

Progettazione e Sviluppo di una applicazione mobile di Marketing ed E-Commerce

Tommaso Berlose

2016-11

Indice

| | | |
|----------|--|----------|
| 1 | Situazione esistente | 4 |
| 1.1 | App Economy | 4 |
| 1.2 | Ecosistema MyGelato | 4 |
| 1.3 | Stato dell'Arte dell'Applicazione Android | 4 |
| 2 | Specifiche di Progetto | 5 |
| 2.1 | Rifare app uguale | 5 |
| 2.2 | temi con flavors | 5 |
| 2.3 | risorse da adattare | 5 |
| 2.4 | parallelismo con iOS | 5 |
| 2.5 | multilingua, login facebook, ecc... | 5 |
| 3 | Scelte di Progetto | 6 |
| 3.1 | Strumenti di Sviluppo | 6 |
| 3.1.1 | Android Studio | 6 |
| 3.2 | Programmazione Nativa | 6 |
| 3.2.1 | Java + XML | 7 |
| 3.3 | Database | 7 |
| 3.3.1 | Realm | 7 |
| 3.4 | Chiamate Server | 7 |
| 3.4.1 | OkHttp | 7 |
| 3.4.2 | API REST | 7 |
| 3.5 | Gestore di Eventi | 7 |
| 3.5.1 | EventBus | 7 |
| 3.6 | Notifiche Push | 7 |
| 3.6.1 | Geofencing | 7 |
| 3.6.2 | Firebase | 7 |
| 3.7 | E-Commerce | 7 |
| 3.7.1 | Stripe | 7 |
| 3.8 | Librerie Minori | 7 |
| 3.9 | Scelte progettuali strane e che non so come chiamare | 7 |
| 3.9.1 | Flavors Custom View | 7 |
| 4 | Implementazione | 8 |
| 4.1 | Shop | 8 |
| 4.1.1 | Ricerca | 8 |
| 4.1.2 | Preferiti | 8 |
| 4.2 | Carte gelato | 8 |
| 4.2.1 | Gelato Master | 8 |
| 4.2.2 | Notifiche Push | 8 |
| 4.3 | Coupon | 8 |
| 4.3.1 | Acquisto | 8 |
| 4.3.2 | Regalo | 8 |
| 4.3.3 | Utilizzo e Validazione | 8 |

| | | |
|-----------|----------------------------------|-----------|
| 5 | Performance e Supporto | 9 |
| 5.1 | Profiling | 9 |
| 5.2 | Supporto vecchi device | 9 |
| 6 | Conclusioni | 10 |
| 7 | Bibliografia | 11 |
| 8 | Figure | 12 |
| 9 | Codice | 13 |
| 10 | Ringraziamenti | 14 |

Intro

1 Situazione esistente

1.1 App Economy

1.2 Ecosistema MyGelato

1.3 Stato dell'Arte dell'Applicazione Android

2 Specifiche di Progetto

2.1 Rifare app uguale

2.2 temi con flavors

2.3 risorse da adattare

2.4 parallelismo con iOS

2.5 multilingua, login facebook, ecc...

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 tenuto conto di tempistiche di aggiornamento, adeguamento alle linee guida del sistema operativo in oggetto, documentazione disponibile e modernità delle tecnologie utilizzate.

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

3.1 Strumenti di Sviluppo

Poichè l'applicazione mobile, oggetto di tesi, era da sviluppare in ambito Android, si è scelto di utilizzare come principale strumento di sviluppo Android Studio. In questo modo si è potuto programmare nativamente limitando la portabilità del codice su altre piattaforme ma sfruttando pienamente l'architettura del sistema operativo sottostante. Come sistema di versioning si è utilizzato

3.1.1 Android Studio

Android Studio [1] 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,

3.2 Programmazione Nativa

Nativo sì, nativo no. Qual è l'approccio migliore? Sicuramente entrambi hanno i loro pro e contro. Mentre da un lato il nativo offre la possibilità di una gestione totale del dispositivo senza la paura di trovare limiti, dall'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, anche se impossibile generalizzare data la diversità degli ambienti appena citati, offre vantaggi vari, ascrivibili a volte ad una minore necessità di programmare e molto spesso alla possibilità di creare applicazioni cross-platform distribuibili su sistemi operativi diversi.

- 3.2.1 Java + XML
- 3.3 Database
 - 3.3.1 Realm
- 3.4 Chiamate Server
 - 3.4.1 OkHttp
 - 3.4.2 API REST
- 3.5 Gestore di Eventi
 - 3.5.1 EventBus
- 3.6 Notifiche Push
 - 3.6.1 Geofencing
 - 3.6.2 Firebase
- 3.7 E-Commerce
 - 3.7.1 Stripe
- 3.8 Librerie Minori
- 3.9 Scelte progettuali strane e che non so come chiamare
 - 3.9.1 Flavors Custom View

4 Implementazione

4.1 Shop

4.1.1 Ricerca

4.1.2 Preferiti

4.2 Carte gelato

4.2.1 Gelato Master

4.2.2 Notifiche Push

4.3 Coupon

4.3.1 Acquisto

4.3.2 Regalo

4.3.3 Utilizzo e Validazione

5 Performance e Supporto

5.1 Profiling

5.2 Supporto vecchi device

6 Conclusioni

7 Bibliografia

- [1] HTML.it, “Come sviluppare app android, ibrido o nativo?,” 2016.

8 Figure

9 Codice

10 Ringraziamenti