



I primi passi nello sviluppo con Android

Android è il secondo sistema operativo mobile per diffusione, e continua a crescere, alimentato da telefoni e tablet in uscita con questa versione di Linux.

Android nasce nel 2003 in una startup di quattro persone: Andy Rubin, Rich Miner, Nick Sears e Chris White. L'obiettivo dei soci era realizzare un sistema operativo per telefoni cellulari, basato su Linux. Lo stesso obiettivo era alla base di diversi altri sviluppi industriali di Intel e Nokia, che hanno dato origine a Maemo e Moblin. Una delle scelte caratterizzanti dell'architettura di Android ha riguardato il linguaggio per la realizzazione di applicazioni. Android lascia da parte il C di Linux per adottare Java con una particolarità: l'ambiente Java non gira su una macchina virtuale standard, come J2ME, ma su una macchina virtuale proprietaria chiamata Dalvik, dal nome del piccolo paese sulle coste della Groenlandia in cui è nato il suo autore Dan

Borstein. Fra Groenlandia e Finlandia si conferma il legame di Linux con i climi freddi.

Nel 2005 la società fu acquisita da Google, che nel 2007 diede origine alla Open Headset Alliance, una alleanza industriale fra produttori di telefoni cellulari, ben disposti verso un sistema operativo gratuito con licenza Apache. Da allora parecchia acqua è passata sotto i ponti e le release, che hanno nomi di dolci e seguono l'ordine alfabetico, sono arrivate a Jelly Bean, partendo da Apple Pie, Cupcake e Donut (manca la B nella serie, la seconda release era Petit Four).

Oggi la quota di mercato di Android viene stimata al 28,45%, secondo i dati di netmarketshare.com, ben distante dal 61,4% di iPhone e iPad, ma comunque molto interessante, considerando

anche che il mercato dei tablet ha visto solo di recente Kindle, Nexus e Galaxy pronti per fare seriamente buoni numeri senza invidiare più di tanto i cugini Apple e forti di un prezzo più che concorrenziale.

Vediamo, quindi, che cosa occorre per muovere i primi passi nello sviluppo per Android.

L'ambiente di sviluppo

La pagina di riferimento per lo sviluppo con il sistema operativo mobile di Google è developer.android.com e il percorso per i novizi inizia cliccando *Develop* e quindi *Training*, nella toolbar.

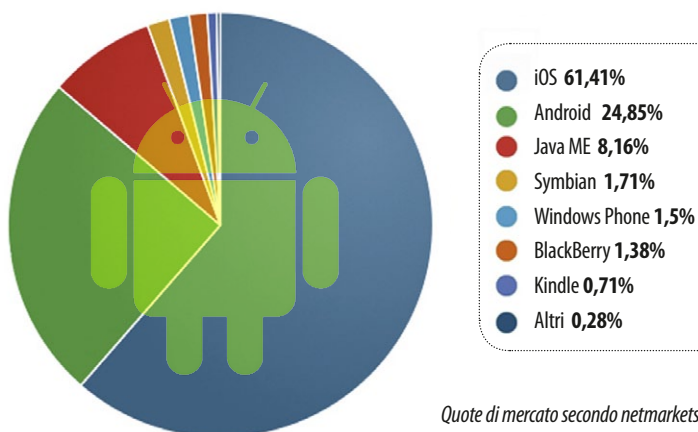
I più frettolosi, possono fare un clic su *Get the Sdk* nella barra in basso. Un vistoso tasto blu permette di scaricare immediatamente l'ambiente di sviluppo Android e l'ambiente Eclipse necessario per utilizzarlo, chiamato *Adt* (Android Developer Tools).

Dopo avere espanso l'archivio che contiene *Adt* troviamo due directory principali, una si chiama *eclipse* e contiene l'ambiente di sviluppo, già configurato con i plugin necessari per lo sviluppo, nell'altra, *sdk*, c'è una versione delle librerie di supporto e dell'emulatore Android, limitata alla versione corrente del sistema operativo.

Un clic sull'icona di Eclipse (in questa edizione due parentesi graffe), un controllo all'impostazione relativa alla posizione dello Sdk nelle preferenze, che deve puntare alla cartella *sdk*, e siamo pronti per partire. Troviamo subito la voce *Android Application Project* nel menu *File*, alla voce *New*. Lasciando tutti i default impostati, ci troviamo in editing visuale dell'interfaccia utente della nostra applicazione.

Scegliendo *Android Sample Project* dal gruppo di template Android, si può

QUOTA DI MERCATO DI ANDROID



Quote di mercato secondo netmarketshare.com.

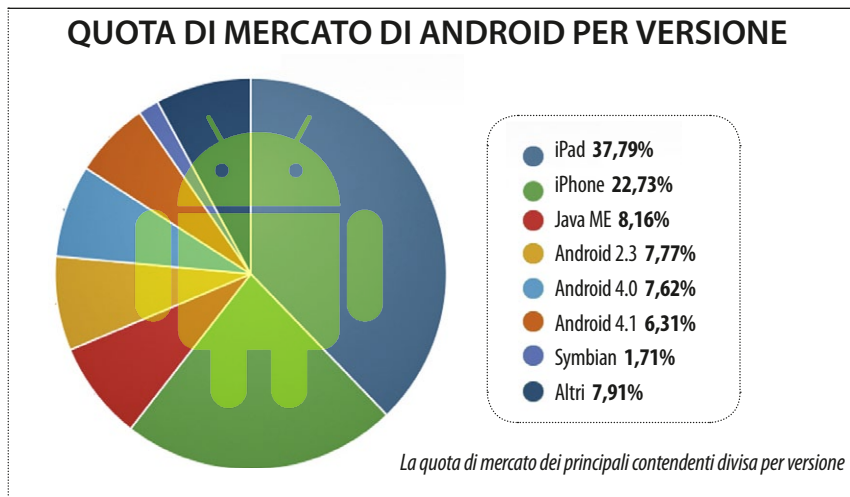
copiare dallo Sdk nell'ambiente di lavoro Eclipse uno dei tre progetti campione. La configurazione di Eclipse dello Adt non si limita a un'installazione Java ridotta all'osso, troviamo infatti il supporto per lo sviluppo Web e Xml, il server Jetty, i tool Apache, come Lucene e l'ambiente di sviluppo C e C++, che può essere utile per eventuali sviluppi nativi (interesserà agli sviluppatori più hard core, per esempio i creatori di giochi interattivi).

L'installazione dello Adt è semplice e rapida e permette a chi vuole iniziare di avere tutto a disposizione fin da subito, senza presupporre un ambiente Eclipse. È un'opzione utile per chi non vuole fare confusione con una configurazione Eclipse attentamente mirata e pesata.

Se si preferisce estendere un ambiente Eclipse già esistente con la capacità di creare applicazioni Android, occorre installare il plugin, seguendo con precisione le istruzioni alla pagina developer.android.com/sdk/installing/installing-adt.html.

Il plugin contiene quello che serve a Eclipse per supportare lo sviluppo, ma le librerie Java, gli strumenti, i progetti di esempio e gli emulatori con le relative immagini sono nello Sdk. Occorre quindi installare lo Sdk, scaricandolo dalla pagina developer.android.com/sdk/index.html.

L'installazione dello Sdk completo consente di sviluppare e testare un'applicazione anche con versioni precedenti della piattaforma Android, una necessità per lo sviluppatore professionista,



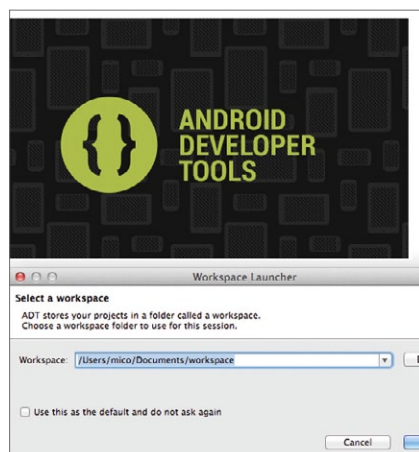
che cerca di supportare il maggior numero di telefoni.

Esiste, infine, un'ultimo componente scaricabile, il *Native Developer Kit*, che serve a chi desidera sviluppare in codice C applicazioni native. Ricordiamo, infatti, che Android è una piattaforma Linux, quindi non tutto il software è destinato a girare all'interno della macchina virtuale e esistono alcune applicazioni che possono guadagnare da una implementazione nativa di determinate componenti. Questo genere di sviluppo è molto specialistico ed è di beneficio solo per applicazioni in cui la componente algoritmica è predominante. Può essere una scelta obbligata quando si hanno librerie in C da riciclare. In tutti gli altri casi, lo Ndk si può tranquillamente ignorare. Un'applicazione in parte nativa viene segmentata in un'applicazione Java

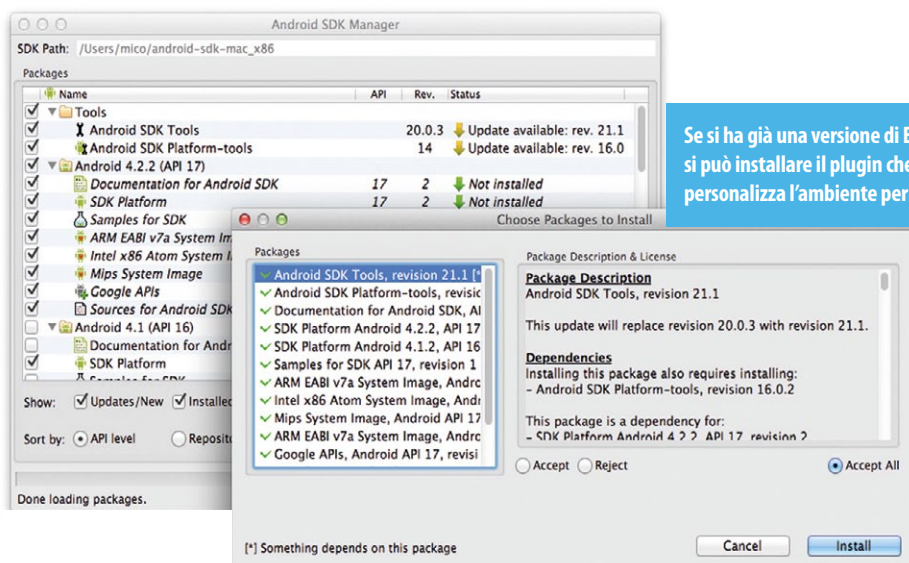
ospite e un insieme di librerie C a cui si accede attraverso la Java Native Interface (Jni). La documentazione sul sito per sviluppatori è abbastanza chiara ed efficace.

Ricordiamo anche che si può sviluppare per Android usando Netbeans o qualche altro ambiente di sviluppo popolare per Java, come IntelliJ Idea, che ha un supporto specifico per Android, e comprende il disegno visuale di interfacce utente. Ulteriori dettagli si possono trovare su www.jetbrains.com/idea/, mentre il plugin per Netbeans si trova a questo indirizzo plugins.netbeans.org/plugin/19545/.

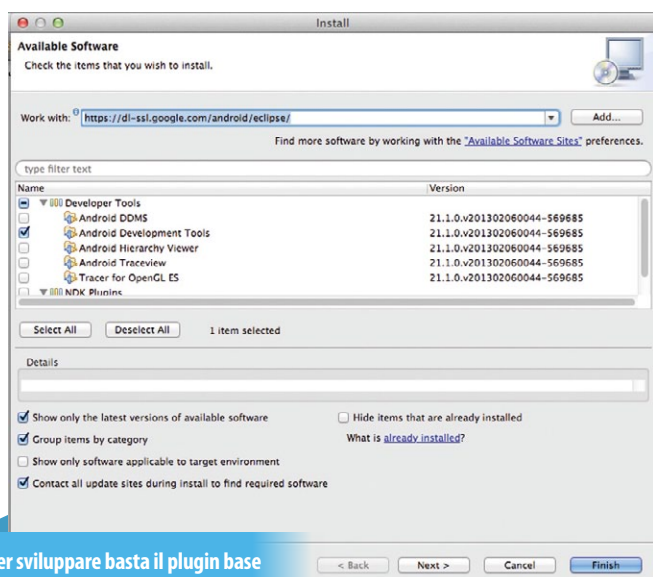
Ricordiamo, comunque, che la strada più lineare per iniziare da zero per creare un'applicazione è scaricare Adt, che pesa poco più di mezzo Gbyte, senza preoccuparsi di configurazioni e dipendenze.



Lo schermo di avvio del pacchetto Adt, che comprende Eclipse e lo Sdk Android Eclipse supporta diversi workspace, può essere conveniente crearne uno specifico per lo sviluppo mobile



Se si ha già una versione di Eclipse, si può installare il plugin che personalizza l'ambiente per Android.



Per sviluppare basta il plugin base

I linguaggi

Come abbiamo anticipato nella sezione precedente, Java è il linguaggio d'elezione per la creazione di applicazioni Android, ma si può utilizzare il C, per ottimizzare sezioni di codice che sarebbero pesanti in Java o per riutilizzare librerie C già sviluppate.

L'ambiente Java di Android non è quello disegnato a suo tempo da Sun per le versioni mobili della macchina virtuale ufficiale J2ME. Le librerie Android sono specifiche per la piattaforma e la macchina virtuale Dalvik non è quella ufficiale. D'altra parte, confrontando il numero di terminali Android con la massa di applicazioni J2ME, è arduo sostenere che il primo non sia lo standard.

Rimane lo svantaggio, per lo sviluppatore Java navigato, di dovere considerare comunque una curva di apprendimento per padroneggiare la piattaforma. Per iniziare a

mentre i *Service* sono componenti che eseguono operazioni destinate a durare in background, anche quando l'applicazione perde il focus, per esempio la riproduzione audio o la navigazione. Il ciclo di vita di un'applicazione su Android è simile a quello di altri sistemi operativi per telefoni: le applicazioni non attive non mantengono costantemente bloccate le loro risorse. Un'applicazione non attiva è in pausa, successivamente può essere fermata e, se occorre, distrutta per recuperare risorse. A ogni passaggio c'è una progressiva liberazione di risorse. Il sistema operativo invia opportuni messaggi alle applicazioni per permettere loro di reinizializzarsi in modo corretto quando l'utente le riattiva, per esempio, navigando indietro.

Una caratteristica di Android è la notevole frammentazione della piattaforma, con dimensioni dello schermo piuttosto variabili fra i molti terminali

familiarizzare con i nuovi package, il sito per sviluppatori mette a disposizione tutorial per aiutare chi viene da fuori (*Api Guides*) e una sezione *Reference*, tarata per non far perdere tempo a chi ha già iniziato a orientarsi.

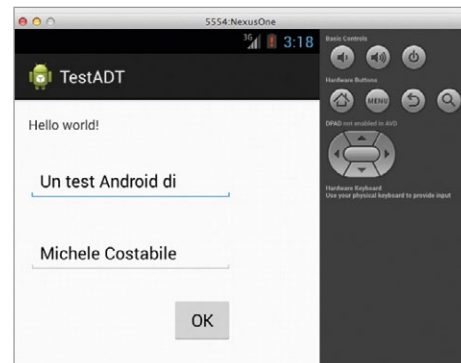
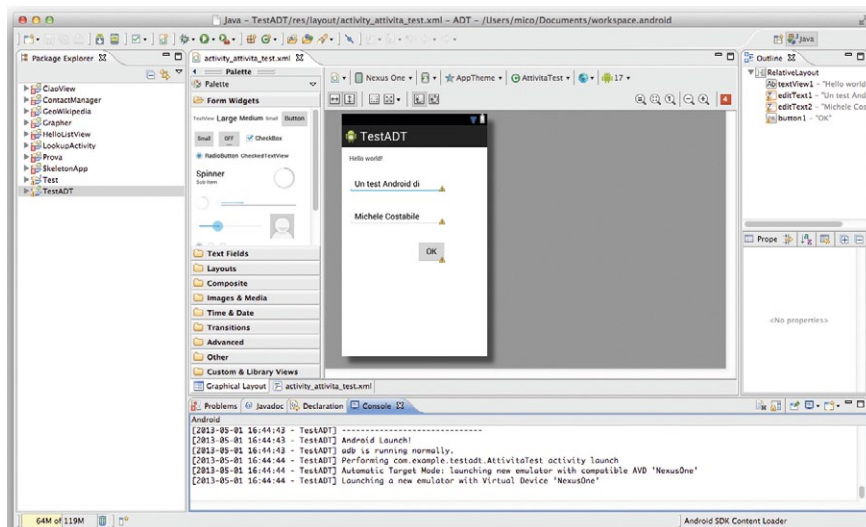
Per comprendere le prime indicazioni, ricordiamo che una *Activity* è una componente dell'applicazione a cui corrisponde uno schermo di interfaccia utente,

in circolazione, per non parlare dei tablet. L'ambiente di sviluppo permette di creare una lista di device virtuali per il debugging dell'applicazione. L'ambiente di sviluppo Eclipse permette di fare il debugging con dispositivi virtuali con diverse specifiche per lo schermo, la memoria e il processore. Naturalmente si può, e conviene, fare il debugging con terminali collegati via Usb, sicuramente la strategia di debugging che dà le sensazioni più accurate riguardo a usabilità e prestazioni.

Iniziare a vendere

Per arrivare con i propri programmi nello store Google Google Play, bisogna aprire la developer console all'indirizzo play.google.com/apps/publish/ e registrarsi. Alla registrazione bisogna pagare una quota di 25 dollari attraverso Google Wallet. Se si distribuiscono applicazioni a pagamento occorre registrarsi come merchant in Google Wallet in modo da attivare un conto su cui possono fluire i proventi dell'applicazione.

Si può scegliere con precisione in quali paesi distribuire l'applicazione e anche adottare una politica di prezzi specifica per ogni paese, in modo, per esempio, da avvicinare i prezzi a soglie psicologiche quasi intere, come 0,99. Il negozio Google supporta anche il pagamento all'interno dell'applicazione, una forma popolare, per esempio per l'acquisto di espansioni a un pacchetto base. Esistono linee guida per lo sviluppo di applicazioni (<http://developer.android.com/distribute/googleplay/quality/core.html>), ma non sono così strettamente applicate come sul negozio Apple. •



La nostra applicazione di prova in esecuzione in un emulatore con le caratteristiche di un Nexus One ▲

◀ Ecco il nostro progetto aperto nel designer visuale di interfaccia utente di Eclipse