

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

Tommaso Berlose

2016-11

# Indice

0.1	Contesto e Finalità . . . . .	4
0.1.1	Application Economy . . . . .	5
0.1.2	Ecosistema MyGelato . . . . .	7
0.2	Specifiche Progettuali . . . . .	9
0.2.1	Ricerca . . . . .	10
0.2.2	Marketing Digitale . . . . .	11
0.2.3	E-Commerce . . . . .	13
0.2.4	Design . . . . .	13
0.3	Progettazione . . . . .	14
0.3.1	Strumenti di Sviluppo . . . . .	14
0.3.2	Programmazione Nativa . . . . .	17
0.3.3	Database . . . . .	18
0.3.4	Chiamate Server . . . . .	18
0.3.5	Gestore di Eventi . . . . .	20
0.3.6	Notifiche Push . . . . .	22
0.3.7	E-Commerce . . . . .	22
0.3.8	Librerie Minori . . . . .	22
0.3.9	Scelte progettuali strane e che non so come chiamare . . . . .	22
0.4	Implementazione . . . . .	23
0.4.1	Marketing Digitale . . . . .	23

0.4.2	E-Commerce . . . . .	23
-------	----------------------	----

## Intro

Il progetto MyGelato nasce nel 2014, da un'idea dell'azienda Carpigiani, come piattaforma di e-commerce e marketing digitale fruibile tramite applicazione mobile. Sviluppato inizialmente da terzi, a causa del cambiamento dei requisiti funzionali e di progettazione, viene poi riportato internamente all'azienda

Il progetto MyGelato, applicazione mobile ideata dall'azienda Carpigiani, nasce nel 2014 come piattaforma di marketing ed e-commerce sviluppata da terzi. Il cambiamento dei requisiti funzionali della piattaforma ha portato a far rinascere il progetto iniziale in un prodotto finale completamente gestito internamente all'azienda. Questa tesi presenta il design e l'implementazione di una piattaforma fruibile da web e dispositivi mobile con lo scopo di supportare un alto numero di utenti mantenendo controllati i costi e le performance. Le tecnologie scelte e la generalità della soluzione consentono di impiegare gli stessi principi per altri progetti con architetture analoghe.

## 0.1 Contesto e Finalità

Per comprendere nella sua complessità un progetto di questo tipo, è necessario tenere conto prima di tutto del contesto in cui viene sviluppato e delle finalità implicite ed esplicite che si vogliono raggiungere. Nel caso di questo elaborato si tratta, prima di tutto, di descrivere lo sviluppo di un applicativo mobile fortemente legato ai sistemi di marketing e alle nuove strategie di business di aziende leader all'interno dei propri mercati, in molti casi saturi o fortemente instabili, scossi dalle innovazioni tecnologiche di questi decenni.

Oltre alle ingerenze esterne sul progetto, come il cambiamento repentino delle tecnologie o la presenza di eventuali competitor, deve essere valutata la posizione del singolo componente all'interno dello sviluppo dell'intero progetto, la quale va tenuta in forte considerazione specialmente durante la prima fase di sviluppo in cui si determinano le prime linee guida da seguire durante tutto il lavoro.

L'application economy descrive perfettamente il trend degli ultimi anni che hanno portato uno grosso sconvolgimento nel sistema di marketing anche di aziende multinazionali a seguito del cambiamento dei metodi con cui le masse ottengono informazioni e si lasciano influenzare: advertising, online coupon, ecc... Proprio in questo contesto è necessario inquadrare le motivazioni che hanno portato l'azienda Carpigiani a dare vita all'ecosistema MyGelato, di cui questa tesi sviluppa solo una componente, così da realizzare quanto progetti di questo tipo siano fondamentali in questo momento storico. Si tratta di uno studio in ambito pubblicitario, che ha finalità implicite legate fortemente all'ambito commerciale e produttivo dell'azienda.



Figura 1: Applications Economy

### 0.1.1 Application Economy

Non vi è forse modo di descrivere la società attuale e questo periodo storico senza valutare l'importanza dello sviluppo tecnologico che ci ha portato nell'*Era dell'Informazione*. La rivoluzione tecnologica che sta avvenendo in questi anni, specialmente a partire dagli Novanta, ha portato la connessione globale ad internet ad assumere un ruolo essenziale in ogni aspetto della società moderna e della nostra vita.

È cambiato drasticamente il modo con cui le persone accedono alle informazioni, che siano queste di tipo personale o di tipo commerciale. Allo stesso modo si sono dovute adeguare le strategie di tutte quelle aziende che hanno visto cambiare in maniera drastica il proprio mercato, invaso molte volte da tecnologie sempre diversificate e innovative.

Secondo il giornalista Paul Mason, è stata la dottrina economica degli ultimi decenni della nostra epoca che, da una parte ha avuto il merito di promuovere la più

grande ondata di sviluppo economico che il mondo abbia mai visto, ma dall'altra ha portato a mercati incontrollati e a vorticosi cambiamenti sociali innescati dalla tecnologia. Si sono diffusi, infatti, concetti come i progetti open-source, la sharing economy e le licenze creative commons che hanno messo in crisi le fondamenta del capitalismo odierno: la proprietà privata. Questo fenomeno unito alla saturazione di molti mercati ha portando tante aziende a dover rivedere la propria business strategy obbligandole a dirigere i propri investimenti sulla diversificazione e sulle nuove strategie di vendita.<sup>1</sup>

Prime su tutte sono diventati di fondamentale importanza l'e-commerce e il marketing digitale, fortemente spinti dalle tecnologie e dalla nuova possibilità di accedere ad una risorsa comune (internet) da parte della maggior parte della persone anche di cultura, età e ambienti sociali diversi. Legato alla diffusione sempre crescente di smartphone, nasce quindi l'Application Economy: lo sviluppo e l'utilizzo di applicazioni mobile multiplatforma per raggiungere gli utenti consumatori. Questa strategia può essere applicata, per esempio, in ambito marketing per pubblicizzare un proprio prodotto, fidelizzare il consumatore alla propria azienda e mantenere una propria immagine in un sistema in cui l'idea che i consumatori hanno dell'azienda è fondamentale.

Forse ancora in via di sviluppo per certi settori ma ormai sempre più diffuso, è l'e-commerce, la possibilità di fare acquisti online, che sta spostando il modo di comprare online e sempre più sui device mobile (una statistica del BI Intelligence riporta che entro il 2020 il commercio mobile supererà il 45% degli acquisti online complessivi). Questa soluzione permette all'utente di pagare direttamente tramite sistemi bancari e non più tramite contanti, spostando inoltre il controllo delle vendite dal rivenditore al fornitore di servizi.

---

<sup>1</sup>Paul Mason. *Postcapitalismo*. 2016.

### 0.1.2 Ecosistema MyGelato

L’ecosistema MyGelato si pone proprio all’interno di questo contesto, ad unire le strategie di marketing dell’azienda produttrice Carpigiani e delle singole gelaterie convenzionate; insieme alla possibilità di comprare, regalare e utilizzare online dei coupons che permettono l’acquisto di gelati.

È un progetto di piattaforma web e mobile che si rivolge al pubblico dei gelatieri e dei consumatori di gelato. Gli obiettivi commerciali sono diversi, primi tra i quali quello di incentivare la compravendita di gelati tramite il sistema di coupons digitali e quello di promuovere il consumo di gelato tramite offerte, fornendo anche ai gelatieri un sistema semplice ed efficace per pubblicizzare il proprio negozio.

Attraverso l’applicativo, l’utente può informarsi sulle gelaterie presenti in zona verificando anche quali facciano parte del circuito MyGelato. È possibile selezionare ogni gelateria così da avere alcune informazioni utili, come indirizzo e numero di telefono, salvarle nei preferiti e verificare se vi sono promozioni attive. Nel momento in cui l’utente aggiunge ai preferiti una gelateria, verrà informato tramite notifica push di eventuali nuove promozioni attive e sarà notificato l’essere arrivato vicino ad una gelateria di proprio interesse. Queste strategie sono ovviamente di carattere pubblicitario e permettono anche a gelaterie minori, facenti però parte del circuito, di essere pubblicizzate dagli utenti che utilizzano la piattaforma.

Questo sistema permette all’azienda Carpigiani di avere un controllo maggiore su un sistema centralizzato di advertasing per le gelaterie di cui è principale fornitore, ottenendo quindi la possibilità di intervenire su realtà locali standardizzandone alcuni meccanismi.

Seconda feature fondamentale dell’applicazione, che riguarda in questo caso solo gli utenti iscritti al sistema, è la possibilità di comprare online dei coupon digitali che permettono l’acquisto, tramite bruciatura, di un gelato di una determinata gelateria. Questi coupon sono visualizzabili in ogni momento e possono essere



regalati anche ad altri utenti tramite un semplice metodo di condivisione/riscatto che permette al destinatario di usufruire dell'oggetto comprato. Ogni gelateria che supporta questo sistema sarà provvista della stessa applicazione con un account gelatiere che permetterà di bruciare i coupon utilizzati, per i quali verrà in seguito rimborsata direttamente dall'azienda.

Questo sistema di e-commerce sposta la vendita di un prodotto molto semplice come i gelati, online; È infatti possibile comprarli tramite carta di credito e la gestione della compravendita viene spostata dalla gelateria all'azienda Carpigiani che si incarica poi di ripagare il rivenditore. Questo sistema, per quanto diffuso in altri mercati come il reseller online di oggettistica è fortemente innovativo in questo campo e rende l'ecosistema MyGelato uno dei primi nel suo settore.

L'ecosistema MyGelato si fonda fisicamente su una piattaforma formata da un'applicazione basata sul framework Ruby on Rails che implementa le funzionalità di creazione, gestione e fruizione di contenuti multimediali affiancate da un sistema e-commerce per la compravendita di coupon digitali. Per poter usufruire di tali servizi è necessario l'utilizzo di un'applicazione mobile, chiamata MyGelato, disponibile per iOS e Android; quest'ultima argomento di questa tesi.

## 0.2 Specifiche Progettuali

L'ecosistema MyGelato è un progetto ampio che si basa sulla cooperazione tra più elementi fondamentali - applicazione mobile, piattaforma di backend, sistema di comunicazione, sistema di e-commerce, ecc... - la maggior parte dei quali sviluppati da persone diverse con tempistiche differenti. Alle specifiche tecniche richieste dall'azienda durante la progettazione dell'ecosistema MyGelato, si sono quindi aggiunte alcune limitazioni legate alle scelte effettuate da altre persone durante lo sviluppo del loro applicativo.

Durante lo studio iniziale si è dovuto tener conto delle specifiche legate alle features richieste e delle specifiche progettuali già implementate cercando di ottenere uno sviluppo lineare e il più semplice possibile.

Le scelte legate alle richieste da parte dell'azienda che ha commissionato il progetto riguardano principalmente le scelte strutturali dell'applicativo come la navigazione, il design e i flussi logici presenti. Aggiungendo alcune richieste tecnologiche come la possibilità di avere un'applicazione multilingua, con autenticazione tramite social network e compatibile con la maggior parte dei device android attualmente sul mercato.

Le specifiche dettate dalla cooperazione tra le varie componenti del sistema, invece, hanno avuto un impatto molto maggiore sulle tecnologie scelte e utilizzate per lo sviluppo. In particolar modo sono state di forte impatto alcune scelte a livello di comunicazione con il backend che ha indirizzato lo sviluppo in una comunicazione bidirezionale tra applicazione e backend tramite API REST e notifiche push. Essenziale anche mantenere la coerenza tra le due applicazioni mobile sviluppate per le piattaforme iOS e Android; come anche per la piattaforma gestionale presente per gli amministratori del sistema.

Le specifiche progettuali di questo taglio prettamente tecnologico saranno trattate nel dettaglio all'interno delle scelte progettuali mentre quelle legate ai flussi

logici dell'applicazione sono descritte nei seguenti capitoli.

### 0.2.1 Ricerca

L'elemento fondamentale di tutto l'applicativo, tema di questo elaborato, è la possibilità di effettuare una ricerca delle gelaterie, facenti parte del sistema My-Gelato. Questa funzionalità è, infatti, l'elemento cardine che lega ogni possibile azione dell'utente all'interno dell'applicazione: risulta infatti essere essenziale sia per il sistema di marketing che di e-commerce. Ogni utente, anche se non registrato, deve poter accedere ad una mappa che visualizzi tutti gli shop, sia sotto forma di marker visibili in base al luogo di ricerca sia sotto forma di lista ordinata in base alla distanza dall'utente. Ogni elemento, scaricato e aggiornato in base alle ultime informazioni presenti sul server, permette di ottenere il nome della gelateria, l'indirizzo ed un eventuale recapito telefonico.

Questo tipo di ricerca permette all'utente di esplorare la zona attorno al luogo in cui trova, o anche ad un luogo di suo interesse, così da ottenere informazioni riguardo agli esercizi presenti in maniera molto diretta. Il consumatore ottiene informazioni dettagliate e mirate rispetto ad una qualsiasi ricerca online che porterebbe anche una serie di informazioni aggiuntive molte volte non pertinenti.

Nel frattempo ogni gelateria presente all'interno del sistema ottiene la possibilità di essere trovati anche da utenti nuovi che non conoscono la zona e di invogliarli tramite una strategia di marketing unificata e standardizzata: le carte promozionali.

Oltre al sistema di marketing la ricerca delle gelaterie è essenziale per il sistema di e-commerce, poichè ogni coupon acquistabile sulla piattaforma è legato ad un singolo esercizio che deve essere specificato una volta deciso il proprio acquisto.

La ricerca deve quindi essere presente sia nel sistema di navigazione principale dell'applicazione che raggiungibile all'interno di ogni flusso logico scelto dall'utente.

## 0.2.2 Marketing Digitale

Uno dei due flussi logici principali presenti all'interno dell'applicazione riguarda il marketing digitale che viene svolto per le gelaterie legate all'ecosistema MyGelato. Il sistema di marketing ha lo scopo di rendere più semplice la ricerca delle gelaterie vicino all'utente, sponsorizzarne eventuali promozioni e permettere all'utente di salvare gli esercizi preferiti così da rimanere aggiornato su eventuali nuove sponsorizzazioni. Queste funzionalità, disponibili in buona parte per qualsiasi utente anche non registrato, seguono un flusso logico che parte dalla ricerca degli shop e trova effetto principale nella scoperta e nell'utilizzo delle carte promozionali, della singola gelateria e del Mastro Gelatiere.

### Carte Promozionali

Le carte promozionali sono il principale mezzo di advertising all'interno dell'applicazione: sono dei volantini digitali formati da un'immagine, un titolo, un testo descrittivo, un eventuale recapito telefonico e un eventuale link per avere maggiori informazioni. Ci sono due tipologie di carte: le carte specifiche di ogni esercizio che sono i volantini informativi della singola gelateria e le carte del mastro gelatiere che invece si rifanno alle promozioni generiche proposte per l'intero sistema.

Le carte promozionali sono il componente fondamentale del sistema di marketing digitale dell'applicativo, poichè

Sono inoltre visualizzabili le Carte dello shop che altro non sono che le ultime promozioni legate all'attività: sconti, novità, messaggi promozionali. Grazie a queste informazioni l'utente può ottenere oltre le informazioni genericamente trovabili online, anche delle informazioni pubblicitarie. Dall'altra parte del sistema la gelateria ha la possibilità di ottenere una maggior copertura pubblicitaria tramite l'utilizzo di un sistema centralizzato e legato da sistemi di advertising.

## Preferiti

Elemento molto importante, presente all'interno della navigazione principale dell'applicazione, è la gestione delle proprie gelaterie preferite. Questo sistema permette ad ogni utente, registrato o meno, di salvare offline sul proprio dispositivo gli esercizi di maggiore interesse.

Questo dà modo all'utente di poter accedere in ogni momento, specialmente in mobilità, alle informazioni che più gli interessavano di ogni gelateria: indirizzo, recapito telefonico ed eventuali promozioni. La ricerca in questo caso è molto semplificata poichè viene presentata una sezione a parte con una lista degli shops preferiti, senza dover obbligare l'utente ad effettuare una nuova ricerca all'interno della mappa.

Il salvataggio all'interno dei preferiti avviene direttamente all'interno della ricerca, sia tramite la mappa che tramite la lista ordinata in base alla distanza dall'utente grazie ad un'icona esemplificativa.

Questa funzionalità è pensata principalmente per fidelizzare il consumatore: ogni utente avendo la possibilità di salvare una gelateria ha anche la possibilità di ottenere velocemente informazioni su nuove promozioni, ottenere velocemente i contatti dell'esercizio come se li avesse salvati in rubrica e valutare ogni volta la distanza tra se e lo shop.

Il passo successivo in questo senso è dato dal rendere bidirezionale questo collegamento, rendendo in alcuni casi non necessaria la ricerca da parte dell'utente di nuove informazioni dandogliele invece ad ogni aggiornamento. Per fare questo, solo nel caso di utenti registrati, l'aggiunta di uno shop ai preferiti include alcune funzionalità di maggiore visibilità: geofencing e notifiche push. Il geofencing permette di ricevere una notifica ogni qualvolta l'utente si trovi a meno di 5 km da una delle proprie gelaterie, così da essere informato di essere vicino in termini di localizzazione. Le notifiche push invece sono attivate per avvertire un utente che

una delle proprie gelaterie ha pubblicato una nuova promozione tramite l'aggiunta una carta sul sistema MyGelato. L'utente così rimane informato costantemente delle ultime promozioni disponibili e il proprietario di una gelateria ha la certezza di effettuare pubblicità diretta tra se e i suoi clienti più affezionati.

### **0.2.3 E-Commerce**

**Acquisto Coupon**

**Utilizzo Coupon**

### **0.2.4 Design**

## 0.3 Progettazione

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.<sup>2</sup>

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

---

<sup>2</sup>Google Inc. *Android Developers*. URL: <https://developer.android.com/about/dashboards/index.html>.

nativa del suddetto sistema operativo.<sup>3</sup>

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*).

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 Postman<sup>4</sup> 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 Studio<sup>5</sup> 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,

---

<sup>3</sup>Google Inc. *Android Developers*. URL: <https://developer.android.com/training/basics/firstapp/index.html>.

<sup>4</sup>Inc. Postdot Technologies. *Postman*. URL: <https://www.getpostman.com/>.

<sup>5</sup>Wikipedia. *Android Studio*. URL: [https://it.wikipedia.org/wiki/Android\\_Studio](https://it.wikipedia.org/wiki/Android_Studio).



## 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.<sup>6</sup> 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 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.

---

<sup>6</sup>Wikipedia. *Git (software)*. URL: [https://it.wikipedia.org/wiki/Git\\_\(software\)](https://it.wikipedia.org/wiki/Git_(software)).

### 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.<sup>7</sup>

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.

---

<sup>7</sup>HTML.it. *Come sviluppare app Android, ibrido o nativo?* URL: <http://www.html.it/pag/48523/alternative-allo-sviluppo-nativo/>.

**Java**

**XML**

### **0.3.3 Database**

**Realm**

### **0.3.4 Chiamate Server**

Attualmente ogni tipo di piattaforma che voglia permettere al proprio utente di accedere da remoto, e quindi da qualsiasi dispositivo, alle proprie informazioni ha come parte fondamentale lo sviluppo e il mantenimento di un server di backend (quindi nascosto nelle funzionalità alla vista dell'utente) che mantenga, gestisca e restituisca i dati necessari alle funzioni degli applicativi. Nello stesso modo la piattaforma MyGelato si basa su un server che mantiene ogni informazione riguardante utenti, shop e cards fornendo delle API REST che vengono sfruttate dalle applicazioni mobile e web per poter fornire i propri servizi.

Il passaggio di informazioni tra i due applicativi avviene tramite chiamate internet con protocollo http protette grazie ad una connessione criptata dal Transport Layer Security (TLS) o dal suo predecessore, Secure Sockets Layer (SSL). Grazie alla API REST è possibile sviluppare in maniera strutturata e di seguire alcuni pattern delle tecnologie che permettono di valutare ogni possibile errore delle connessioni: errori nei dati, nell connessione, ecc... In Java le chiamate http sono abbastanza difficili da gestire poichè non sono automatizzate e son stati effettuati dei cambiamenti tra le ultime versioni che non permettono di lavorare correttamente su tutti i dispositivi Android. Per questo motivo è stata utilizzata la libreria open-source OkHttp sviluppata da Square, che permette di automatizzare le chiamate http sfruttando anche una notevole semplificazione del codice utilizzato per

programmare.<sup>8</sup>

## OkHttp

La libreria OkHttp nasce dal fatto che il protocollo http è l'attuale mezzo di comunicazione per ogni applicativo moderno. Per questo si pone l'obiettivo di semplificarne l'utilizzo in più di una tecnologia migliorandone le prestazioni e fornendo allo sviluppatore alcuni automatismi comodi durante lo sviluppo e la programmazione.

Supporta ogni tipo di comunicazione http, ne gestisce la latenza, il caching, la riconnessione, l'utilizzo di protocolli sicuri come TLS, l'utilizzo di indirizzi IP multipli ed è molto semplice da integrare e utilizzare all'interno del proprio software. Grazie ad un'interfaccia molto semplice permette di automatizzare le chiamate http utilizzando dei Builder che lasciano specificare solo i parametri necessari e restituiscono un unico oggetto da cui si ottiene la risposta, positiva o negativa, della richiesta.

```
OkHttpClient client = new OkHttpClient();
Request request = new Request.Builder()
    .url(endpoint)
    .addHeader("Accept", "application/json")
    .build();

Response response = client.newCall(request).execute();
```

Inizializzato il client OkHttp basta creare un oggetto request a cui vanno passati i parametri necessari; basta poi eseguire la chiamata per ottenere un oggetto response all'interno del quale sono presenti i dati scaricati, il codice di risposta della chiamata e anche eventuali errori.

---

<sup>8</sup>Square. *OkHttp*. URL: <http://square.github.io/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 richieste senza poter sfruttare veri e propri automatismi. Una forte limitazione del sistema operativo è dovuta, per esempio, al fatto 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 utili a limitare i conflitti sull'accesso alle risorse condivise (in questo caso le view).

Questo problema necessita di ampia valutazione durante lo sviluppo e anche in questo caso si è dovuto strutturare l'applicazione in modo 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 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 GreenRobot<sup>9</sup> 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 centra-

---

<sup>9</sup>GitHub. *EventBus*. URL: <https://github.com/greenrobot/EventBus>.

lizzata monitorando le performance e gestendo gli errori che potrebbero essere sollevati da uno sviluppo multithreading.<sup>10</sup>

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

Durante lo svolgimento di questa tesi si è definita una struttura abbastanza comune formata da: - AsyncTask, oggetto che permette di elaborare una serie di informazioni in background, utilizzato per eseguire il download delle informazioni dal server; - View specifica dell'activity con determinati componenti da aggiornare in base allo stato dell'applicazione; - Metodo `updateUI()` che preso in gresso i dati scaricati doveva aggiornare le view rispetto allo stato.

Per far dialogare questi elementi si è utilizzato EventBus così da sfruttare il lavoro di thread in background senza limitare l'utente nell'attesa di un determinato evento. Si procedeva quindi creando un evento specifico per l'interazione da dover gestire di cui qui viene riportato un esempio:

```
public static class UpdateCouponUiEvent<T> {  
    public T data;  
    public UpdateCouponUiEvent() {  
    }  
    public UpdateCouponUiEvent(T data) {  
        this.data = data;  
    }  
}
```

EventBus permette di registrare dei metodi eseguendo una `subscribe` ad un determinato evento: ogni qual volta un evento di quel tipo viene sollevato, il

---

<sup>10</sup>GreenRobot. *EventBus*. URL: <http://greenrobot.org/eventbus>.

metodo viene eseguito. La subscribe deve essere definita nel codice nel seguente modo:

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

Grazie all'utilizzo di eventi custom è dunque possibile passare anche dei dati tramite l'evento che viene lanciato, nel codice messo ad esempio si tratta della variabile data, definito di tipo generico.

Infine all'interno del AsyncTask, a conclusione del download delle informazioni e dell'aggiornamento dello stato, era sollevato un evento grazie ad una 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 Marketing Digitale**

**Ricerca**

**Preferiti**

Geofencing e notifiche push

### **0.4.2 E-Commerce**

**Acquisto Coupons**

**Regalo Coupons**

**Utilizzo e Validazione Coupons**





# 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>.
- Mason, Paul. *Postcapitalismo*. 2016.
- Postdot Technologies, Inc. *Postman*. URL: <https://www.getpostman.com/>.
- Square. *OkHttp*. URL: <http://square.github.io/okhttp/>.
- Wikipedia. *Android Studio*. URL: [https://it.wikipedia.org/wiki/Android\\_Studio](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	Applications Economy . . . . .	5
---	--------------------------------	---