

Introduzione

Il lavoro dello sviluppatore mobile sta diventando sempre più difficile. Le piattaforme che stanno ormai dominando il mercato da alcuni anni, Android e iOS, rilasciano nuove versioni con un ritmo impressionante, mettendo a disposizione dello sviluppatore strumenti sempre nuovi e più potenti.

Se da un lato questo è sicuramente un aspetto positivo, per la possibilità di realizzare applicazioni sempre nuove, dall'altro richiede un enorme sforzo per rimanere aggiornati con tutte le API che vengono sfornate in continuazione da Google e Apple. A ogni *Google IO*, la conferenza degli sviluppatori di Google, viene annunciata una nuova versione del sistema operativo Android, con modifiche anche sostanziali e lo stesso succede per iOS in corrispondenza della WWDC di Apple.

Nello specifico, all'ultimo *Google IO* di maggio 2015 è stata annunciata la versione 6 di Android, denominata Marshmallow, che sarà argomento del presente testo, mentre all'ultimo WWDC è stato presentato iOS 9, che continua il percorso successivo all'introduzione del nuovo linguaggio Swift, giunto alla versione 2.1.

Il lavoro di chi scrive libri tecnici è quindi ancora più complicato in quanto i tempi di realizzazione di un testo come questo devono essere necessariamente serrati. La sfida è quella di trattare argomenti che sono alla base dello sviluppo delle applicazioni Android e che siano il più possibile indipendenti da quello che accadrà. Potrà quindi succedere che qualche immagine sia leggermente diversa da quello che il lettore ritroverà durante lo sviluppo, ma si tratterà comunque di modifiche non sostanziali. Come per il precedente testo dedicato ad Android 4, abbiamo deciso di seguire un approccio pratico, che consiste nella realizzazione di un'applicazione da zero. La versione 6 di Android è caratterizzata non solo da nuove API, ma anche dall'introduzione di un nuovo design, che si chiama *Material Design*, che prevede l'utilizzo di alcuni pattern grafici e di interazione che Google consiglia di seguire nella realizzazione delle applicazioni non solo per la piattaforma Android ma anche per iOS e il Web. Questo motivo, unito alla necessità di creare cose sempre nuove, ci porterà questa volta a realizzare una nuova applicazione che chiamiamo *ApoBus* e che permetterà di visualizzare informazioni relative ai percorsi e agli orari dei bus di un'ipotetica città. Ci occuperemo dello sviluppo dell'applicazione Android utilizzando servizi fittizi per quello che riguarda la raccolta delle informazioni, ma, attraverso un buon design dei vari componenti, il lettore potrà adattare il proprio lavoro ad altri servizi, magari per la propria città. Nella realizzazione di questa applicazione utilizzeremo *Android Studio*, nella versione 1.5.

Struttura del testo

Il Capitolo 1 è introduttivo e ci permetterà di scaldare i motori in preparazione del progetto *ApoBus*. Vedremo che cosa è Android, che relazione ha con Java e descriveremo i concetti fondamentali alla base del linguaggio. Non si tratterà, naturalmente, di un manuale Java completo, ma di una descrizione, molto pragmatica, di quello che ci sarà utile successivamente nella creazione delle applicazioni. In particolare vedremo che cosa è un *package*, in quanto legato al concetto di applicazione Android, ed i principali concetti di *object orientation*. Ci occuperemo quindi di *Delegation Model*, ovvero del meccanismo che sta alla base della gestione degli eventi, di *inner classes* e quindi di *generics*. L'obiettivo di questo primo capitolo è quello di preparare il lettore al viaggio verso la realizzazione di *ApoBus* che, oltre che essere una applicazione Android, sarà anche un esercizio utile e divertente. Il Capitolo 2 è di fondamentale importanza, in quanto descrive le parti fondamentali di un qualunque progetto Android. Vedremo infatti come creare il progetto e come utilizzare al meglio *Gradle* come strumento per il *build* dell'applicazione. Vedremo come creare diverse versioni di una stessa applicazione attraverso il concetto di *build variants*, insieme alla possibilità di gestire le varie dipendenze. Una volta creato il progetto ci occupiamo di descrivere il ruolo delle sue parti fondamentali, ovvero dei sorgenti Java, delle risorse e del file di configurazione. L'ultima parte del capitolo esegue il collaudo dell'applicazione in un emulatore e poi in un dispositivo reale.

Nel Capitolo 3 andremo a descrivere la nostra applicazione *ApoBus*, definendo una bozza di quello che sarà il flusso di navigazione. Con il pretesto di creare una schermata di *Splash*, ci occuperemo di vari aspetti relativi alla gestione delle *Activity* e dei principali tipi di risorse. In particolare ci occuperemo del ciclo di vita di una *Activity* e delle varie modalità con cui questi componenti possono comunicare tra loro. Sempre in questo capitolo ci occuperemo dei concetti fondamentali di *Intent* e *IntentFilter*, alla base di tutta la piattaforma Android. Poi avremo l'opportunità di affrontare uno dei problemi più comuni nello sviluppo delle applicazioni Android, ovvero l'individuazione e la risoluzione dei *memory leak*.

Il Capitolo 4 è dedicato ai *Fragment*, che permettono non solo di suddividere l'interfaccia utente (UI) di un'applicazione in parti da riutilizzare in display di dimensioni diverse, ma forniscono anche un meccanismo per il mantenimento dello stato a seguito di una rotazione o comunque di una variazione della configurazione dell'applicazione, che vedremo comportare il riavvio dell'attività corrente. Vedremo diversi tipi di *Fragment* e in particolare il ciclo di vita, oltre che particolari specializzazioni per la visualizzazione delle finestre di dialogo. Utilizzeremo poi i *Fragment* per creare lo scheletro della nostra applicazione, attraverso la *Design Support Library* con i primi componenti *Material Design* e in particolare il *DrawerLayout* e la *NavigationView*.

Sempre in questo capitolo accenneremo alle risorse di tipo menu, che sono invece argomento del Capitolo 5 insieme a due componenti fondamentali delle applicazioni Android ovvero *ActionBar* e quindi *ToolBar*. Vedremo infatti come creare e gestire una *ActionBar* attraverso opportune risorse di tipo menu, e poi come implementare le stesse funzionalità attraverso un componente molto più flessibile, ovvero la *ToolBar*. Nello specifico vedremo come gestire alcune modalità di navigazione come quello a *Tab* e poi come gestire delle *ActionView*, che non sono altro che particolari componenti, anche personalizzati, che vengono collocati nella parte superiore del display e che permettono di eseguire operazioni particolari, come quelle di ricerca o selezione multipla. Vedremo

questi concetti e strumenti attraverso una ricca serie di esempi, per poi concludere il capitolo applicando alcuni di questi concetti alla nostra applicazione *ApoBus*.

Il Capitolo 6 è di fondamentale importanza, perché descrive le caratteristiche principali di ciascun componente visuale, ovvero di tutti quei componenti descritti da classi che estendono, direttamente o indirettamente, le classi `View` e `ViewGroup`. Vedremo i principali layout e approfondiremo la nostra conoscenza su altri tipi di risorse, come `Drawable` `style` e `theme`. Sempre in questo capitolo impareremo a utilizzare i principali componenti messi a disposizione dalla *Design Support Library* e a gestirne la coordinazione attraverso il *CoordinatorLayout* e poi vedremo come creare una *Custom View*.

Il Capitolo 7 tratta le modalità di utilizzo di due componenti fondamentali per ogni applicazione Android, ovvero `ListView` e `RecyclerView`. Attraverso moltissimi esempi pratici, vedremo come utilizzare questi componenti e soprattutto come personalizzarli per quello che riguarda l'aspetto e l'interazione con l'utente. Anche in questo capitolo avremo modo di applicare concetti di *Material Design*.

Il Capitolo 8 è dedicato a una nuova libreria che sta riscuotendo moltissimo successo: la *Data Binding Library*. Essa permette di semplificare l'operazione di *binding*, ovvero di mappatura dei dati di un modello sulle `View` di un particolare layout. Per chi ha esperienza di sviluppo web ci sarà l'occasione per ricordare diversi meccanismi caratteristici delle pagine JSP (*Java Server Pages*) utilizzate nella creazione di pagine web dinamiche, il tutto trasportato in un ambiente mobile.

Fino a questo punto ci siamo occupati principalmente degli aspetti legati alla gestione della UI mentre nel Capitolo 9 inizieremo a descrivere ciò che riguarda la gestione dei dati. In particolare vedremo tutto ciò che riguarda la persistenza delle informazioni in ambito Android. Cominceremo a descrivere le `SharedPreferences`, che ci permetteranno di gestire le preferenze dell'applicazione. Dopo una breve descrizione di come i file vengano gestiti dalla piattaforma, vedremo in dettaglio come gestire un DBMS come `SQLite`. A tale proposito vedremo diversi casi d'uso che ci permetteranno di implementare la funzionalità di ricerca delle fermate dei bus. Concluderemo questo fondamentale capitolo con la descrizione dei `ContentProvider`, attraverso i quali le applicazioni possono condividere informazioni tra processi diversi.

Il Capitolo 10 è forse il più importante di tutto il testo, in quanto tratta i concetti che stanno alla base della realizzazione di applicazioni con un grande livello di *reattività* ovvero della capacità di interagire con l'utente nel modo più efficiente possibile, senza alcun rallentamento. Il capitolo è dedicato ai concetti di *multithreading*. Dopo una descrizione dei concetti legati alla gestione dei `Thread` in Java, si passa ai concetti di `Handler` e `Looper`. Si tratta di oggetti che stanno alla base dell'utilizzo degli `AsyncTask`, che utilizzeremo nella nostra applicazione per accedere al database in modalità asincrona. Una tipica informazione che viene visualizzata in modalità asincrona è quella delle notifiche. La parte centrale del capitolo è dedicata proprio a questo importante componente, che ha avuto delle modifiche nelle ultime versioni della piattaforma. Vedremo come funziona il `NotificationManager` e creeremo un esempio che permetta di descrivere ciascun tipo di notifica. La terza parte del capitolo sarà dedicata alla descrizione di un altro componente di fondamentale importanza, ovvero i `Service`. Si tratta di un componente ideale per l'esecuzione di task di lunga durata; possono essere di due tipi: `Started` e `Bound`. Vedremo in dettaglio che cosa sono e alcuni casi d'uso. Un altro componente tipico della piattaforma Android legato alla possibilità di eseguire operazioni in modalità asincrona è `BroadcastReceiver`, cui viene dedicata la penultima parte del capitolo, per poi passare alla

descrizione dei Loader. Si tratta di un piccolo framework che ci permetterà di accedere alle informazioni in modo sincrono, ma soprattutto di ricevere notifiche nel caso in cui i dati dovessero cambiare. In pratica il capitolo tratta tutto ciò che può essere gestito in modo asincrono al di fuori di quello che abbiamo chiamato *thread principale* o *thread della UI*. Il Capitolo 11 tratta un argomento che ha subito modifiche sostanziali in Marshmallow, ovvero la gestione dei permessi. Attraverso una prima gestione della Location per l'applicazione *ApoBus* vedremo infatti come gestire sia i permessi di tipo *normal*, sia quelli classificati come *dangerous*. Nella seconda parte del capitolo vedremo invece come eseguire richieste a un server. Anche in questo caso avremo due possibili soluzioni. La prima consiste nell'utilizzo del framework costruito attorno alla classe `URLConnection` fornita con la piattaforma, mentre la seconda consiste nell'utilizzo del framework *Volley*. Faremo uno studio approfondito delle due soluzioni, sviluppando con esse una nuova versione del servizio di ricerca delle fermate nella nostra applicazione.

Il Capitolo 12 è dedicato alla gestione delle animazioni o comunque di tutto ciò che si muove sul display. Nella prima parte vedremo che cosa sia un `ValueAnimator` e poi un `ObjectAnimator` per passare infine a descrivere i diversi modi di definizione e applicazione di un'animazione a `View` e `layout`. La gestione delle animazioni si è evoluta molto nelle ultime versioni della piattaforma, per cui il capitolo descriverà diversi modi per implementare funzionalità all'apparenza simili. La parte finale del capitolo è dedicata a una funzionalità molto importante, specialmente alla luce delle regole guida del *Material Design*, ovvero le `Transition`. Vedremo come gestire lo spostamento di `View` in varie situazioni, comprese quelle di passaggio da una `Activity` a un'altra o da un `Fragment` a un altro. Sebbene le animazioni non siano caratteristiche fondamentali di un'applicazione, rappresentano spesso un aspetto che fa la differenza e che fa preferire un'applicazione rispetto a un'altra.

Il Capitolo 13 è dedicato a tutto ciò che riguarda la gestione della `Location`, informazione fondamentale nella realizzazione della nostra applicazione *ApoBus*. Nella prima parte vedremo come utilizzare i *Google Play services* per reperire in modo ottimizzato l'informazione di `Location`. Nella seconda parte descriveremo invece un insieme di API di fondamentale importanza, ovvero le *Google Maps API*. Utilizzando come pretesto la nostra applicazione, ci occuperemo di tutte le funzionalità che ci permetteranno di visualizzare in una mappa un insieme di fermate vicine a una data posizione.

Ringraziamenti

La scrittura di un libro può essere paragonata a un viaggio molto lungo. Anche quando si affrontano argomenti consolidati si scopre sempre qualcosa di nuovo, che porta entusiasmo e voglia di continuare. Spesso però non mancano le difficoltà, legate al fatto che comunque ci sono anche altre cose da fare, le scadenze incombono e non sempre si ha l'ispirazione per descrivere le cose utilizzando un linguaggio che sia il più comprensibile possibile. Le difficoltà sono però proprio quello che rendono questa esperienza degna di essere vissuta. Scrivere un libro equivale a leggerne più di cento. Come tutte le avventure, sono comunque più belle se vissute in compagnia di persone che ti danno un sostegno, che sono poi quelle che devo ringraziare. Mi sto riferendo a mia moglie Cristina e alle mie figlie Alice e Marta che, dopo avermi seguito a Londra, mi hanno sostenuto anche questa volta. Infine ringrazio tutti coloro che mi hanno incoraggiato in questa avventura all'estero, che mi ha permesso di rinascere sotto tutti i punti di vista.