



UNIVERSITÀ DI PISA

**Corso di Laurea in Informatica  
Umanistica**

RELAZIONE

**Dal web alle applicazioni native: un'app per verbalizzare  
esami**

**Candidato:** *Sara Dariozzi*

**Relatore:** *Maria Simi*

**Correlatore:** *Elvira Todaro*

Anno Accademico 2010-2011

# Indice generale

1. Introduzione.....	3
1.1 Mobile Web e Mobile Web Initiative.....	3
1.2 Perché sviluppare per iPhone.....	5
1.3 App o Web App?.....	7
1.4 Servizio Statini.....	9
1.5 Gradeup.....	10
1.5.1 Stato dell'arte.....	11
2. Fasi della progettazione.....	12
2.1 jQuery Mobile per lo stile e la struttura.....	12
2.2 Phonegap per accedere all'hardware.....	13
2.3 Progettazione grafica.....	14
2.3.1 Launch image e icone.....	14
2.3.2 Le Interfacce.....	16
2.4 Implementazione degli strumenti e gestione delle interfacce.....	17
2.4.1 Login.....	17
2.4.2 Le interfacce Materia, Esame e Database.....	19
2.4.3 Comunicazione con il Server.....	22
3. Compilare l'app.....	24
3.1 Strumenti di compilazione.....	24
3.1.1 Phonegap Build.....	24
3.1.2 Adobe Dreamweaver.....	25
4. Distribuire l'app: Apple Store.....	26
5. Conclusioni.....	27
5.1 App: Write once, Run Anywhere?.....	27
6. Bibliografia.....	31
7. Appendice.....	34

# 1. Introduzione

Il progetto realizzato è un'applicazione nativa per iPhone, Gradeup, che si interfaccia al sistema telematico di verbalizzazione degli esami via web dell'Università di Pisa: Servizio Statini.

Il progetto è stato creato per avere un'applicazione accessibile e usabile anche ai dispositivi mobili, in quanto Servizio Statini non è fruibile da tali strumenti.

Il metodo di creazione di Gradeup è alternativo a quello usato per lo sviluppo di un'app nativa per iPhone: sono state utilizzate tecnologie web come HTML, CSS e Javascript, un framework per la creazione delle pagine e un compilatore di applicazioni per creare l'app nativa, accedere all'hardware del dispositivo mobile e testarne le funzionalità.

## 1.1 Mobile Web e Mobile Web Initiative

« (...) *The Web is designed, in turn, to be universal: to include anything and anyone. This universality includes an independence of hardware device and operating system, as I mentioned, and clearly this includes the mobile platform. (...)* »

( *Tim Berners-Lee, w3.org, voce Talks by W3C Speakers* )

Il web è stato progettato per essere accessibile a chiunque e a qualsiasi dispositivo, indipendentemente dall'hardware e dal sistema operativo.

Negli ultimi anni, il web si è evoluto molto; i siti, da contenitori di informazione o semplici vetrine, sono diventati sempre più uno strumento di interazione con l'utente, tanto da essere definiti “applicazioni web”.

Si sono evoluti anche gli strumenti con i quali si accede ai servizi e l'attenzione degli sviluppatori si è concentrata sul mercato del dispositivo mobile.

I dati statistici raccolti ed elaborati da Netmarketshare, per il periodo compreso tra il 12/2010 e il 10/2011, mostrano che *“le piattaforme mobili e tablet rappresentano oltre il 5% di tutta la navigazione su Internet. Questi dispositivi comprendono: dispositivi iOS [iPhone, iPad, iPod Touch], [e dispositivi sui quali girano sistemi operativi come] Android, Blackberry, Symbian, Windows Mobile e i dispositivi Java ME.<sup>1</sup>”* (Netmarketshare, voce *Browser Share*) “v. tab. 1”.

I dati mostrano l'aumento costante, mese dopo mese, dell'accesso al web tramite

---

1 Netmarketshare, <http://www.netmarketshare.com/>

smartphone, tablet, iPhone, e contemporaneamente un calo degli accessi da piattaforme desktop.

Nel 2005 alcune società di tecnologia mobile, si sono associate nella *Mobile Web Initiative* (MWI) all'interno del Consorzio del World Wide Web; l'intento del MWI è quello di standardizzare la produzione di contenuti affinché siano accessibili a tali dispositivi.

Il MWI si occupa di fornire le linee guida per migliorare l'interoperabilità e l'usabilità dei contenuti per i dispositivi mobili.

I documenti pubblicati descrivono le caratteristiche dei vari dispositivi ma danno anche informazioni sulle API<sup>2</sup> da mettere a disposizione degli sviluppatori.

Queste caratteristiche aggiungono al web la capacità di comunicare col dispositivo, gestendo: la grafica, il web storage ovvero la memorizzazione dei dati in un browser; la comunicazione attraverso messaggi mediante l'uso di link (sms://, mailto://), la connettività attraverso la rete recuperando informazioni tramite richieste asincrone, la geolocalizzazione che permette di comunicare con i sensori, il microfono, l'accelerometro. E ancora i form che costituiscono uno degli elementi principali di interazione con l'utente, le API per la gestione dell'audio e del video, il packaging per la persistenza dell'applicazione web nel dispositivo e infine un'attenzione particolare all'ottimizzazione del sito web, dal momento che la CPU e la batteria di un mobilephone hanno dei limiti rispetto ad un dispositivo fisso. “v. fig. 1”.

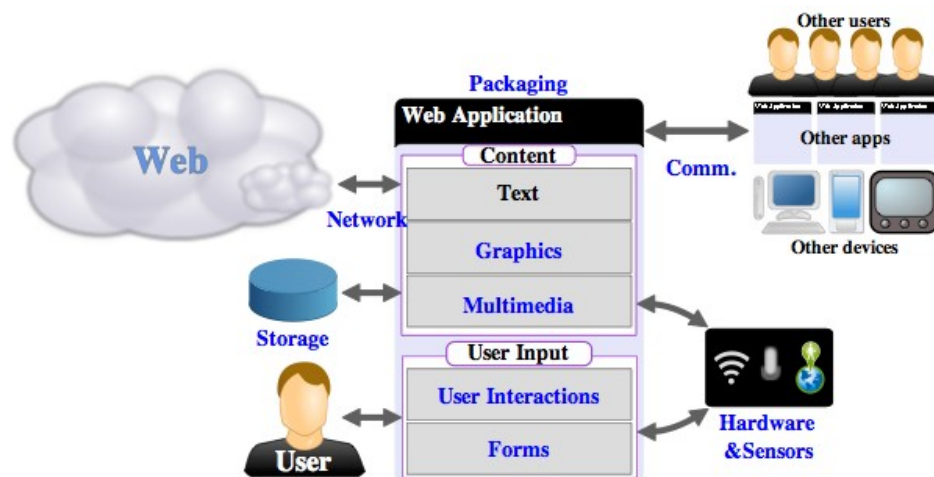
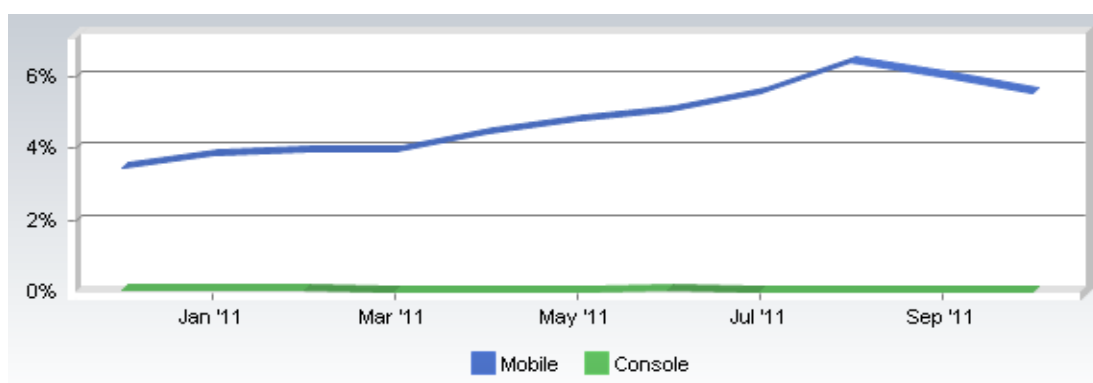


Figura 1. Il Web come una piattaforma di sviluppo di un'applicazione

2 API: acronimo di Application Programming Interface, le API sono interfacce aperte e accessibili di un software o di un sito web utilizzabili dagli sviluppatori i quali possono integrarle nei loro siti web o programmi.

Il MWI mette a disposizione anche strumenti di validazione del codice, come il W3C mobileOK Checker che determina quanto una pagina sia mobile-friendly.



Mese	Desktop	Mobile + Tablet	Console
Dicembre, 2010	96.29%	3.44%	0.04%
Gennaio, 2011	95.91%	3.79%	0.03%
Febbraio, 2011	95.81%	3.88%	0.03%
Marzo, 2011	95.80%	3.92%	0.02%
Aprile, 2011	95.26%	4.43%	0.03%
Maggio, 2011	94.92%	4.79%	0.03%
Giugno, 2011	94.62%	5.02%	0.03%
Luglio, 2011	94.13%	5.50%	0.02%
Agosto, 2011	93.20%	6.37%	0.02%
Settembre, 2011	93.65%	5.98%	0.02%
Ottobre, 2011	94.16%	5.52%	0.02%

Tabella 1. Trend della navigazione da dispositivo (Da Dicembre 2010 a Ottobre 2011)

## 1.2 Perché sviluppare per iPhone

Il fenomeno del Mobile Web, in crescita, è stato sicuramente coadiuvato dalla nascita di dispositivi mobili che non sono stati creati solo per essere dei comuni telefoni cellulari.

Infatti questi strumenti sono dotati di applicazioni che aggiungono funzionalità al telefono, permettendo a chi ne possiede uno, di organizzare, gestire e avere

informazioni in modalità diverse da un normale computer.

Ultimamente i colossi di Mountain View e non solo, sono in lotta tra di loro per la conquista dei mercati, dando così vita a una “guerra” simile a quella dei browser degli anni Novanta: *la guerra dei brevetti*. Brevettare i propri software, così da indurre un potenziale cliente a scegliere un brand piuttosto che un altro e mettere fuori gioco la concorrenza.

Sempre da netmarketshare.com i dati statistici mostrano che i principali attori di questo scontro sono proprio quelli che possiedono maggiori quote di mercato.

I sistemi operativi maggiormente utilizzati sono iOS della Apple, installato su iPhone, iPad, iPod Touch e Android prodotto da Google, seguendo a ruota Java ME, Symbian installato su dispositivi Nokia e BlackBerry. Questi dati sono un prospetto relativo al 10/2011. “v. fig. 2”

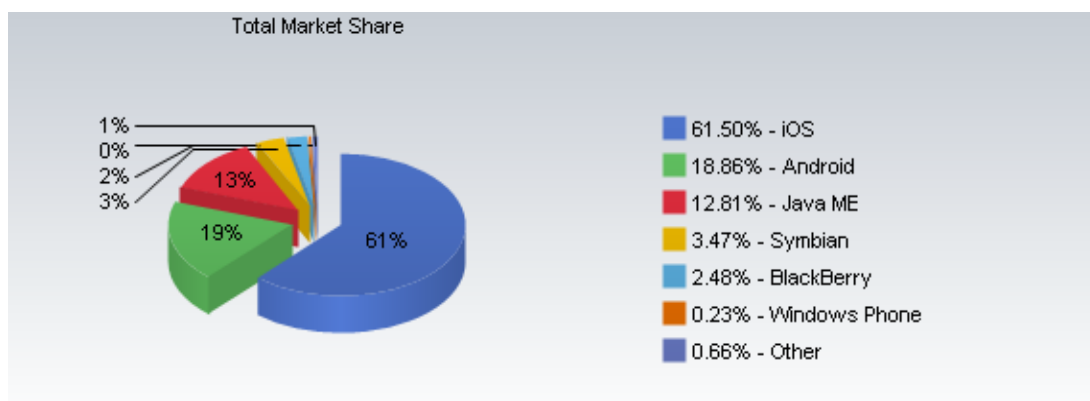


Figura 2. Quota di mercato dei sistemi operativi di dispositivi mobili/tablet

Considerando lo stesso periodo temporale (Ottobre 2011) i browser più utilizzati per accedere al web sono: Safari in testa con il 62%, Android Browser al 18% circa e Opera Mini, 13%. “v. fig. 3”.

Dal momento che il dispositivo mobile più usato per accedere al web è l'iPhone, è stato scelto quest'ultimo come piattaforma per sviluppare l'applicazione per verbalizzare esami.

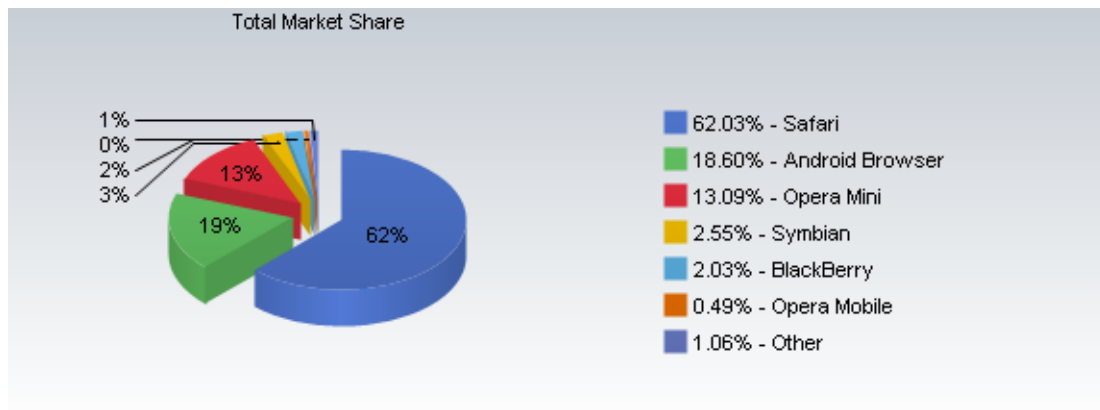


Figura 3. Quota di mercato dei browser mobili e tablet

### 1.3 App o Web App?

Ci sono due tipologie di applicazioni che possono essere create per il melafonino e occorre analizzarne le differenze per poter capire la scelta di una piuttosto che dell'altra.

Una web app è un sito web ottimizzato per browser mobili e le tecnologie utilizzate per realizzarla sono quelle del web: HTML5, CSS3, Javascript.

Le caratteristiche<sup>3</sup> che la distinguono da un'applicazione nativa sono:

1. non occorre conoscere Objective-c per poterla sviluppare, dal momento che si realizza con le tecnologie web sopracitate;
2. è raggiungibile ad una URL, quindi, per funzionare, il dispositivo mobile deve essere connesso ad Internet;
3. viene visualizzata all'interno del browser;
4. non ha accesso all'hardware del telefono cellulare;
5. non deve sottostare alle regole di un'app per l'approvazione dell'Apple Store, in quanto per poterla fruire basta solo raggiungere l'URL;
6. è gratuita;
7. l'utente non deve ogni volta scaricare gli aggiornamenti in quanto la web app è a tutti gli effetti un sito web.

Un'applicazione nativa invece è un piccolo software applicativo, realizzato attraverso un linguaggio proprietario come Objective-C. Comunica con l'hardware e il software del dispositivo e le caratteristiche che la differenziano da una web app sono:

1. non è raggiungibile ad una URL perché si installa sul dispositivo mobile;

<sup>3</sup> Stark, Jonathan 2010. Building iPhone Apps. United States of America, O' Reilly Media

2. non è necessaria la connessione ad Internet perché può funzionare anche offline;
3. non viene visualizzata dentro il browser e può accedere all'hardware del cellulare;
4. deve essere approvata dall'Apple Store affinché l'utente possa fruirne;
5. può essere gratuita o avere un costo per l'utente finale, il quale, per averla la può scaricare o acquistare tramite iTunes Store.

Quindi sostanzialmente, la scelta di creare un'app o una web app può essere fatta in base all'utilizzo che se ne deve fare. Se si ha la necessità di comunicare con l'hardware del telefono allora la scelta più indicata è sicuramente l'applicazione nativa, altrimenti può essere creata una web app.

Per il tipo di progetto realizzato si sarebbe potuto optare sia per una web app che per una app.

Una web app amplia l'accessibilità a più dispositivi ma in un'attenta analisi è emerso che gli strumenti di sviluppo sono ancora in evoluzione e non tutti i browser mobile riescono a gestire le specifiche rilasciate dal MWI.

Anche lo stesso MWI, pur lavorando agli standard per siti web mobile-friendly, presenta molte lacune nello sviluppo delle tecnologie elencate nelle specifiche<sup>4</sup>.

Quindi ciò che emerge è un'accessibilità limitata ai dispositivi più venduti e più usati dagli utenti e le caratteristiche di gestione della comunicazione con l'hardware vengono meno. Anche la memorizzazione dei dati avviene all'interno del browser mobile, quindi potrebbe far insorgere problemi di sicurezza, perché viene utilizzato da altri siti web che potenzialmente potrebbero avere accesso a tali dati.

Un'applicazione nativa, essendo sviluppata per una piattaforma ben precisa riduce l'accessibilità dell'app stessa a un unico dispositivo. Dal momento però che Gradeup è stata sviluppata con un compilatore multi-piattaforma, questo ostacolo è stato in parte aggirato e in più si è potuto accedere, con una maggiore garanzia di supporto, all'hardware del dispositivo. Per quanto riguarda invece la memorizzazione dei dati, questi rimangono in locale sul telefono; i dati infatti non sono disponibili per altre applicazioni né tanto meno per il browser. Quindi molto più gestibile il problema della sicurezza.

---

4 Mobile Web Initiative, <http://www.w3.org/Mobile/>



Progettare un'applicazione che si interfaccia con il sistema Servizio Statini, ha posto un altro dubbio sulla scelta. Una delle differenze tra app e web app è che la prima può funzionare offline e la seconda no. Dal momento che c'è necessità di recuperare i dati dal gestionale universitario via web, sia l'una che l'altra avrebbero potuto funzionare.

Alla fine il confronto è ricaduto sulla gestione tra interfaccia e hardware e OS ed è stata scelta l'applicazione nativa perché più funzionale e più sicura nella gestione dei dati.

## **1.4 Servizio Statini**

Servizio Statini è un'applicazione web per la verbalizzazione degli esami online dell'Università di Pisa.

È un servizio che velocizza l'inserimento degli esami nel gestionale universitario, si sostituisce alla compilazione degli statini cartacei. In questo modo gli studenti non devono aspettare settimane o mesi per vedersi gli esami riconosciuti.

Dal momento che risiede a livello fisico ad un indirizzo ip è raggiungibile da chiunque ma con un elaborato sistema di sicurezza, attraverso un doppio login, in realtà è accessibile solo ai docenti che hanno abilitato il proprio numero di cellulare al servizio.

Il doppio login consiste in:

1. inserimento del codice docente e della password;
2. se il primo login va a buon fine, si presenta una nuova interfaccia in cui viene richiesta una ulteriore password. Quest'ultima è recapitata tramite sms al numero di cellulare del docente abilitato. A quel punto il docente deve copiare e inserire la nuova password dove richiesta.

Una volta superata la fase di accesso, attraverso un settaggio di cookie, viene avviata una nuova sessione, con un tempo limitato e quindi soggetta a scadenza. Per cui un docente potrebbe non riuscire ad inserire oltre un certo numero di esami. Quando la sessione scade l'utente deve ripetere la fase del login.

Servizio Statini offre altri servizi al docente:

- ricerca: con la quale è possibile cercare esami nei verbali già registrati nel sistema;

- annullamento esami: se un docente ha sbagliato a registrare l'esame può scegliere di non confermarlo e quindi annullarlo;
- statistiche: fornisce un'analisi degli esami inseriti.

## 1.5 Gradeup

*Gradeup* è l'applicazione nativa progettata per interfacciarsi al sistema di registrazione, al quale anche Servizio Statini si affaccia, e renderlo così accessibile al dispositivo mobile.

Il nome nasce da un gioco di parole e dal suono di come queste si pronunciano. *Grade* è il nome dell'applicazione e significa dare un voto, mentre *up* gioca a livello fonologico con *app*, rafforzando il significato di *to grade* e porta in sé anche le iniziali di Università di Pisa (UP).

Progettata per iPhone e quindi per un dispositivo mobile, Gradeup ha funzionalità ridotte rispetto al Servizio Statini, non fornisce la possibilità della gestione statistica degli esami.

Gradeup fornisce quindi il servizio di verbalizzazione e annullamento degli esami.

È stato mantenuto il doppio login di Servizio Statini con alcune differenze, che saranno analizzate successivamente, nella gestione della one-time-password.

Non è stata creata la stessa interfaccia perché lo strumento di fruizione è diverso a partire dallo schermo di 3.5" per arrivare alla tecnologia multi-touch che cambia il modo di interazione tra utente-interfaccia e complica le azioni da effettuare.

Sono state progettate di conseguenza, interfacce più semplici e usabili che limitano i "tocchi" dello schermo, i quali si sostituiscono ai classici click del mouse.

Nello scenario del Mobile Web e delle applicazioni, alcuni sviluppatori hanno ideato dei framework per creare app e web app attraverso tecnologie web.

Tra i più utilizzati ci sono jQuery Mobile, Sencha Touch, iWebKit per le web app e Phonegap, Appcelerator Titanium per le app.

Una web app, a sua volta, può essere trasformata in app attraverso strumenti di compilazione.

Gli strumenti per la realizzazione del progetto sono: Phonegap per compilare in applicazione nativa e accedere alle funzionalità dell'hardware del dispositivo e jQuery Mobile per creare la struttura interna delle interfacce con un look and feel

iPhone.

### 1.5.1 Stato dell'arte

Esistono applicazioni native che permettono al docente di gestire esami, classi e studenti. Sono però uno strumento di gestione degli esami al fine di tenerne traccia per uso statistico, informativo, che non offre in alcun modo un servizio di verbalizzazione integrato con il sistema informativo universitario. Un'applicazione che si sostituisce al classico “statino” universitario a oggi non c'è.

Le applicazioni che sono state prese in considerazione sono:

1. **Gradebook**: aiuta i docenti a tenere traccia dei voti e dell'andamento delle classi, con possibilità di calcolare la media, porre degli obiettivi e vedere quali studenti hanno raggiunto i traguardi prefissati.
2. **TeacherPal**: anche questa permette di tenere traccia dei voti, di gestire la comunicazione coi parenti degli studenti tramite e-mail e di settare un profilo per ogni studente.
3. **TeacherTool**: più complessa di TeacherPal permette di tenere un profilo per gli studenti, memorizzando anche le loro assenze ingiustificate, i voti. Tiene traccia anche dei corsi di insegnamento, con possibilità di inserire note sull'andamento del corso stesso.

Queste applicazioni sono un'estensione del classico registro di classe. Si sostituiscono ai fogli di calcolo e ad altri strumenti di gestione dei dati, delle note.

*Gradeup* invece è molto diversa, è stata studiata appositamente per il sistema telematico di registrazione degli esami dell'Università di Pisa e si potrebbe estendere alle altre università solo a patto di rivedere le interfacce con il sistema server.

Di fatto è un'interfaccia alternativa a quella esistente ma usabile e accessibile ai dispositivi mobili. Una verbalizzazione elettronica che sostituisce le matrici stampate.

## 2. Fasi della progettazione

### 2.1 jQuery Mobile per lo stile e la struttura

jQuery Mobile è un potente framework open source in versione beta, scritto in Javascript che si appoggia alla libreria jQuery per funzionare.

La filosofia che abbraccia questo framework è quella del “*write less do more!*”: invece di scrivere un'applicazione per ogni dispositivo mobile in circolazione, questo strumento permette di scrivere una web app che sia fruibile dalle più popolari piattaforme di smartphone e tablet<sup>5</sup>.

jQuery Mobile sfrutta le potenzialità di HTML5, CSS3 e Javascript per creare animazioni, contenuti dinamici, tutto tramite la manipolazione del DOM.

Per ridurre il peso delle pagine, jQuery Mobile usa le novità introdotte nei CSS3 come gli angoli arrotondati, le ombre sui box o sul testo, senza dover utilizzare immagini.

Il framework usa il CSS per controllare il contenuto visivo della web app. Questo file si suddivide in due parti:

- temi: i quali controllano la parte visiva degli elementi del DOM, come i font, i colori dei background, etc... ;
- struttura: la quale controlla le posizioni degli elementi, i margini, i pulsanti, etc... .

Ogni tema può includere uno o più *swatches*. Gli swatches sono schemi che contengono differenti proprietà di colori di un tema e di default sono cinque.

Possono essere cambiati all'interno della pagina in modo molto semplice. Sono ordinati secondo le lettere dell'alfabeto: *a, b, c, d, e*, e per convenzione lo schema *a* è quello che ha il livello più alto di priorità. jQuery Mobile gestisce il livello di priorità degli swatches se non viene specificato uno schema alternativo.

Per il progetto sono stati creati tre schemi di colore alternativi: *a, c, d*.

Sono state modificate anche proprietà a livello strutturale del CSS, per la personalizzazione dell'app.

jQuery Mobile, in versione 1.0rc1, è stato scelto per Gradeup, in quanto permette di confezionare pagine in modo veloce, mantenendo tutte le caratteristiche di un'app.

---

<sup>5</sup> jQuery Mobile, [www.jquerymobile.com/](http://www.jquerymobile.com/)

## 2.2 Phonegap per accedere all'hardware

Phonegap, sviluppato dalla Nitobi, è un compilatore di applicazioni in HTML5, open source. A differenza di jQuery Mobile, Phonegap permette di scrivere l'intera applicazione in HTML5, CSS3, Javascript e di compilarla in applicazione nativa.

Un'altra differenza che intercorre tra i due strumenti è che Phonegap consente di accedere all'accelerometro, al file system, alla fotocamera, e al GPS e anche di pubblicare nell'App Store.

Phonegap non produce delle vere e proprie applicazioni native ma interfaccia l'applicazione web a iOS utilizzando delle *webview* che non sono altro che richiami del browser Safari all'interno dell'app.

iOS dispone di un framework che si chiama Webkit, il quale fornisce classi da implementare per includere del contenuto web all'interno di un'app<sup>6</sup>. Una di queste è Webview. Phonegap, di conseguenza, è un ulteriore livello interpretativo di codice che ingloba Webkit il quale si interfaccia a iOS."v. fig. 4."

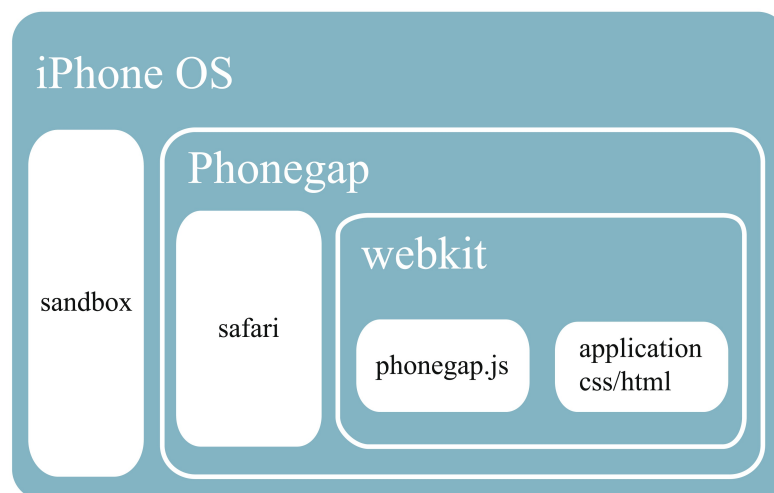


Figura 4. Phonegap

Un punto di forza di Phonegap è lo sviluppo per più piattaforme come iOS, Android, Palm, BlackBerry, Symbian, WebOS e si contestualizza nel "*write once, run anywhere*" con Phonegap Build, per compilare le app per iniziare ad utilizzarle e distribuirle.

Altre caratteristiche che lo contraddistinguono sono la possibilità di creare dei plugin per implementare le funzionalità con il dispositivo mobile, una community di

<sup>6</sup> Apple, "WebKit Objective-C Programming Guide". <http://developer.apple.com>

supporto agli sviluppatori e molti altri strumenti di sviluppo.

Infine questo strumento mette a disposizione per l'accesso all'hardware dodici API tutte supportate da iPhone: Accelerometer, Camera, Capture, Compass, Connection, Contacts, Device, Events, File, Geolocation, Media, Notification, Storage<sup>7</sup>.

È stato scelto Phonegap nella versione 1.1.0, per la compilazione dell'app per la sua efficienza e anche perché l'App Store ne approva le applicazioni native.

## 2.3 Progettazione grafica

Per quanto riguarda l'aspetto grafico, il punto di riferimento è stato il sito web dell'Università di Pisa nella sua veste grafica rinnovata. Questa scelta è dovuta al fatto che, essendo Gradeup progettata per essere uno strumento dell'università, si è voluto mantenere un'identità ben precisa a partire dal cherubino e dal tipico colore blu che sono segni distintivi. Allo stesso tempo, però, è stato creato un logo nuovo, che identificasse l'applicazione e la potesse distinguere dalle altre possibili dell'Università. “v. fig. 5”.

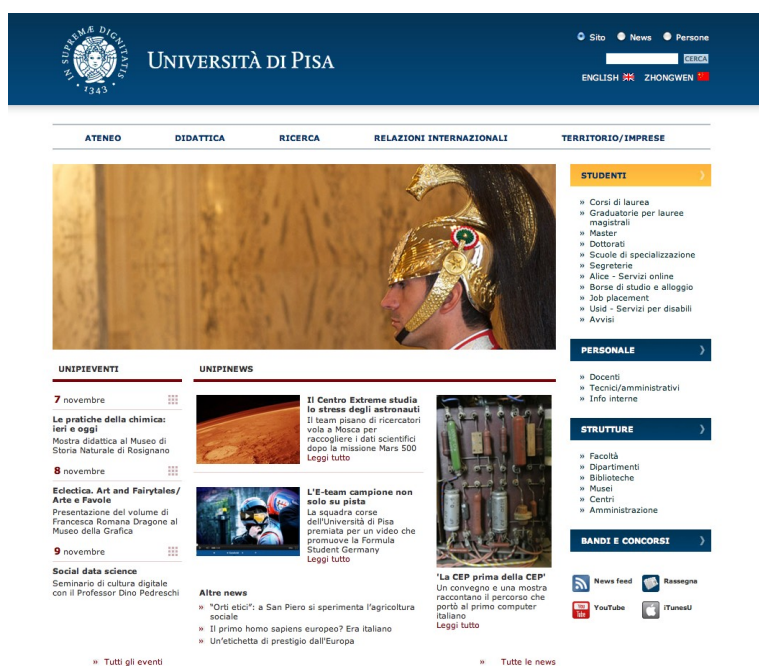


Figura 5. Sito web ufficiale dell'Università degli Studi di Pisa.

### 2.3.1 Launch image e icone

Una app non può essere pubblicata nell'App Store se è sprovvista di icone e *launch*

<sup>7</sup> Phonegap, <http://docs.phonegap.com>

*image* o *splashscreen*. La Apple fornisce delle linee guida per la loro realizzazione<sup>8</sup>.

Le icone obbligatorie sono:

1. icona applicazione;
2. icona per App Store;

L'icona applicazione è quella che viene installata sulla home screen dell'iPhone. Deve essere riconoscibile tra tutte le altre icone, con un design pulito e d'impatto.

Per la progettazione di un'app universale, l'icona deve essere creata in tre dimensioni diverse e avere un nome specifico, che poi il sistema operativo iOS assocerà in base al dispositivo utilizzato. “v. tab. 2”.

<b>Descrizione</b>	<b>Dimensioni per iPhone e iPod touch (in pixel)</b>	<b>Nome</b>
Icona Applicazione (richiesta)	57 x 57 114 x 114 (alta risoluzione)	icon.png icon@2x.png
Icona per App Store (richiesta)	512x512	
Launch Image (richiesta)	320x480 640 x 960 (high resolution)	Default.png Default@2x.png

Tabella 2. Icone e launch image

Il formato dell'immagine deve essere PNG e iOS provvederà ad aggiungere gli angoli arrotondati, l'ombra e l'effetto riflesso. Quest'ultimo può essere modificato nel caso in cui ci sia la necessità di aggiungere un riflesso diverso da quello di default.

L'icona per l'App Store è quella che identifica l'app in iTunes, e può essere anche diversa dall'icona dell'applicazione, può essere più dettagliata. L'importante è che sia identificativa dell'app e sia facilmente riconoscibile.

Infine la launch image è la prima immagine che viene mostrata quando l'app viene avviata e dovrebbe essere molto simile alla prima pagina. La sua funzione è quella di ingannare l'attesa. “v. fig. 6”.

---

<sup>8</sup> Apple Developer, "iOS Human Interface Guidelines", [www.developer.apple.com/](http://www.developer.apple.com/)



Figura 6. Icona Applicazione, Icona App Store, Launch Image

### 2.3.2 Le Interfacce

Tenendo presente sempre l'utenza finale alla quale è rivolta l'applicazione, cioè la docenza, è stato scelto un modello grafico pulito e semplice.

Dall'icona applicazione alle interfacce il motivo ricorrente è il logo o parte di esso.

Il logo completo che viene visualizzato nello splashscreen e nella prima pagina ha caratteristiche come ombre e spigoli che vengono poi riportate nelle interfacce.

Ogni singola interfaccia si compone di un header, di un corpo e di un footer.

L'header, nelle applicazioni native sviluppate per iPhone, è una barra che riporta il titolo della pagina e può avere le funzionalità di una barra di navigazione. In Gradeup contiene la coccarda del logo in contrasto e il titolo della pagina. Nel corpo, invece, vengono utilizzati dei campi input che richiamano le caratteristiche del logo come le ombreggiature e gli spigoli. Infine il footer riporta la parte del logo rimossa che accompagna il cherubino: "Università degli Studi di Pisa". "v. fig. 7."





Figura 7. Interfaccia Materia, Gradeup

## 2.4 Implementazione degli strumenti e gestione delle interfacce

Tenendo sempre presente il sistema di gestione dei dati di Servizio Statini, è stata creata una struttura che potesse emularne i servizi al fine di testare l'applicazione.

### 2.4.1 Login

Come già descritto in precedenza, Servizio Statini, per garantire la sicurezza dell'accesso a tale sistema, si basa su un doppio login:

1. il primo è un inserimento del codice docente e password;
2. il secondo login avviene con l'inserimento di password inviata tramite sms sul numero di cellulare del docente;

Per Gradeup sono state pensate alcune alternative per rendere automatico il processo di login saltando il secondo passaggio per il docente, dal momento che l'applicazione gira su un dispositivo mobile.

Un'alternativa è stata quella di rilevare il numero di cellulare dal dispositivo attraverso Objective-C. Non è stata una soluzione valida in quanto iOS, per motivi di sicurezza, impedisce di accedere a questo tipo di informazioni per evitare lo SPAM da parte di sviluppatori che vogliono sponsorizzare applicazioni, ma non solo.

Inoltre l'unico modo per ottenere il numero di telefono è accedere alle informazioni di iTunes ma lo stesso iTunes non aggiorna il numero in caso di cambio di *SIM card* contenente un'altra utenza telefonica. La possibilità di non ottenere il numero corretto ha reso questa soluzione non idonea.

Un'altra soluzione è stata quella di mantenere il sistema di invio sms, cambiandone il contenuto: invece di ricevere come testo una password, è stato scelto di inserire un link all'applicazione.

Il link è composto da:

- la url dell'applicazione registrata nell'*info.plist*, in questo caso ***gradeup://*** ;
- la query string che contiene la coppia chiave-valore da gestire, ***gradeup://?psw="one-time-password"***.

Nell'iPhone SDK c'è la possibilità di registrare gli url-schema per lanciare applicazioni dal browser Safari o da altre applicazioni, per manipolare dati o gestire la comunicazione con altre app come per esempio quella degli sms.

È stata scelta questa soluzione perché permette di mantenere la sicurezza della ricezione dell'sms sul numero di telefono del docente ma anche perché elimina il passaggio di digitazione della one-time-password la quale viene gestita automaticamente grazie al link.

Il Servizio Statini, legato all'sms, ha anche un sistema di scadenza dei cookies, i quali vengono memorizzati nel browser e avviano una sessione di registrazione degli esami.

Gradeup è un'applicazione nativa e anche se Phonegap richiama all'interno dell'app il browser, iOS non permette di recuperare e memorizzare cookies come una normale web app.

L'applicazione non ha accesso ai cookies settati nel browser ma ha accesso ad un repository nel quale questi possono essere memorizzati e recuperati tramite delle modifiche fatte nell'*appDelegate.m* .

È stato possibile recuperare i cookies e inserirli nel repository della sandbox dell'app

che iOS mette a disposizione attraverso questo codice:

```
NSHTTPCookieStorage *cookieStorage = [NSHTTPCookieStorage
                                     sharedHTTPCookieStorage];
[cookieStorage
 setCookieAcceptPolicy:NSHTTPCookieAcceptPolicyAlways];9
```

## 2.4.2 Le interfacce Materia, Esame e Database

Dopo che l'accesso al sistema di verbalizzazione degli esami è andato a buon fine, nel Servizio Statini si avvia la sessione nella quale è possibile registrare più esami alla volta finché la sessione non scade.

Per questa fase è stata tenuta in considerazione la differenza che intercorre tra una tastiera e uno schermo multi-touch e inevitabilmente anche la differenza che c'è tra uno schermo di un computer e uno schermo di un iPhone.

È stato necessario ridurre il numero di interfacce per non rendere troppo complesse la compilazione dei campi e la *"user experience"*.

Le pagine sono state create con jQuery Mobile, secondo la seguente struttura:

```
<div data-role="page" data-theme="a" id="id-pagina">
  <div data-role="header" class="head">
    <div>
      
      <h1>TITOLO PAGINA</h1>
    </div>
  </div>
  <div data-role="content">
    <!-- contenuto della pagina -->
  </div>
  <div data-role="footer" data-position="fixed"
  id="footer">Università degli Studi di Pisa</div>
</div>
```

Una delle novità di HTML5, su cui si basa jQuery Mobile, è la possibilità di aggiungere agli elementi del DOM, attributi personalizzati. Infatti da codice, è visibile una struttura pulita e chiara anche nella lettura.

Ogni pagina .html può contenere una singola interfaccia oppure, sfruttando una funzionalità di jQuery Mobile, più pagine e quindi più interfacce. Nel `<body>` basta aggiungere la struttura descritta sopra per quante volte è necessario. Ci penserà il

---

<sup>9</sup> Google Group, <http://groups.google.com/>

framework a gestire la comunicazione tra le pagine, aggiungendo transizioni personalizzabili, nascondendo blocchi, etc... , tutto tramite una comunicazione basata su ajax.

Ogni pagina creata è composta da un contenitore al quale è possibile associare un tema:

```
<div data-role="page" data-theme="a">
```

all'interno del contenitore `page` ci sono altri 3 contenitori:

```
<div data-role="header"></div>
```

```
<div data-role="content"></div>
```

```
<div data-role="footer"></div>
```

Questi definiscono la struttura visiva della pagina. L'`header` contiene il logo e il titolo dell'interfaccia, il `content` invece il corpo della pagina e infine il `footer` la barra che visualizza la scritta: "Università degli Studi di Pisa".

La pagina `Materia` è una multi-page. Si suddivide in:

1. **materia**: interfaccia per l'inserimento dei dati come il codice della materia, il membro della commissione e la data dell'esame;
2. **esame**: riepilogo dei dati immessi nella pagina `Materia` e infine campi input per l'inserimento della matricola, nome e cognome dello studente e il voto dell'esame.

`Materia` del punto 1 ha delle relazioni di comunicazione con altre pagine al suo interno. Per esempio, per l'inserimento del codice della materia, è possibile cliccare sul pulsante "matita" per avere la lista di tutte le materie legate al codice docente. La lista è a sua volta una ulteriore interfaccia e le sue voci sono a loro volta dei pulsanti che rimandano alla pagina `Materia`. Così vale per la scelta del membro della commissione e per la data. In sostanza non è una struttura che va in profondità ma una struttura che rimanda sempre alla pagina principale. "v. fig. 8 e fig. 9"

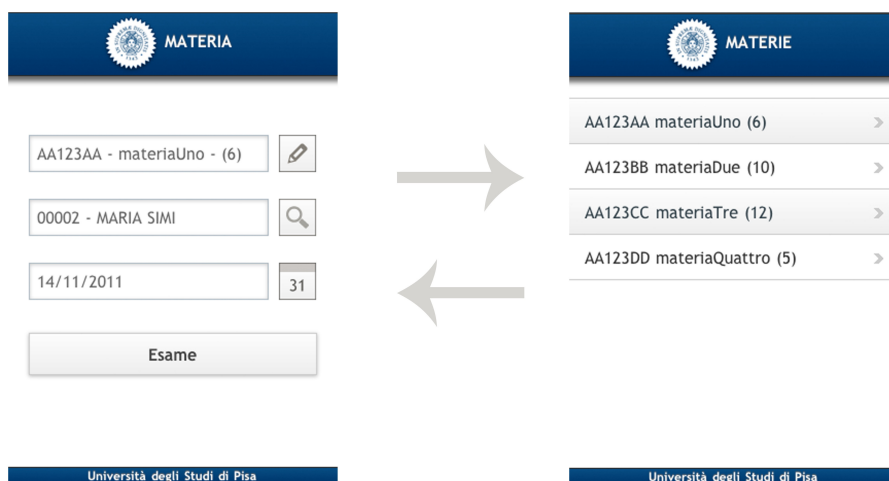


Figura 8. Comunicazione tra interfacce Materia - Materie

Dal momento che un docente può trovarsi a registrare più di un esame alla volta, in questo modo sono state eliminate molte azioni che sarebbero pesate sull'utente e avrebbero aumentato il tempo di compilazione dei moduli per la verbalizzazione.

Per quanto riguarda Esame invece l'approccio è molto diverso in quanto è in parte il risultato di scelte fatte nei precedenti passi.

Si presenta come una scheda riassuntiva dei dati inseriti nell'interfaccia Materia e in più c'è la possibilità di inserire la matricola, il nome e cognome dello studente (questi ultimi due vengono inseriti automaticamente) e infine il voto dell'esame.



Figura 9. Comunicazione tra interfacce Materia - Calendario

In questo caso non ci sono rimandi ad altre interfacce e per concludere la verbalizzazione basta premere sul pulsante di registrazione.

Il docente in una sessione esami avrà più di uno studente al quale verbalizzare lo stesso esame, con Gradeup c'è la possibilità di mantenere la parte dei dati inseriti in Materia così da facilitare l'inserimento di più esami in modo pratico e veloce.

Tutto questo è possibile grazie all'uso di API come `localStorage` e `sessionStorage`, del MWI che Phonegap utilizza per la memorizzazione e persistenza dei dati.

Nel momento in cui devono essere cambiati i dati di Materia, ad ogni registrazione dell'esame c'è la possibilità di scegliere se continuare a verbalizzare con i dati di Materia immessi o se cambiarli.

Per quanto riguarda invece l'interfaccia Database, una volta verbalizzato l'esame nel gestionale universitario, l'esame viene memorizzato anche nel database locale dell'applicazione. Questo per dare la possibilità al docente di compiere eventuali modifiche in caso di errata digitazione dei dati, per poi successivamente confermare e infine aggiornare la registrazione nel gestionale universitario.

### **2.4.3 Comunicazione con il Server**

Gradeup deve reperire i dati da una fonte esterna al dispositivo mobile, come le matricole, codici delle materie e così via. Questa fonte esterna è il server che fornisce i dati a Servizio Statini.

La Cross Domain Policy è stata un primo ostacolo per il reperimento. I dati necessari al funzionamento infatti risiedono ad un URL diverso da quello in cui risiedono i file dell'app. Quindi non è stato possibile utilizzare l'oggetto `XMLHttpRequest` per lo scambio di dati con Ajax.

La libreria jQuery che fa funzionare jQuery Mobile però ha messo a disposizione delle API per la gestione di una comunicazione Cross-Domain.

Per Gradeup è stato scelto un plugin scritto per jQuery<sup>10</sup> e la comunicazione con il server avviene tramite lo scambio di dati in formato JSONP.

Un esempio:

```
$.jsonp({  
  url:"http://saradari.interfree.it/materie.json",
```

---

<sup>10</sup> jquery-json: "JSON plugin for jQuery, provides simple ways to convert to JSON and back again"  
<http://code.google.com/p/jquery-json/>

```
callback:"codici",
success: function(data,status){
    $.each(data,function(i,value){
        cod=value.codice;
        mat=value.materia;
        $("<option value="+cod+">"+cod+"</option>").appendTo("#materie");
    });
}
```

È possibile usare anche oggetti JSONP e specificare la funzione di callback, cioè il padding che contiene l'oggetto JSON, elaborare la risposta, ottenere lo status della richiesta, gestire gli errori della chiamata e così via.

## 3. Compilare l'app

### 3.1 Strumenti di compilazione

Nello scenario del web prestatato alle applicazioni native, ci sono molti strumenti a disposizione degli sviluppatori, in grado di compilare applicazioni, di convertire web app in app.

Ci sono strumenti che creano app per più piattaforme, altri solo per iOS.

Gradeup è stata progettata per iPhone usando tecnologie web e di conseguenza non è stata necessaria una compilazione per più piattaforme, quindi lo strumento usato è Xcode con sviluppo per Device.

#### 3.1.1 Phoneygap Build

Phoneygap Build<sup>11</sup> è uno strumento per la compilazione di applicazioni cross-platform, realizzate in HTML, CSS, Javascript.

Phoneygap non solo permette di testare l'applicazione sul simulatore ma dal sito web è possibile compilare l'app per la distribuzione.

Phoneygap Build è un tool gratuito, in versione beta, per progetti open-source, mentre per compilare applicazioni private c'è la possibilità di scegliere tra varie licenze con delle restrizioni sulla quantità di app da creare.

La parte interessante di questo strumento è la compilazione per più piattaforme, mantenendo così il *"write once, run anywhere!"*. A oggi i sistemi operativi supportati sono: iOS, Android, BlackBerry, WebOS, Symbian.

Il primo passo da fare è la creazione di un account sul sito web di Phoneygap Build.

Dopo aver effettuato l'accesso, viene richiesto di caricare il file del progetto in formato zip.

Per quanto riguarda la compilazione dell'applicazione per iPhone, se non verranno forniti il certificato e il provisioning profile l'operazione non andrà a buon fine.

Automaticamente Phoneygap Build compilerà l'app anche per Android, Blackberry, WebOS e Symbian.

A fine operazione saranno fruibili le app da inviare all'App Store, e le altre da distribuire secondo i canali di distribuzione dei vari brand.

---

<sup>11</sup> Phoneygap Build, [www.phoneygap.com](http://www.phoneygap.com)



### **3.1.2 Adobe Dreamweaver**

Un'alternativa a Xcode e Phonegap Build è sicuramente Adobe Dreamweaver 5.5. Nell'ultimo anno questo software shareware ha integrato la libreria jQuery Mobile e Phonegap.

Nella pagina di avvio del programma c'è la possibilità di scegliere se creare applicazioni web o applicazioni native per dispositivi mobili come Android e iOS<sup>12</sup>.

---

<sup>12</sup> Adobe Dreamweaver 5.5, [www.adobe.com/it/products/dreamweaver.html](http://www.adobe.com/it/products/dreamweaver.html)

## 4. Distribuire l'app: Apple Store

Una volta compilata l'applicazione con Phonegap o con i metodi descritti nei capitoli precedenti, è pronta per essere inviata all'Apple Store.

Gradeup non è stata inviata allo store perché è in fase di sviluppo ma grazie agli strumenti utilizzati potrebbe essere un passo successivo da fare.

La Apple, per tutelare gli sviluppatori e soprattutto gli utenti finali, si riserva il diritto di accettare o rifiutare allo sviluppatore la pubblicazione dell'app e quindi la distribuzione di essa.

Non ci sono regole precise per assicurare la buona riuscita di questa fase o almeno alcune di queste non sono "*descritte nei termini e condizioni di utilizzo dell'SDK*<sup>13</sup>". Ci sono, però, online molte testimonianze di sviluppatori che possono risultare utili al fine di eliminare potenziali rischi per la non accettazione.

Brian Stormont è uno di questi, sviluppatore con 45 applicazioni pubblicate con successo, che ha deciso di condividere la sua esperienza.

Nel suo articolo<sup>14</sup> elenca almeno 11 possibili elementi che potrebbero far rifiutare la distribuzione.

Nella lista, Stormont sostiene che alcune cose da non fare riguardano ciò che viene descritto nelle *Human Interface Guidelines* distribuite da Apple come:

- assicurarsi di aver creato tutte le icone richieste come l'icona Applicazione e l'icona App Store;
- splashscreen non adeguato;
- uso di icone della toolbar con un diverso significato da quello a cui sono chiamati a rispondere.

---

13 Tratto da "Come evitare il rifiuto di un'applicazione per iPhone. Alcuni consigli pratici" [www.melablog.it](http://www.melablog.it)

14 Brian Stormont. "Avoiding iPhone App Rejection from Apple" [www.mobileorchard.com](http://www.mobileorchard.com)

## 5. Conclusioni

### 5.1 App: Write once, Run Anywhere?

Modellare applicazioni native attraverso tecnologie web non è semplice. Gli strumenti utilizzati per realizzare Gradeup sono tutti in fase beta, quindi in continuo aggiornamento. Il supporto informativo che affianca questi framework, non sempre è soddisfacente, anche perché sono strumenti in via di sviluppo e in continua evoluzione.

La complessità sta nel fatto di poter realizzare un'app con linguaggi del web e allo stesso tempo riuscire a ottenere quella interoperabilità e accessibilità a qualsiasi dispositivo mobile. C'è sempre una scelta da fare: per esempio creare un'applicazione nativa con un look and feel per iOS e compilarla anche per Android potrebbe generare confusione nell'utente che ha una certa esperienza con un dispositivo piuttosto che con un altro.

Un primo caso di differenza appena descritta da poter analizzare è la seguente:

- *Prontuario Farmaceutico*<sup>15</sup> è sviluppata solo per Android, senza il framework Phonegap, "v. fig. 10, immagine a sinistra";
- *AteneoWeb*<sup>16</sup> è disponibile solo per Android ed è sviluppata con Phonegap. "v. fig. 10, immagine a sinistra";

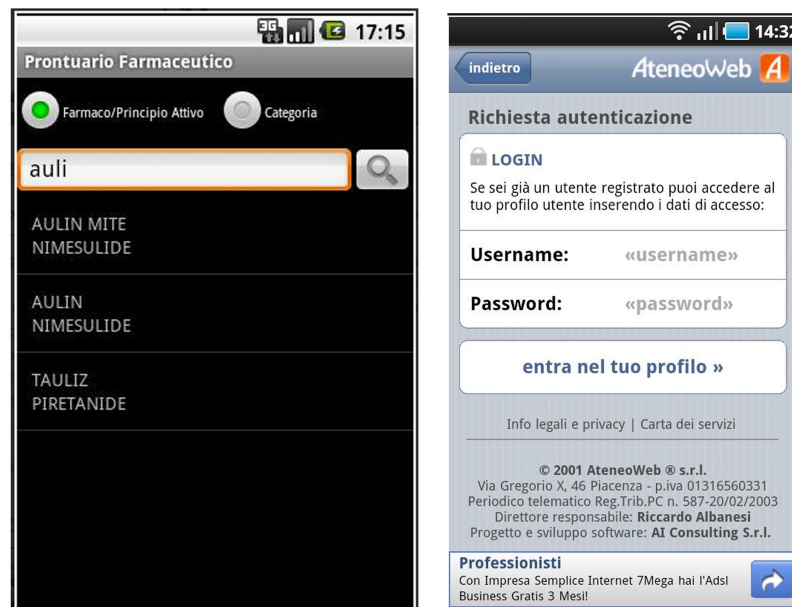


Figura 10. Due applicazioni di Android a confronto.

15 Android Market, <http://market.android.com/>

16 Phonegap Apps, <http://phonegap.com/apps>

Entrambe sono sviluppate per la stessa piattaforma ma AteneoWeb sembra a colpo d'occhio sviluppata per iOS. Lo sfondo a righe celeste è tipico delle applicazioni di utilità di iPhone, il form del login anche, per citarne alcuni.

Un altro esempio a sostegno della coerenza di sviluppo per un dispositivo è dato dal confronto di *iInterazione*<sup>17</sup> e *iProntuario*<sup>18</sup>. Create dallo stesso sviluppatore sia per iPhone che per Android senza Phonegap. "v. fig. 11"



Figura 11. Applicazioni coerenti al dispositivo.

Nell'immagine si possono vedere le due interfacce di "Info" che mostrano le caratteristiche di ciascun dispositivo. Queste schermate sono molto simili, infatti rimandano entrambe allo sviluppatore.

Già nella tabbar di tutte e due le app, sono visibili le peculiarità di visualizzazione dell'elemento sia di Android che iPhone. Colori, grandezza dei pulsanti, dimensione della navbar e così via.

L'ultimo esempio è di un'applicazione sviluppata con Phonegap per più piattaforme: Airport Car Rental<sup>19</sup>. "v. fig. 12". In questa schermata a un primo sguardo sembra prevalere un look and feel iPhone.

L'analisi però mostra come le differenze di visualizzazione di un'app siano annullate, in quanto la stessa interfaccia sarà visualizzata nello stesso modo su più dispositivi,

17 iTunes, <http://itunes.apple.com/it/app/iinterazioni/id420408868?mt=8>

18 Android Market, <http://market.android.com/>

19 Phonegap Apps, <http://phonegap.com/apps>

facendo prevalere un tipo di grafica piuttosto che un'altra.

Phonegap dal lato della comunicazione con l'hardware e con il sistema operativo funziona su più piattaforme e può essere uno strumento utile per chi vuol sfruttare la conoscenza dei linguaggi del web, senza dover imparare linguaggi proprietari o simili.

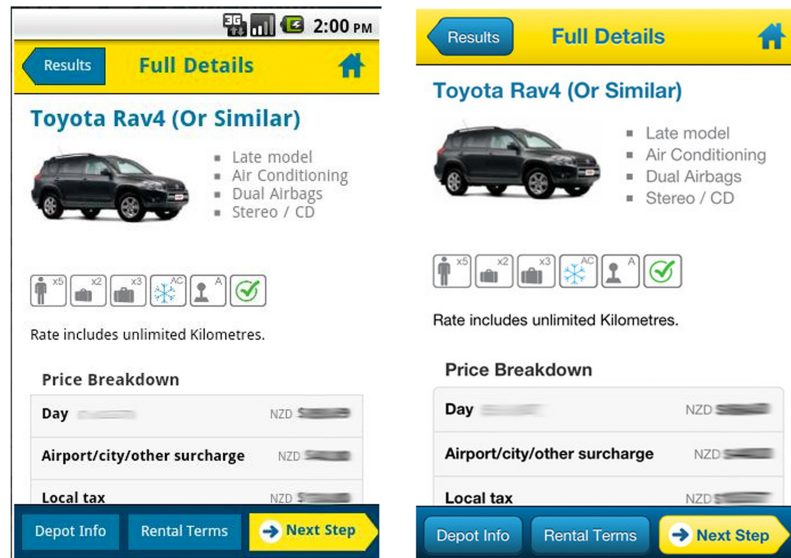


Figura 12. App multi-piattaforma

In questo modo il risparmio di tempo nella realizzazione di un'applicazione per più dispositivi è garantito.

Dall'altro lato, l'approccio alla realizzazione di un'app nativa come fosse un sito web o una web app, porta a livello stilistico la mancata personalizzazione che distingue un mobilephone rispetto ad un altro.

Va tenuto presente che è possibile creare diversi fogli di stile e diverse strutture qualora si vogliano mantenere le caratteristiche grafiche di un dispositivo piuttosto che di un altro. Questo però, di conseguenza, porta a dover gestire più file, sia a livello di progettazione grafica, sia a livello di produzione, nel momento in cui lo sviluppatore decida di aggiornare l'applicazione.

Questo aspetto aumenta i tempi di sviluppo oltre che a porsi come un limite per "write once, run anywhere" di Phonegap.

Questi strumenti mostrano quanti passi avanti siano stati fatti nel mondo del Mobile Web e nel mondo delle App e apre uno scenario molto interessante, quello delle

applicazioni native e multi-piattaforma scritte con i linguaggi del web.

Pensare a questi piccoli software con una mentalità da sviluppatore di siti web porta a compiere degli errori come già visto precedentemente.

A oggi questa interoperabilità sembra ancora lontana. Inoltre crea molta confusione riguardo i due modi di sviluppare applicazioni. Web app e app non appartengono entrambe alla sfera del web ma il fermento che c'è riguardo questi fenomeni sembra andare verso quella direzione.

Se da una parte c'è la voglia degli sviluppatori di imbattersi nel nuovo mercato del dispositivo mobile, dall'altra ci sono troppe difficoltà e troppe scelte da dover fare riguardo:

- ai dispositivi;
- ai linguaggi da utilizzare vincolati anche al dispositivo;
- la tipologia di applicazione;
- gli strumenti da usare;
- le licenze da acquistare;

che in un modo o in un altro sono legate al business e all'andamento dei mercati del mobile. Il tutto richiama la lontana *guerra dei browser*. Chi ne subì le conseguenze purtroppo furono gli utenti finali.

## 6. Bibliografia

Jonathan Stark *Building iPhone Apps*. United States of America, O' Reilly Media, 2010

### 6.1 Siti Web

Adobe Dreamweaver 5.5, voce *Help*

[http://help.adobe.com/it\\_IT/dreamweaver/cs/using/WSefff8bffc80208478c8d43312e240fe0ad-8000.html](http://help.adobe.com/it_IT/dreamweaver/cs/using/WSefff8bffc80208478c8d43312e240fe0ad-8000.html)

Android Market, voce *Applicazioni* (iProntuario)

[https://market.android.com/details?id=res.farmacidirectory&feature=search\\_result#?t=W251bGwsMSwyLDEsInJlcy5mYXJtYWNPZGlyZWV0b3J5I10](https://market.android.com/details?id=res.farmacidirectory&feature=search_result#?t=W251bGwsMSwyLDEsInJlcy5mYXJtYWNPZGlyZWV0b3J5I10)

Apple Developer, *iOS Dev Center*

"WebKit Objective-C Programming Guide"

<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/DisplayWebContent/DisplayWebContent.html>

Apple Developer, voce *iOS Dev Center*

"iOS Human Interface Guidelines"

<http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Introduction/Introduction.html>

Apple, voce *iTunes* (iInterazione)

<http://itunes.apple.com/it/app/iinterazioni/id420408868?mt=8>

Google, voce *Gruppi* (Phonegap)

[http://groups.google.com/group/phonegap/browse\\_thread/thread/3290c5ac4e05be69?fwc=1](http://groups.google.com/group/phonegap/browse_thread/thread/3290c5ac4e05be69?fwc=1)

Google Code, voce *Project Hosting*  
<http://code.google.com/p/jquery-json/>

jQuery Mobile, voce *Docs*  
<http://jquerymobile.com/demos/1.0rc3/>

jQuery Mobile, voce *Platforms*  
<http://jquerymobile.com/gbs/>

Melablog, voce *News*  
"Come evitare il rifiuto di un'applicazione per iPhone. Alcuni consigli pratici"  
<http://www.melablog.it/post/8601/come-evitare-il-rifiuto-di-unapplicazione-per-iphone>

Mobile Orchard, voce *News*  
"Avoiding iPhone App Rejection from Apple"  
<http://mobileorchard.com/avoiding-iphone-app-rejection-from-apple/>

Mobile Web Initiative, link *List of standards for Web Application on Mobile*  
<http://www.w3.org/2011/02/mobile-web-app-state.html>

Netmarketshare  
<http://www.netmarketshare.com/>

Phonegap, voce *Apps (AteneoWeb)*  
<http://phonegap.com/app/ateneoweb/>

Phonegap, voce *Apps (Airport Car Rental)*  
<http://phonegap.com/app/airport-car-rental/>

Phonegap, voce *Developers*  
<http://docs.phonegap.com>



Phonegap, voce *Phonegap Build*

<https://build.phonegap.com/>

## 7. Appendice

Application Programming Interface	AJAX
Asynchronous Javascript and XML	Api
Cascading Style Sheets	CSS
Document Object Model	DOM
Figura	Fig.
HyperText Markup Language	HTML
Mobile Web Initiative	MWI
Send Phenomenal Amounts of Mail	SMS
Short Message Service	SIM card
Subscriber Identity Module Card	SPAM
Tabella	Tab.
World Wide Web Consortium	W3C