



NOTIFICHE

Programmazione di sistemi embedded

05/07/2017

Martini Davide, Nigro Simone

Tubaldo Alberto



SOMMARIO

Introduzione	2
1. Notifiche	3
1.1 Dettagli implementativi	3
1.2 Android Nougat	5
1.2.1 Novità sui dettagli implementativi	6
1.3 Android O	7
2. Il progetto	8
2.1 Applicazione Base	8
2.2 Applicazione Chat	9
3. Bibliografia	10

Introduzione

Una notifica è un messaggio che l'utente può visualizzare al di fuori dell'interfaccia normale dell'applicazione. Quando un sistema emette una notifica, essa viene visualizzata come una piccola icona in un'area dedicata, alla quale l'utente potrà accedere in qualsiasi momento.

Al giorno d'oggi quasi tutti i dispositivi informatici generano decine di notifiche al giorno, in particolare gli smartphone per diversi motivi emettono una notifica avvisando l'utente dell'arrivo di un messaggio, di una nuova email oppure di una novità pubblicata in un social network o da un servizio utilizzato. Tecnicamente queste informazioni vengono ricevute sotto forma di notifiche push.

Le persone sono sempre più abituate a risolvere i bisogni quotidiani attraverso le applicazioni e i dispositivi mobili (come prenotare un servizio, acquistare un prodotto online, cercare un ristorante, ecc). Per chi ha un'attività, avere una buona presenza sul mobile diventa strategico ed avere un'applicazione propria lo è ancora di più. Il marketing via app è sempre più efficiente grazie al grande potenziale che le notifiche push hanno. Esse infatti garantiscono l'invio di offerte e coupon a clienti interessati, oppure di informazioni in anteprima ai clienti più fedeli, ma anche molto altro. Fare un buon lavoro significa anche riuscire a segmentare la propria clientela secondo gli specifici interessi ed inviare loro informazioni coerenti con i loro bisogni. Le notifiche tengono il cliente sempre aggiornato sugli argomenti di interesse senza che lui debba sforzarsi di farlo da solo.

1. Notifiche

Le notifiche devono fornire informazioni utili in modo rapido ed efficace, per cui devono essere presentate all'utente con un'interfaccia grafica essenziale ed efficiente.

Una notifica può presentarsi in diversi modi che variano in base a che attività sta svolgendo l'utente e ad altri parametri che possono essere decisi dall'applicazione specifica o dal sistema Android. Essa può essere costituita anche solo da un'icona, un titolo o un testo e se espansa dall'utente può contenere azioni interattive, immagini o layout personalizzati. Si trova di base nel drawer delle notifiche che l'utente raggiunge mediante uno swipe verso il basso, ma potrebbe anche apparire nella parte superiore dello schermo in formato Heads-up se l'utente sta utilizzando un'altra app.

Android permette una personalizzazione abbastanza profonda per la realizzazione di una notifica, che consente di renderla adatta al contenuto che si vuole far giungere all'utente all'esterno della propria applicazione.

1.1 Dettagli implementativi

`Notification.Builder`: i metodi di questa classe consentono la costruzione e la personalizzazione di una notifica. La più semplice deve utilizzare obbligatoriamente:

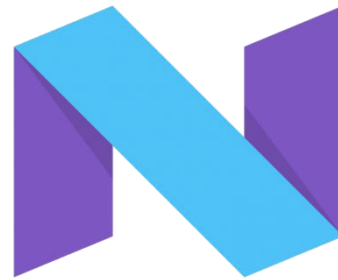
- ◆ `setSmallIcon`: si deve indicare la risorsa icona che si vuol far apparire nella notifica. Solitamente si sceglie l'icona dell'app poiché consente all'utente di riconoscere in modo immediato la provenienza della notifica.
- ◆ `setContentTitle`: imposta il titolo. E' la parte testuale più in evidenza e deve perciò contenere un'informazione identificativa sull'argomento che ha generato la notifica (potrebbe essere semplicemente il nome dell'applicazione).
- ◆ `setContentText`: imposta il testo che di solito descrive il contenuto della notifica.

Oltre a questi sono presenti altri metodi che possono essere utilizzati a seconda delle necessità:

- ◆ **addAction:** permette di aggiungere un'azione. Deve essere descritta da un testo e attraverso l'utilizzo di un `PendingIntent` consente di rimandare a una sezione specifica dell'app o altro (non verranno visualizzati su piattaforme precedenti a Android 4.1. I pulsanti di azione dipendono dalle notifiche espanse, disponibili solo in versione Android 4.1 e successive).
- ◆ **setPriority:** la priorità consente al sistema Android di scegliere se proporre all'utente la notifica in modo più o meno invasivo a seconda delle necessità. E' molto importante in quanto permette di notificare anche informazioni non indispensabili senza infastidire l'utente con inutili interruzioni.
- ◆ **setVibrate, setSound, setLights:** permette di personalizzare la vibrazione, la suoneria e il colore dei LED.
- ◆ **setAutoCancel:** consente di eliminare la notifica dopo un click su di essa oppure di restare attiva in caso la si voglia di tipo persistente.
- ◆ **setContent:** fornisce una `RemoteView` personalizzata da utilizzare al posto del modello di piattaforma.
- ◆ **build:** combina tutte le opzioni che sono state impostate e restituisce un nuovo oggetto `Notification`.

1.2 Android Nougat

L'obiettivo primario delle novità introdotte da Android N è quello di permettere all'utente di accedere il più semplicemente possibile alle informazioni fornite dagli elementi base della notifica. Fra questi il più importante è il contenuto primario, che occupa lo spazio maggiore all'interno del layout e possiede un carattere più grande rispetto alle informazioni secondarie (riportate sulla parte alta). Nel caso di app di messaggistica l'avatar del mittente viene posizionato sulla destra affiancando il contenuto principale. Quando la notifica viene espansa può presentare, nella parte inferiore, delle azioni che l'utente può effettuare, costituite da semplice testo su uno sfondo di colore differente rispetto a quello della parte principale della notifica per evidenziarne l'interattività. Alcune azioni possono essere completate senza dover aprire l'applicazione.



Nougat permette di formare un gruppo in presenza di più notifiche generate dalla stessa app. Queste vengono organizzate in modo gerarchico: il gruppo viene compresso in una notifica che mostra un riassunto dei contenuti; l'utente può espandere il gruppo visualizzando quindi le singole notifiche.

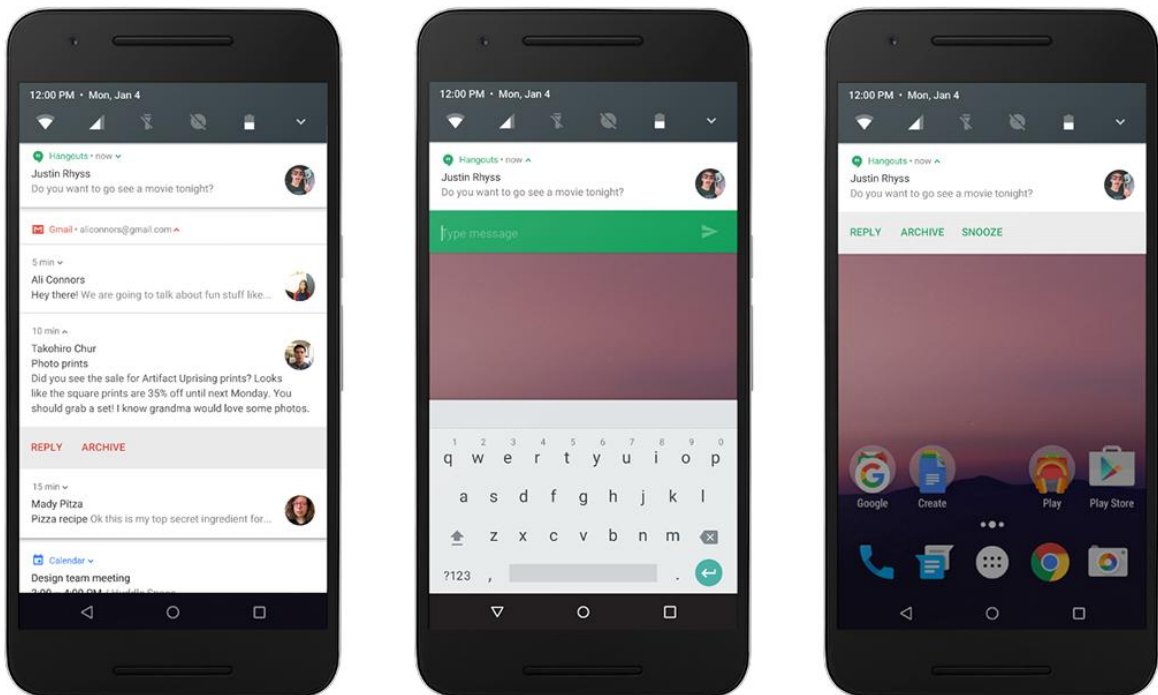
Una delle novità più importanti è quella di permettere all'utente di scrivere direttamente all'interno della notifica attraverso un'azione. Questa funzionalità è pensata per permettere una risposta rapida ed immediata in contesti come la messaggistica. Nel caso in cui sia necessario un inserimento di testo più lungo si deve comunque rimandare all'app in modo classico.

L'utente può accedere, mediante pressione prolungata sulla notifica, a una sezione in cui scegliere come notifiche simili a quella presentata devono essere mostrate in futuro: può scegliere di silenziarle o bloccarle, di conseguenza è necessario prestare molta attenzione al presentare sempre notifiche che si considerano di interesse per l'utente.

1.2.1 Novità sui dettagli implementativi

Alcune novità introdotte a livello implementativo rispetto le versioni precedenti riguardano:

- ◆ Azione di risposta: la classe `RemoteInput.Builder` consente di gestire l'input dell'utente e di identificarlo attraverso una label. Si associa poi l'istanza di questa classe a `Notification.Action` e si crea la notifica aggiungendo questa azione (tramite metodo del builder `addAction`). La risposta viene successivamente gestita con il metodo `getResultsFromIntent` di `RemoteInput`. La notifica resta in attesa che la gestione della risposta venga completata e a quel punto può essere eliminata.
- ◆ Notifiche raggruppate: con il metodo `setGroup` della classe `Notification.Builder` si sceglie una chiave che poi consente di raggruppare le notifiche per tipo.
- ◆ Custom Views: `Notification.DecoratedCustomViewStyle` consente di mantenere le decorazioni di sistema (header e azioni) per la notifica anche usando un layout personalizzato.



1.3 Android O

La nuova versione, prossimamente in uscita, molto probabilmente prenderà il nome di Android 8.0 Oreo. Essa porterà alcune modifiche rispetto Nougat: per quanto riguarda le notifiche introdurrà una nuova veste cromatica per la

tendina; dopo diversi anni con una colorazione grigio scura protagonista, Google ha deciso dunque di cambiare radicalmente l'aspetto di una delle parti più utilizzate del proprio sistema, implementando una colorazione differente. Inoltre è stato deciso di ridisegnare parte della UI cambiando la posizione di alcuni pulsanti.

Ma una delle novità più attese sarà la creazione dei Notification Channel, ovvero la possibilità di categorizzare le notifiche in gruppi personalizzando l'importanza, l'aspetto, il suono, il modo in cui appaiono e il colore dell'eventuale Led di notifica dello smartphone.

Per esempio un utente potrebbe avere su Twitter un account personale e un account aziendale: tramite i Notification Channel sarà possibile separare le notifiche dei due account per evitare confusione. Inoltre ci sarà la possibilità di impostare lo "snooze" per le notifiche: basterà un click per ricevere la notifica di una cosa importante in un momento successivo.

Un'altra novità a cui Google ha pensato sono i "Notification Dots", dei piccoli pallini che saranno visualizzati sopra le icone delle app che hanno ricevuto delle notifiche. Con una pressione prolungata sarà possibile estendere questi pallini ed avere un'anteprima delle notifiche in questione. Premendo su di esse, queste verranno aperte nella relativa app.



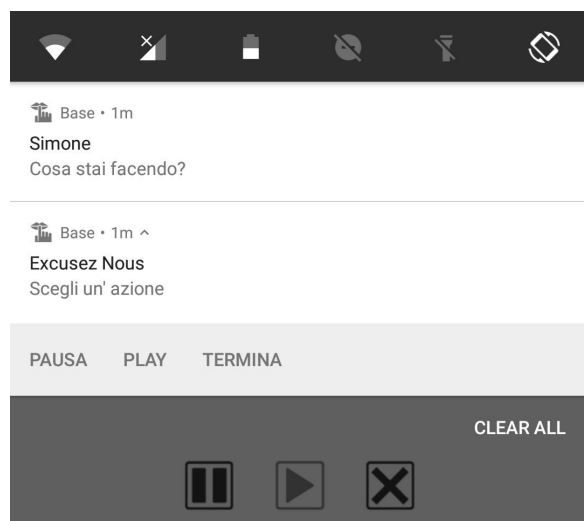
2. Il Progetto

Tenendo conto delle richieste, abbiamo deciso di dividere il progetto in due parti.

2.1 Applicazione Base

Il primo progetto sviluppato è stato un'app che periodicamente inviava notifiche. L'idea iniziale era quella di utilizzare un thread gestito dall'interfaccia utente che mandasse, ad intervalli regolari variabili tra i 5 ed i 10 secondi, notifiche con solo title, contentText e smallIcon, campi minimi obbligatori per la sua creazione. Dopo varie prove, notato il vistoso rallentamento dell'applicazione, abbiamo deciso di virare verso una soluzione differente, ossia l'utilizzo di un'entità che interagisce tra l'interfaccia utente e il thread, per ridurre il carico di lavoro di quest'ultimo. È stato quindi inserito un service che, attraverso i comandi dell'utente, attiva un processo per inviare la notifica desiderata. Questa scelta si è rivelata vincente poiché ha portato a prestazioni molto più fluide e ad una facilità di gestione del dialogo, da un lato tra service e thread e dall'altro lato con l'interfaccia utente. Durante la seconda fase di sviluppo, cercando tra la documentazione ufficiale le caratteristiche della classe Thread, ci siamo imbattuti nella classe AsyncTask e abbiamo deciso di utilizzarla per l'invio della notifica e l'attesa di qualche secondo tra una e l'altra, poiché ci siamo resi conto che è molto adatta alle nostre esigenze.

Un ulteriore miglioramento che abbiamo deciso di apportare all'applicazione è la possibilità di poterla utilizzare anche se l'interfaccia utente è chiusa. Per renderlo possibile abbiamo utilizzato una notifica che può interagire con le stesse potenzialità dell'interfaccia utente. I comandi verso il service si possono quindi utilizzare sia dalla notifica in questione, sia dalla schermata principale dell'applicazione. Inizialmente è sorto un problema di sincronizzazione tra le azioni

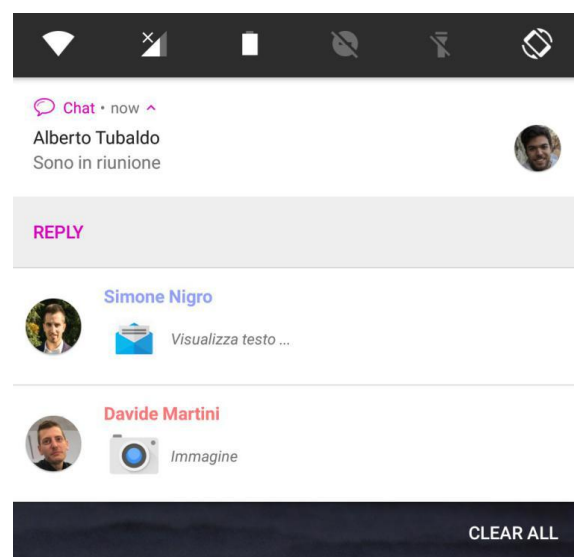


della notifica e i bottoni dell'interfaccia utente quando entrambe erano attive. Ci siamo trovati di fronte ad una scelta obbligata: i tre bottoni della MainActivity sono stati resi statici, per questo quando viene premuto un tasto dalla notifica che gestisce il service, viene cambiato anche lo stato dei bottoni dell'interfaccia utente. Nonostante la ricerca di soluzioni alternative, purtroppo questa si è rivelata l'unica che ci ha permesso di raggiungere il risultato richiesto pur portando ad una perdita di incapsulamento.

2.2 Applicazione Chat

Il secondo lavoro sviluppato è centrato sulle potenzialità e le novità introdotte da Android 7.0 in ambito di notifiche. La nostra idea è stata quella di sviluppare un'applicazione che simula lo scambio di messaggi tra un mittente e un destinatario, scelto da una lista di contatti disponibili. Una conversazione ha inizio con l'invio di un messaggio da parte dell'utente che utilizza l'applicazione. Il mittente, dopo

l'attesa di alcuni secondi, riceverà una risposta con un contenuto casuale, composto da un testo oppure da un'immagine; tale scambio di messaggi avrà fine nel momento in cui l'utente deciderà di non dare più risposta. La mancanza di senso logico nel dialogo è dovuta alla scelta di una maggior concentrazione sull'aspetto estetico e tecnico della notifica in se. La nostra prima idea è stata quella di utilizzare un service che tramite intent gestisse le risposte provenienti da conversazioni differenti. E' sorto però un problema di interferenza tra i vari intent in quanto, l'utilizzo del metodo getIntent ci consentiva di poter continuare a interagire solo con il destinatario contattato per ultimo. Consultando il sample MessagingService, messo a disposizione da Android, abbiamo deciso di implementare un sistema di BroadcastReceiver, grazie al quale siamo riusciti a gestire più conversazioni contemporaneamente, risolvendo il nostro problema. Giunti a questo punto ci siamo concentrati sul lato concreto del progetto, ossia l'implementazione delle novità di Nougat.



Queste consistono nella possibilità di rispondere ai messaggi direttamente dall'area di notifica e di personalizzarne a proprio piacimento la grafica e la vibrazione differenziando così i vari contatti. Per far ciò abbiamo usato i RemoteViews che ci hanno permesso di avere un'anteprima del messaggio differente a seconda che si tratti di un'immagine o di un testo.

3 Bibliografia

Principali fonti e strumenti utilizzati per la realizzazione dell'applicazione:

- ◆ <https://developer.android.com/samples/MessagingService/index.html>
- ◆ <https://developer.android.com/training/implementing-navigation/nav-drawer.html>
- ◆ <http://stacktips.com/tutorials/android/android-seekbar-example>
- ◆ <http://stackoverflow.com/questions/15326290/get-android-seekbar-value-and-display-it-on-screen>
- ◆ <https://developer.android.com/reference/android/widget/RemoteViews.html>
- ◆ <https://developer.android.com/guide/topics/ui/notifiers/notifications.html>
- ◆ <https://developer.android.com/reference/android/app/Notification.html>