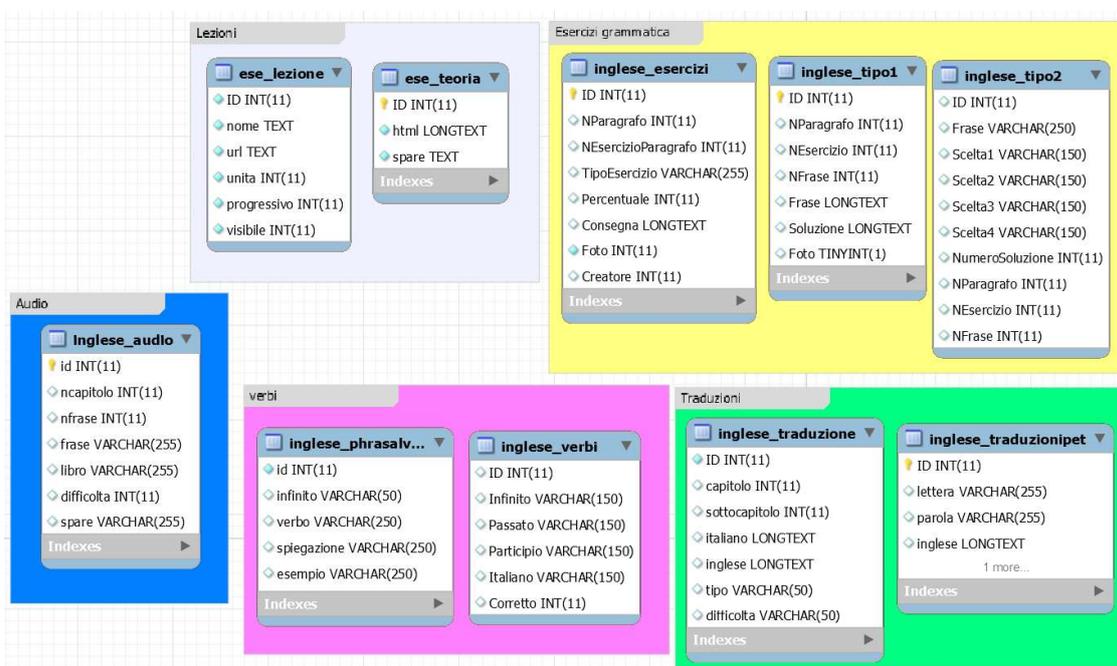


Figura 6 Panoramica delle tabelle usate nel progetto

La creazione, modifica o cancellazione dei vari database SQLite avviene attraverso classi Helper (*SQLiteOpenHelper*) che il sistema Android mette a disposizione.

Per rendere l'accesso al database "*object oriented*", sono state scritte delle classi Java che ricalcano la struttura del database. Oggi esistono framework ORM (object-relational mapping) il cui compito è quello di nascondere il funzionamento di un database relazionale, permettendo di pensare al modello dei dati come ad un insieme di oggetti. Vista la quantità limitata di tabelle e relazioni tra esse, si è deciso di scrivere a mano tutte le classi.

Una volta create le tabelle necessarie al primo avvio, si è creato un servizio per recuperare i dati necessari.

4.2 IntentService

Quando l'applicazione viene lanciata Android fa partire il processo principale (*main thread*) che si occupa principalmente dell'interazione con l'utente e dell'interfaccia grafica.

Invece, per operazioni che non necessitano di modificare l'interfaccia grafica e sono di lunga durata, il sistema operativo permette l'aggiunta di altri processi che lavorano in background (un esempio può essere la classe *IntentService*).

Per creare e popolare i database la prima classe Java creata è stata *MyWebRequestService*, che estende appunto la classe *IntentService*.

Come spiegato nel capitolo 3 la creazione dei database viene gestita grazie alle classi *DatabaseHelper*. Invece per popolare le tabelle la App deve effettuare delle semplici chiamate http, recuperare i dati in formato csv, elaborarli e infine inserirli con una query.

MyWebRequestService si occupa inoltre di comunicare con il mondo esterno per altri servizi:

- per iscrivere l'utente
- per effettuare il login
- per aggiornare l'applicazione, scaricando nuovi contenuti
- salvare i punteggi sul server web

La classe *IntentService* dispone di un metodo *sendBroadcast(Intent)* che passa il risultato del lavoro svolto in background al main thread. Per ricevere i dati bisogna però inizializzare una apposita classe *BroadcastReceiver*, nel nostro caso *MyWebRequestReceiver*.

4.3 Activity

La classe *Activity* è un componente cruciale di un'App Android. A differenza dei paradigmi di programmazione in cui le app vengono lanciate con un metodo *main()*, il sistema Android avvia il codice in un'istanza *Activity* richiamando metodi di callback specifici che corrispondono a fasi specifiche del suo ciclo di vita.

La nostra applicazione avendo solamente una *webView* è costituita da una sola *activity*. Il cuore dell'applicazione è quindi la classe *MainActivity*.

Essa racchiude 2 sottoclassi: *MyTask* e *WebAppInterface* il cui compito è di simulare il web server.

WebAppInterface è la classe che contiene il metodo *JsInterface* che si occupa di far comunicare il Javascript con la parte Java dell'applicazione. Quando l'utente preme un pulsante o un link questa azione, che solitamente viene gestita con una chiamata http, viene intercettata nella App da questo metodo. L'argomento passato dal Javascript è una stringa che contiene tutte le informazioni necessarie delimitate da caratteri speciali. Una volta che la WebAppInterface ha ricevuto la stringa, crea un oggetto *QuerystringData*. Esso racchiude tutte le informazioni necessarie al web server locale per creare e visualizzare la pagina richiesta. Quando l'utente vuole navigare da una pagina ad un'altra viene richiamata la classe *MyTask*. Essa si occupa di creare e far visualizzare l'html prodotto nella webview. *MyTask* è una classe che estende *AsyncTask* ovvero la soluzione che Android consiglia per brevi operazioni da effettuare in background ma necessitano di usare i dati elaborati nel processo principale.

In figura 7 si vede, grazie al sequence diagram, il processo di visualizzazione di una nuova pagina.

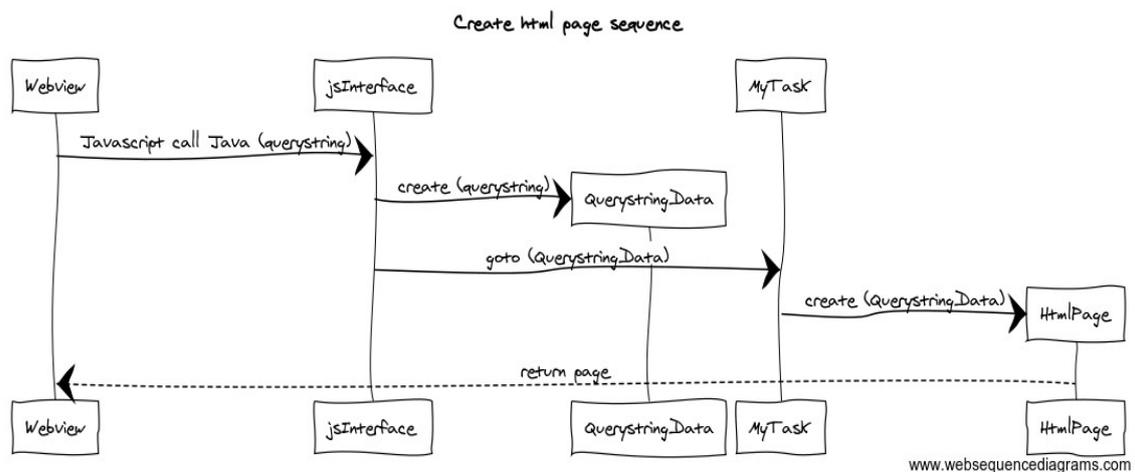


Figura 7 Sequence diagram della creazione di una pagina

La creazione della pagina html è molto semplice, come si può notare dal questo piccolo esempio di codice: in base al parametro *pagename* che identifica quale pagina si vuole creare, viene creata una nuova classe *PageData* opportunatamente inizializzata.

```
public Masterpage(QuerystringData myQuerystringData) {  
  
    switch (myQuerystringData.myPageName) {  
  
        case HOMEPAGE:  
            myPageData = new HomePage();  
            break;  
        case LISTGRAMMAR:  
            myPageData = new ListGrammarLessons(myQuerystringData.lessonID);  
            break;  
        case LESSON:  
            myPageData = new GrammarLesson(myQuerystringData.lessonID);  
            break;  
        case EXERCISETYPE1:  
            myPageData = new ExerciseType1(myQuerystringData.isPostBack, myQuerystringData.lessonID);  
            break;  
        case EXERCISETYPE2:  
            myPageData = new ExerciseType2(myQuerystringData.isPostBack, myQuerystringData.lessonID);  
            break;  
        case LISTTRANSLATE:  
            myPageData = new ListTranslate();  
            break;  
        case LISTTRANSLATEBYTOPIC:  
            myPageData = new ListTranslateTopic(myQuerystringData.lessonID);  
            break;  
    }  
}
```

Figura 8 Codice per la creazione di una nuova pagina html

Tutte le pagine web sono create dalle loro classi omonime, per esempio l'home page viene creata dalla classe *homepage.java*. In particolare, ogni pagina estende la classe *PageData* si può definire come una *masterpage*: la pagina principale che definisce l'aspetto e il comportamento standard che si desidera per tutte le pagine nella propria applicazione [10].

Finito di elaborare l'html viene riconsegnato alla classe *MyTask* che si occupa di aggiornare la webview.

4.4 Registrazione e login

Per creare un database di utenti e una possibile community che possa ruotare attorno al sito ed all'App, si è scelto di dividere gli utenti in tre tipologie:

- utenti non registrati che possono usufruire di alcuni contenuti e visualizzano la pubblicità.
- utenti registrati che possono usufruire di tutti i contenuti ma continuano a visualizzare la pubblicità.
- utenti premium sono come gli utenti registrati ma hanno acquistato dei punti sul sito quindi non visualizzano la pubblicità.

La fase di registrazione avviene effettuando una chiamata https con i dati dell'utente concatenati e criptati.

Una volta che l'utente avrà confermato la mail ed effettuato il login si salvano sul disco, tramite il sistema di *SharedPreferences*, la mail e la password criptate. Queste credenziali verranno usate ogni volta che l'utente vuole effettuare un'operazione ritenuta sensibile con le API del sito.

4.5 Gamification

Per cercare di limitare uno dei più grandi problemi dell'e-learning ovvero l'alto tasso di abbandono, si è sviluppato un sistema di gamification dell'esperienza: quando l'utente riesce a completare un esercizio, in base alla percentuale di successo, guadagna dei punti.

Tutti i punti guadagnati nell'App devono essere sincronizzati con il sito web e viceversa. Per ogni esercizio, se l'utente si trova online, viene chiamata la API dedicata al salvataggio.

Per visualizzare i vari risultati degli esercizi, viene utilizzato un piccolo script in Javascript, che viene eseguito dopo che la pagina è stata caricata.

4.6 Lettore Mp3

Un aspetto importante nell'imparare una lingua è il "*listening*", ovvero riuscire a comprendere la lingua parlata. Nel corso degli anni sono stati prodotti diversi mp3 che vengono proposti come esercizi.

Un problema inaspettato nello sviluppo dell'applicazione è stato scoprire che il sistema Android, per questioni di sicurezza, non permette di poter eseguire direttamente nella webview un qualsiasi player html5 che acquisisca i file audio dal web.

Il sistema operativo fortunatamente dispone di un *mediaPlayer* integrato che gestisce, tramite Api di sistema, la riproduzione di un audio. In figura 9 si può osservare il solito meccanismo di interazione: l'utente premendo su

pulsante richiama il Javascript e invoca la funzione Java passando uno dei seguenti parametri:

- Load mp3
- Play
- Pause
- Stop
- Goto(second)

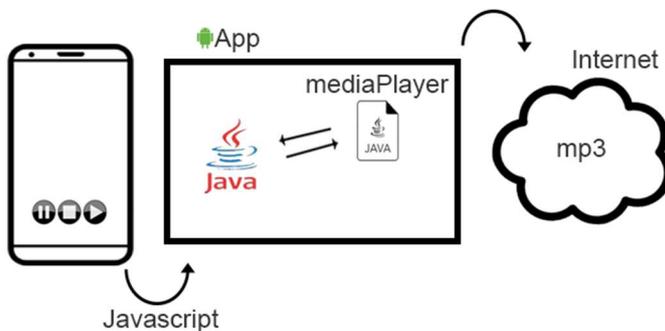


Figura 9 Lettore mp3

In figura 10, invece, si riporta un pezzo di codice.

```
1 case PLAYMP3:
2 Mp3Player myPlayer = new Mp3Player(myQuerystringData);
3 switch (myQuerystringData.Mp3State) {
4     case Variables.mp3PlayerCommandLoad:
5         myPlayer.LoadMp3();
6         SetMp3Length(myPlayer.totalTimeString, myPlayer.totalSeconds);
7         break;
8     case Variables.mp3PlayerCommandPlay:
9     case Variables.mp3PlayerCommandPause:
10        myPlayer.Play();
11        break;
12    case Variables.mp3PlayerCommandStop:
13        myPlayer.Stop();
14        break;
15    case Variables.mp3PlayerCommandGoto:
16        myPlayer.Goto(Integer.valueOf(myQuerystringData.mixValue));
17        break;
18 }
```

Figura 10 Codice che gestisce il comportamento del player

4.7 Sicurezza

Un altro requisito richiesto per la App è la possibilità di registrarsi al sito internet e alla relativa App fornendo e-mail e password. Se l'utente risulta già iscritto al sito bisogna offrire la possibilità di effettuare il login.

Le richieste di autenticazione vengono gestite tramite un servizio API REST che però espone le credenziali dell'utente. Infatti, nonostante il sito disponga di una connessione https, nei log degli accessi si potrebbe trovare una stringa:

```
https://api.website.com?mail=mail@yhaoo.com&password=mypassword
```

Per gestire in sicurezza le informazioni dell'utente in fase di registrazione, login e salvataggio degli esercizi, si è deciso di criptare e-mail e password prima di usarle per autenticarsi presso il server web.

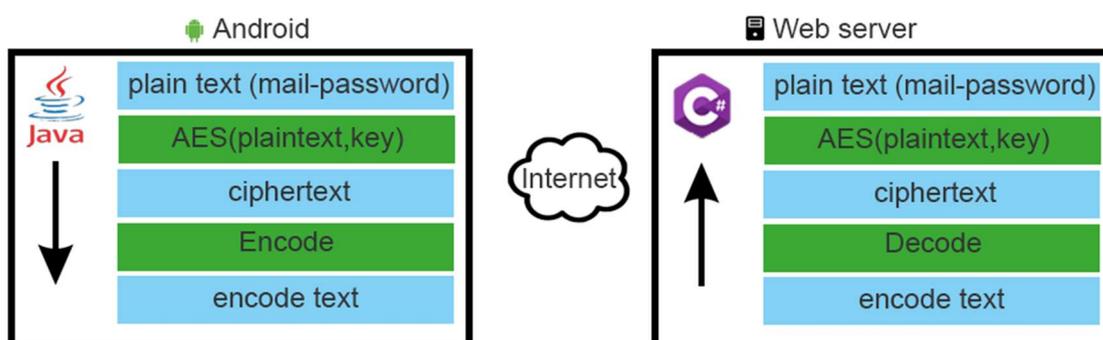


Figura 11 Criptazione delle informazioni sensibili

Come si può osservare in figura 11, il processo di criptazione deve essere effettuata nel linguaggio Java mentre la decriptazione avviene usando il linguaggio C#.

Fortunatamente entrambi i linguaggi hanno delle librerie dedicate alla sicurezza. Per criptare si è usato l'algoritmo di cifratura *Advanced Encryption Standard*, mentre per la codifica e la decodifica del testo criptato si è usato l'algoritmo *Base64*.

L'Advanced Encryption Standard (AES) è un algoritmo di cifratura a blocchi che implementa una rete a sostituzione e permutazione utilizzando inoltre il principio crittografico di Shannon di "confusione e diffusione". AES è veloce sia se sviluppato in software sia se sviluppato in hardware, è relativamente semplice da implementare, richiede poca memoria ed offre un buon livello di protezione/sicurezza [7].

L'algoritmo Base64 è un sistema di numerazione posizionale che usa 64 simboli. Viene usato principalmente come codifica di dati binari nelle e-mail per convertire i dati nel formato ASCII [8].

Si deve utilizzare questo ulteriore passaggio di codifica in Base64 poiché non tutti i caratteri in un url sono ammissibili e spesso il testo criptato è composto da caratteri non accettati.

4.9 Test API

Per il testare le API è stato creato un piccolo programma Windows che richiama tutti gli URL e controlla che:

- l'url sia raggiungibile
- venga restituita una risposta
- il csv sia ben formattato
- la sequenza dei tipi dei vari campi del csv (es. int, string, int) sia corretta
- Il tempo di risposta del server

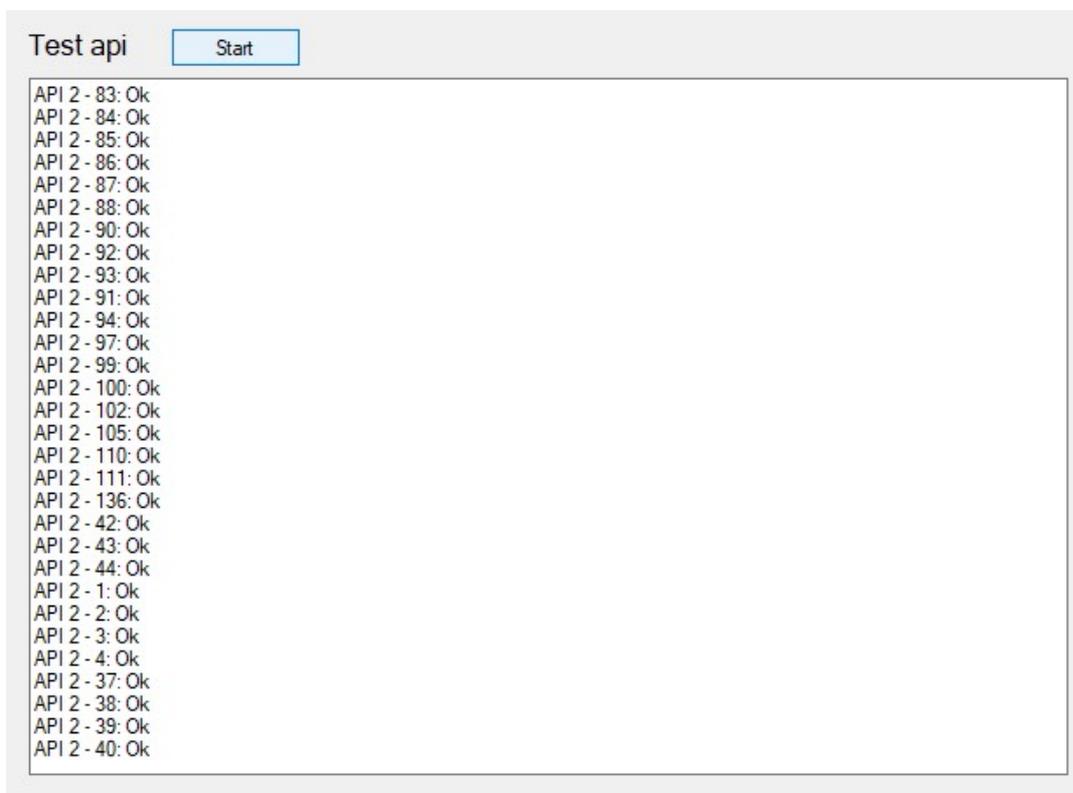


Figura 12 Test di tutte le api

5. Conclusioni

Il Corso di Laurea in Informatica si pone come obiettivo di fornire una solida conoscenza di base e metodologica dei principali settori dell'informatica, la conoscenza delle tecnologie proprie del settore con lo scopo di fornire gli strumenti necessari a comprendere ed assimilare le costanti innovazioni che caratterizzano le scienze informatiche. [13]

Il progetto di tesi mi ha permesso infatti, attraverso tutti i passaggi necessari allo sviluppo di un software relativamente complesso, di applicare le conoscenze acquisite durante il corso di Laurea.

In particolare, la creazione di un *framework* riutilizzabile per altre App di e-learning mi ha obbligato a utilizzare dei *design pattern* noti per mantenere il codice gestibile e riutilizzabile. Il lavoro di scrittura, che ha richiesto un oneroso sforzo lavorativo, mi ha permesso però di modellare il framework secondo le effettive esigenze, con il risultato di non aver appesantito la App di funzionalità non desiderate.

Le difficoltà maggiori le ho incontrate con il sistema operativo Android nel comprendere come poter implementare efficacemente tutte le funzioni richieste. Lo sviluppo per i dispositivi mobili, con il loro ridotto accesso alle

risorse, obbliga a dover dividere il lavoro su più thread aumentando la difficoltà nel farli comunicare tra loro.

Questa tesi mi ha confermato ulteriormente la necessità di avere una visione globale del problema per poter affrontare con la giusta professionalità tutto lo sviluppo di un software. Infatti, tutti i vari piccoli problemi incontrati sono stati risolti con soluzioni che non compromettessero la struttura del framework costruito.

Riassumendo posso affermare che dal punto di vista tecnico ho ampliato la mia conoscenza del sistema operativo Android e lo sviluppo di un'interfaccia API REST. Dal punto di vista organizzativo questo lavoro di tesi mi ha permesso di capire la difficoltà di stimare i tempi relativi ad un progetto importante e rispettare le scadenze preventivate.

Gli sviluppi futuri di questa App dovrebbero andare incontro alle varie esigenze che emergeranno come feedback dagli utilizzatori. Tutto il lavoro invece che ha portato alla creazione di un framework proprio ma riutilizzabile per altre App di e-learning del gruppo sarà ulteriormente migliorato.

esercizinglese Android App

Specifica dei Requisiti

Varese 02/10/17

Introduzione

La presente sezione ha lo scopo di riportare la visione globale dell'intero documento di Specifica dei Requisiti. La struttura del documento è quella suggerita dallo standard ANSI/IEEE 830 noto come SRS (Software Requirements Specifications)

Obiettivo

Lo scopo del presente documento è di rappresentare, nel modo più preciso, consistente, non ambiguo e comprensibile, la soluzione studiata per lo sviluppo di una app in android per il sito internet di elearning esercizinglese.com.

Più precisamente, la specifica dei requisiti del sistema intende esprimere esclusivamente i bisogni progettuali richiesti, mentre le ipotesi su come tali bisogni possano essere soddisfatti sono argomento del documento di progettazione (architettura del sistema, disegno di massima, aspetti tecnologici).

Campo d'applicazione

Il Marketplace delle App è diventato un luogo molto importante. La App serve per aumentare il prestigio del brand e andare incontro alle esigenze degli utenti.

Descrizione generale

Descriviamo ora i principali fattori che riguardano il prodotto software.

Inquadramento

Il software vuole fornire un accesso ai contenuti presenti sul sito proposti in modalità offline. Il sistema è composto dalle seguenti sezioni principali:

Registrazione

L'utente sceglie il proprio nome utente, una password, un avatar e fornisce una mail valida (alla quale verrà spedito un link di conferma).

Login

L'utente si autentica fornendo mail e password al web server del sito internet. Da quel momento può accedere al sistema di salvataggio dei progressi. Il salvataggio deve avvenire sia in locale sia aggiornando i dati dell'utente che risiedono sul server. Non è possibile gestire più utenti su uno stesso device.

Gamefication dell'esperienza

L'utente guadagna punti effettuando gli esercizi proposti con un punteggio maggiore del 60% o rispondendo alle domande del forum. L'utente spende i punti sbloccando le spiegazioni grammaticali o facendo domande nel forum.

Lezioni di grammatica inglese

Le lezioni di grammatica vengono divise in corso base, corso intermedio e corso avanzato. L'utente può leggere le lezioni del corso base ed intermedio ma deve invece spendere i punti guadagnati per sbloccare le lezioni del corso avanzato.

Esercizi

Variano in base alla tipologia: fill the gaps, crocetta, esercizi audio, verbi irregolari, phrasal verbs. Permettono di guadagnare punti se la percentuale della correzione supera il 60%.

Postare una domanda sul forum

Un utente può chiedere attraverso il forum un dubbio riguardante un esercizio o una spiegazione grammaticale.

Rispondere ad una domanda sul forum

Un utente può rispondere ad una domanda del forum e se la risposta è ritenuta positiva può guadagnare dei punti.

News del giorno

Per aumentare le visite giornaliere viene proposta una lettura della notizia del giorno. Deve essere scaricata da internet, necessita della connessione dati.

Dizionario inglese italiano

Bisogna implementare un dizionario inglese-italiano

Possibilità di scaricare i nuovi contenuti prodotti dal sito

L'utente può scaricare tutti i nuovi aggiornamenti che vengono prodotti.

Interfaccia sistema/utente

Gli utenti che utilizzeranno l'applicazione si suppone che non dispongano, come regola, di una significativa preparazione nell'uso di strumenti informatizzati. Per questo motivo, durante la realizzazione dell'interfaccia utente si dovrà tenere conto della sua accessibilità. L'accessibilità dell'interfaccia utente deve essere approvata tramite test preproduzione.

Semplicità dell'interfaccia utente

Occorre prestare grande attenzione alla progettazione di un'interfaccia utente semplice ed intuitiva.

Usabilità

L'usabilità implica che le informazioni debbano essere organizzate e strutturate in maniera da garantire la massima fruibilità. In altre parole, deve essere realizzata un'interfaccia che abbia lo scopo di essere graficamente accattivante, ma con l'obiettivo che sia essenziale e che

l'informazione sia organizzata in modo tale che gli utenti siano in grado di realizzare i loro obiettivi in modo semplice e non ambiguo.

Navigabilità

Per rispettare questo requisito, bisogna seguire una direttiva essenziale: rendere l'accesso alle informazioni ridondante, ossia il prevedere strade diverse per accedere allo stesso contenuto. Deve essere possibile accedere a ciascuna funzione percorrendo almeno due strade alternative.

Interfaccia hardware

La applicazione deve coprire il maggior segmento di mercato. Da dati Google [<https://developer.android.com/about/dashboards/index.html>] la applicazione deve poter funzionare dalla versione 4.4 e successive.

Interfaccia di comunicazione

La applicazione deve poter comunicare tramite il web server con una comunicazione sicura https per poter effettuare tutte le operazioni necessarie. Devono essere previsti dei controlli per prevenire azioni non conformi al normale utilizzo dell'applicazione.

Vincoli relativi all'occupazione di memoria

L'applicazione deve occupare la minor memoria possibile. Si deve prevedere un sistema interno di gestione della memoria che permettere all'utente di cancellare gli eventuali servizi non necessari.

Vincoli per installazione

L'utente è consapevole che verranno inserite inserzioni pubblicitarie. Se un utente acquista dei punti extra le inserzioni non devono essere visualizzate.

Vincoli generali

Compatibilità logica con i sistemi informativi esistenti, ovvero la logica dell'applicazione Android deve andare a modellarsi e cercare di integrarsi con il sistema informativo già esistente.

Requisiti da analizzare in futuro

L'applicazione deve essere sviluppata pensando che si possa replicare facilmente per gli altri progetti di elearning del gruppo.

Specifiche funzionali e non funzionali del sistema

Vengono proposti solamente tre esempi delle schede dei requisiti funzionali.

RF01	<i>Registrazione</i>
Input	Username, email, password, avatar, provincia di residenza, anno di nascita
Processo	L'utente inserisce: <ul style="list-style-type: none">• username• email• password• avatar• provincia di residenza [dropdown, non obbligatorio]• anno di nascita [non obbligatorio]• spunta di accettare le condizioni di utilizzo dei dati Una volta completati i campi e premuto il pulsante di registrazione viene effettuata una chiamata https per l'invio dei dati al web server. Si visualizza un messaggio di errore se: <ul style="list-style-type: none">• l'utente non ha completato tutti i campi• è assente la connessione internet• utente già registrato
Output	Se la chiamata https è andata a buon fine viene visualizzato un messaggio di operazione completata con successo.

RF02	<i>Login</i>
Input	Email, password
Processo	L'utente inserisce: email

	<p>password</p> <p>Una volta completati i campi e premuto il pulsante di login viene effettuata una chiamata https per l'invio dei dati al web server.</p> <p>Si visualizza un messaggio di errore se:</p> <ul style="list-style-type: none"> ● l'utente non ha completato tutti i campi ● è assente la connessione internet ● mail o password risultano sbagliati ● l'utente non ha confermato la propria mail <p>Se la mail di conferma non viene inviata l'utente può richiederla tramite un pulsante.</p>
Output	<p>Se la chiamata https è andata a buon fine viene visualizzato un messaggio di operazione completata con successo.</p> <p>Una volta che l'utente ha effettuato il login non verranno più visualizzati i pulsanti di registrazione e login nella home page.</p>

RF03	<i>Gamefication dell'esperienza</i>
Input	Risultato dell'esercizio
Processo	<p>Il numero di punti che ottieni dipende dalle azioni che intraprendi all'interno del sito:</p> <ul style="list-style-type: none"> ● sbloccare una lezione avanzata (-5 punti) ● svolgendo un esercizio e ottenendo un voto minore del 60% (-1 punti) ● svolgendo un esercizio e ottenendo un voto minore del 80% (+1 punti) ● svolgendo un esercizio e ottenendo un voto maggiore del 80% (+3 punti) ● un accesso giornaliero alla news (+ 1 punti) ● domanda sul blog (-5 punti) ● risposta votata come positiva sul blog (+ 1 punto) <p>L'utente incomincia con 20 punti.</p>
Output	L'utente può vedere il messaggio di operazione conclusa con successo.

Citazioni

- [1] https://it.wikipedia.org/wiki/Apprendimento_online
 - [2] Mercato e-learning: trend e previsioni 2017-2021 - docebo
 - [3] <https://www.devinterface.com/it/blog/app-ibride-o-app-native-questo-e-il-dilemma>
 - [4] <https://it.wikipedia.org/wiki/Framework>
 - [5] <https://stackoverflow.com/questions/12442979/android-understanding-the-apk-installation-process>
 - [6] <https://stackoverflow.com/questions/9836233/delete-database-file-from-assets-folder>
 - [7] https://it.wikipedia.org/wiki/Advanced_Encryption_Standard
 - [8] <https://it.wikipedia.org/wiki/Base64>
 - [9] https://en.wikipedia.org/wiki/Computer_programming
 - [10] <https://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx?f=255&MSPPError=-2147217396>
 - [11] [https://it.wikipedia.org/wiki/Progettazione_\(ingegneria_del_software\)](https://it.wikipedia.org/wiki/Progettazione_(ingegneria_del_software))
 - [12] https://it.wikipedia.org/wiki/Representational_State_Transfer
 - [13] <http://informatica.dista.uninsubria.it/?q=content/corso-di-laurea-triennaleIntro>
- [Template allegato A]
http://didattica.agentgroup.unimore.it/wiki/images/d/d7/IEEE830_ITA.pdf