

Progettazione e Sviluppo di un'Applicazione Mobile di Marketing ed E-Commerce

Tommaso Berlose

2016-11

Indice

0.1	Situazione esistente	4
0.1.1	App Economy	4
0.1.2	Ecosistema MyGelato	4
0.1.3	Stato dell'Arte dell'Applicazione Android	4
0.2	Specifiche di Progetto	5
0.2.1	Rifare app uguale	5
0.2.2	temi con flavors	5
0.2.3	risorse da adattare	5
0.2.4	parallelismo con iOS	5
0.2.5	multilingua, login facebook, ecc...	5
0.3	Scelte di Progetto	6
0.3.1	Strumenti di Sviluppo	6
0.3.2	Programmazione Nativa	8
0.3.3	Database	9
0.3.4	Chiamate Server	9
0.3.5	Gestore di Eventi	9
0.3.6	Notifiche Push	11
0.3.7	E-Commerce	11
0.3.8	Librerie Minori	11
0.3.9	Scelte progettuali strane e che non so come chia- mare	11
0.4	Implementazione	12
0.4.1	Shop	12
0.4.2	Carte gelato	12
0.4.3	Coupon	12
0.5	Codice	16

0.6	Ringraziamenti	17
-----	--------------------------	----

Intro

Provaaaaaaaaaaaaaa

0.1 Situazione esistente

0.1.1 App Economy

App econom, bla bla



Figura 1: App Economy

0.1.2 Ecosistema MyGelato

0.1.3 Stato dell'Arte dell'Applicazione Android

0.2 Specifiche di Progetto

0.2.1 Rifare app uguale

0.2.2 temi con flavors

0.2.3 risorse da adattare

0.2.4 parallelismo con iOS

0.2.5 multilingua, login facebook, ecc...

0.3 Scelte di Progetto

Definiti gli obiettivi progettuali dell'applicazione si sono valutati gli strumenti di sviluppo da utilizzare durante lo svolgimento della tesi: come parametri si è tenuto conto di tempistiche di aggiornamento, adeguamento alle linee guida del sistema operativo in oggetto, documentazione disponibile e modernità delle tecnologie utilizzate.

Durante la prima fase di sviluppo strutturale si è dovuto valutare la tecnologia da utilizzare in modo da tenere il passo con le ultime specifiche di Android e tenendo conto che, anche secondo gli ultimi report pubblicati da Google, rimane una forte frammentazione della distribuzione del sistema operativo, dovuta ai molteplici produttori di hardware.¹

Era quindi da tenere in considerazione la retrocompatibilità delle librerie utilizzate, avendo scelto di supportare fino alla versione 16 del SDK (Android 4.1 Jelly Bean), e la possibilità di utilizzare l'applicazione anche su device con schermi e hardware differenti.

Si sono inoltre adottate alcune strategie implementative, come la creazione di custom view, per poter meglio strutturare il progetto favorendone anche eventuali modifiche o ampliamenti.

0.3.1 Strumenti di Sviluppo

Considerato che l'applicazione mobile, oggetto di tesi, era destinata a lavorare unicamente sulla piattaforma Android, si è scelto di utilizzare come principale strumento di sviluppo **Android Studio**, software dedicato alla programmazione nativa del suddetto sistema operativo.²

Durante lo sviluppo del progetto si è utilizzato come sistema di versioning il software **Git**, così da poter mantenere uno storico del lavoro svolto e ottenendo tutti i vantaggi dell'utilizzo di un Sistema per il Controllo di Versione (*Version Control System - VCS*).

¹Google Inc. *Android Developers*. URL: <https://developer.android.com/about/dashboards/index.html>.

²Google Inc. *Android Developers*. URL: <https://developer.android.com/training/basics/firstapp/index.html>.

Sono inoltre da citare alcuni software utilizzati durante il lavoro di tesi che, anche se di minore impatto, sono serviti a testare e sviluppare alcune parti fondamentali di codice. Primo fra tanti la web application Postman³ che permette di testare e sviluppare le APIs con cui due applicativi possono dialogare, in maniera semplice ed intuitiva. Utile per valutare le chiamate al server e per verificare le risposte ottenute.

Altri da citare???

Android Studio

Android Studio⁴ è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android. È stato annunciato il 16 maggio 2013 in occasione della conferenza Google I/O e la prima build stabile fu rilasciata nel dicembre del 2014. Basato sul software della JetBrains IntelliJ IDEA, Android Studio è stato progettato specificamente per lo sviluppo di Android.[4] È disponibile il download su Windows, Mac OS X e Linux,[5][6] e sostituisce gli Android Development Tools (ADT) di Eclipse, diventando l' IDE primario di Google per lo sviluppo nativo di applicazioni Android. Permette di creare un progetto gradle, apk, github,

Git

Git è un software di controllo versione distribuito (*Distributed Version Control Systems - DVCS*) utilizzabile da interfaccia a riga di comando, creato da Linus Torvalds nel 2005.⁵ Nacque per essere un semplice strumento per facilitare lo sviluppo del kernel Linux ed è diventato uno degli strumenti di controllo versione più diffusi.

Per quanto riguarda qualsiasi tipo di progetto di IT (*Information Technology*), specialmente se si tratta di un lavoro da dover svolgere in team, Git dà la possibilità di mantenere in memoria tutte le versioni del proprio lavoro (sia in locale che online). Questo permette a più

³Inc. Postdot Technologies. *Postman*. URL: <https://www.getpostman.com/>.

⁴Wikipedia. *Android Studio*. URL: https://it.wikipedia.org/wiki/Android_Studio.

⁵Wikipedia. *Git (software)*. URL: [https://it.wikipedia.org/wiki/Git_\(software\)](https://it.wikipedia.org/wiki/Git_(software)).

persone di poter accedere alla cronologia del lavoro condiviso, avendo anche la possibilità di verificare la presenza di errori e di eliminarli tornando ad una versione precedente del progetto. Git sfrutta alcuni algoritmi avanzati per calcolare le differenze riga per riga tra i file di diverse versioni così da verificare la presenza di errori o conflitti.

Git è fortemente direzionato verso uno sviluppo progettuale non lineare. Supporta diramazione e fusione (*branching and merging*) rapide e comode, e comprende strumenti specifici per visualizzare e navigare l'intero storico delle versioni anche se proposte da sistemi differenti. È veloce e scalabile nella gestione di grandi progetti ed è molto utile nello sviluppo in team, specialmente se affiancato dal modello GitFlow che permette di gestire rami specifici per il developing o per le release.

Infine grazie all'utilizzo di piccoli accorgimenti come il file `.gitignore` (che permette di non salvare anche i file temporanei e non importanti del progetto) e soluzioni online come GitHub o BitBucket (quest'ultimo utilizzato durante lo sviluppo di questa tesi) il processo di sviluppo software viene drasticamente avvantaggiato.

0.3.2 Programmazione Nativa

Questa tesi si inserisce all'interno del lavoro di modifica e miglioramento delle applicazioni mobile dell'ecosistema MyGelato, il quale includeva un progetto sia per piattaforma Android sia per piattaforma iOS. Essendosi creati due differenti progetti, durante la prima fase di progettazione, si è potuto valutare in maniera più libera l'utilizzo o meno della programmazione nativa.

Sicuramente lavorare in questo modo o meno ha i propri pro e contro. Mentre da un lato il nativo offre la possibilità di una gestione totale del dispositivo senza la paura di trovare limiti, d'altra parte richiede spesso una programmazione molto professionale e si concentra esclusivamente su una piattaforma impedendo un'agile riciclo dei propri sforzi su altri mercati del mobile. Il non-nativo, invece, offre diversi vantaggi ascrivibili ad una minore necessità di programmare e molto

spesso alla possibilità di creare applicazioni cross-platform distribuibili su sistemi operativi diversi.⁶

Non avendo quindi necessità di mantenere alta la portabilità del codice su altre piattaforme, si è preferito sviluppare tramite programmazione nativa; potendo quindi sfruttare pienamente l'architettura del sistema operativo sottostante.

Java + XML

0.3.3 Database

Realm

0.3.4 Chiamate Server

OkHttp

API REST

0.3.5 Gestore di Eventi

Un passaggio fondamentale durante lo sviluppo di un software è l'aggiornamento delle view a seguito di un cambiamento nello stato dell'applicazione. In Android è necessario intervenire direttamente sugli elementi dell'interfaccia modificando ogni oggetto in base alle specifiche. Una forte limitazione del sistema operativo è che l'unico thread che ha il permesso di accedere e di modificare l'interfaccia utente è il Main Thread; questo complica l'interazione con eventuali sistemi multithreading anche se ovviamente fa parte delle specifiche di sistema così da evitare conflitti.

Questo problema necessita di ampia valutazione durante lo sviluppo e anche in questo caso si è dovuto strutturare l'applicazione così che non incorresse in errori di inconsistenza tra le view a cui il programma voleva accedere e le view realmente visualizzate al momento. Nel caso si

⁶HTML.it. *Come sviluppare app Android, ibrido o nativo?* URL: <http://www.html.it/pag/48523/alternative-allo-sviluppo-nativo/>.

utilizzino script asincroni è molto facile tentare di accedere ad elementi, o anche intere activity, non più presenti nell'interfaccia utente. Per questo motivo si è scelto di utilizzare la libreria EventBus sviluppata da GreenRobot⁷ e pensata esattamente per risolvere questo tipo di errori.

EventBus

EventBus è una libreria open-source per Android che utilizza il pattern publish/subscribe per far dialogare tutti i componenti di un'applicazione. Disaccoppia le classi che interagiscono sulla stessa interfaccia e le gestisce in maniera centralizzata monitorando le performance e gestendo gli errori che potrebbero essere sollevati da uno sviluppo multithreading.⁸

I principali benefici che si riscontrano tramite l'utilizzo di questa libreria sono la semplificazione tra componenti, il disaccoppiamento tra mittenti e riceventi degli eventi, miglioramenti nelle performance, facile sviluppo e integrazione ed infine anche caratteristiche avanzate come priorità e indirizzamento a thread specifici.

Nel caso di questa tesi è stato necessario creare un insieme di eventi, specifici per ogni update da inviare all'UI, che avessero come metodi necessari il costruttore ed eventuali attributi da poter passare tra mittente e ricevente.

```
public static class UpdateCouponUiEvent {  
    public UpdateCouponUiEvent() {  
    }  
}
```

Grazie a questi eventi era possibile aggiornare l'interfaccia utente ad ogni modifica dello stato dell'applicazione sollevando l'invio di un evento specifico, gestito poi da EventBus, che contenesse le informazioni necessarie all'update (anche da thread diversi). Ogni evento aveva collegati dei metodi dell'activity che avevano effettuato il subscribe

⁷GitHub. *EventBus*. URL: <https://github.com/greenrobot/EventBus>.

⁸GreenRobot. *EventBus*. URL: <http://greenrobot.org/eventbus>.

ad un evento in particolare, così da essere scatenati solo in cui fosse sollevato quel particolare evento.

Esempio di subscribe:

```
@Subscribe
public void updateUI(UpdateCouponUiEvent event) {
    // update UI
}
```

Esempio di publish:

```
EventBus.getDefault().post(new UpdateCouponUiEvent());
```

0.3.6 Notifiche Push

Geofencing

Firebase

0.3.7 E-Commerce

Stripe

0.3.8 Librerie Minori

0.3.9 Scelte progettuali strane e che non so come chiamare

Flavors' Custom View

0.4 Implementazione

0.4.1 Shop

Ricerca

Preferiti

0.4.2 Carte gelato

Gelato Master

Notifiche Push

0.4.3 Coupon

Acquisto

Regalo

Utilizzo e Validazione

Bibliografia

- GitHub. *EventBus*. URL: <https://github.com/greenrobot/EventBus>.
- GreenRobot. *EventBus*. URL: <http://greenrobot.org/eventbus>.
- HTML.it. *Come sviluppare app Android, ibrido o nativo?* URL: <http://www.html.it/pag/48523/alternative-allo-sviluppo-nativo/>.
- Inc., Google. *Android Developers*. URL: <https://developer.android.com/about/dashboards/index.html>.
- *Android Developers*. URL: <https://developer.android.com/training/basics/firstapp/index.html>.
- Postdot Technologies, Inc. *Postman*. URL: <https://www.getpostman.com/>.
- Wikipedia. *Android Studio*. URL: https://it.wikipedia.org/wiki/Android_Studio.
- *Git (software)*. URL: [https://it.wikipedia.org/wiki/Git_\(software\)](https://it.wikipedia.org/wiki/Git_(software)).

Elenco delle figure

1	App Economy	4
---	-----------------------	---

0.5 Codice

0.6 Ringraziamenti