

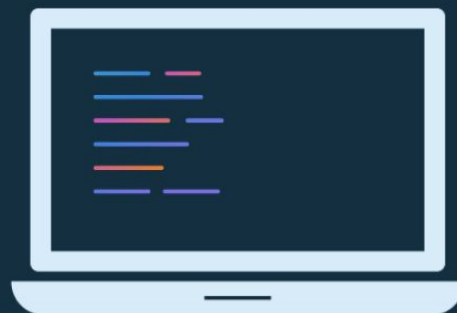


Lezione 2.0

Introduzione ad Android



[Paranoid Android - Radiohead](#)



Derivato da:
Google Developers Training

Programmazione mobile

This work is licensed under a [Creative Commons Attribution 4.0 International License](#).



Questa lezione



- L'ecosistema Android
- Architettura della piattaforma Android
- Versioni di Android
- Anatomia di un'app
- Sfide per lo sviluppo di app su Android

L'ecosistema Android

Cos'è Android?

- Sistema operativo mobile basato sul [kernel Linux](#)
- Interfaccia utente basata su schermi touch
- Usato su oltre il 70% degli smartphone
- Utilizzato anche su device come watches, TV, ed auto
- Oltre 2 milioni di app nel Google Play store
- Altamente personalizzabile per i device dai produttori
- Open source

Interazione utente

- Gesti touch: swiping, tapping, pinching
- Virtual keyboard per caratteri, numeri, ed emoji
- Supporto per Bluetooth, controller USB e periferiche

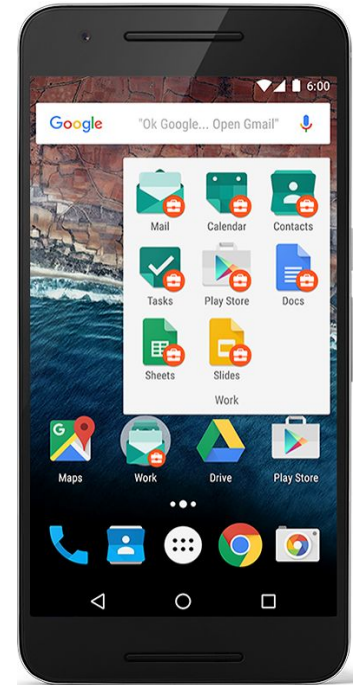
Android ed i sensori

Sensori possono interpretare le azioni dell'utente e rispondere di conseguenza, ad esempio:

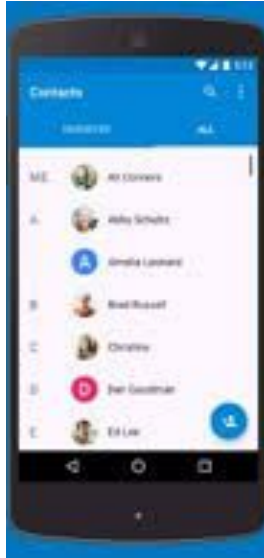
- Il contenuto può ruotare assieme al dispositivo
- La posizione su una mappa viene adattata durante la camminata
- L'inclinazione del dispositivo può far sterzare un'auto virtuale o controllare un dispositivo fisico
- Movimenti troppo rapidi disabilitano l'interazione nei giochi

Android home screen

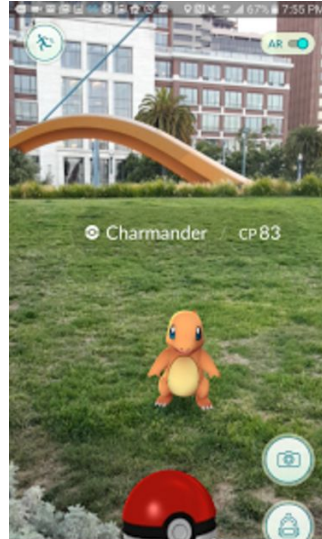
- Icone di lancio di app
- Self-updating widgets per contenuti aggiornati
- Può includere pagine multiple
- Cartelle per organizzare le app
- Funzioni integrate di ricerca ("OK Google")



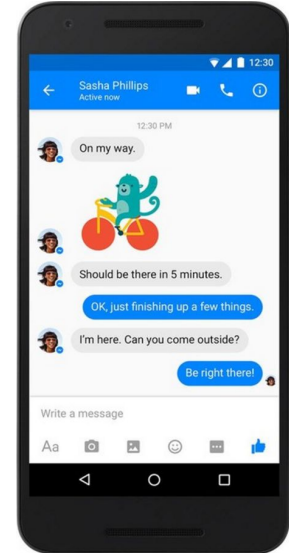
Android app: esempi



Rubrica



Pokemon GO

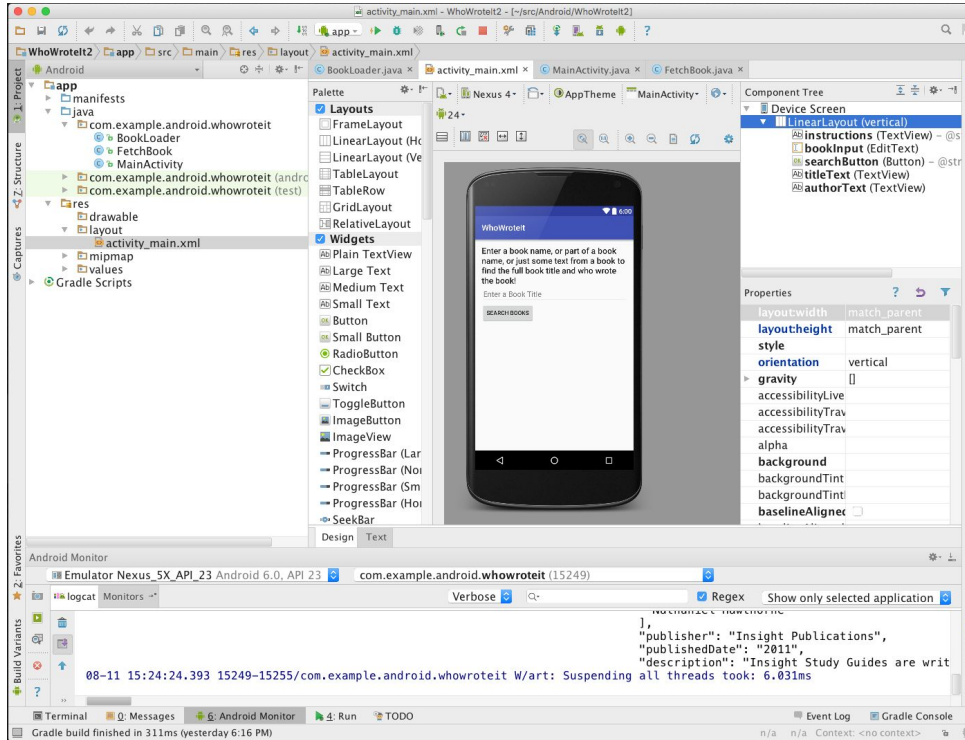


Facebook
Messenger

Android Software Developer Kit (SDK)

- Strumenti di sviluppo (debugger, monitor, editor)
- Librerie (gestione di mappe, gestione di wearable devices)
- Virtual devices (emulatori)
- Documentazione (developers.android.com)
- Codice di esempio

Android Studio



- Official Android IDE
- Develop, run, debug, test, and package apps
- Monitors and performance tools
- Virtual devices
- Project views
- Visual layout editor

Google Play store

Pubblicazione di app tramite il [Google Play](#) store:

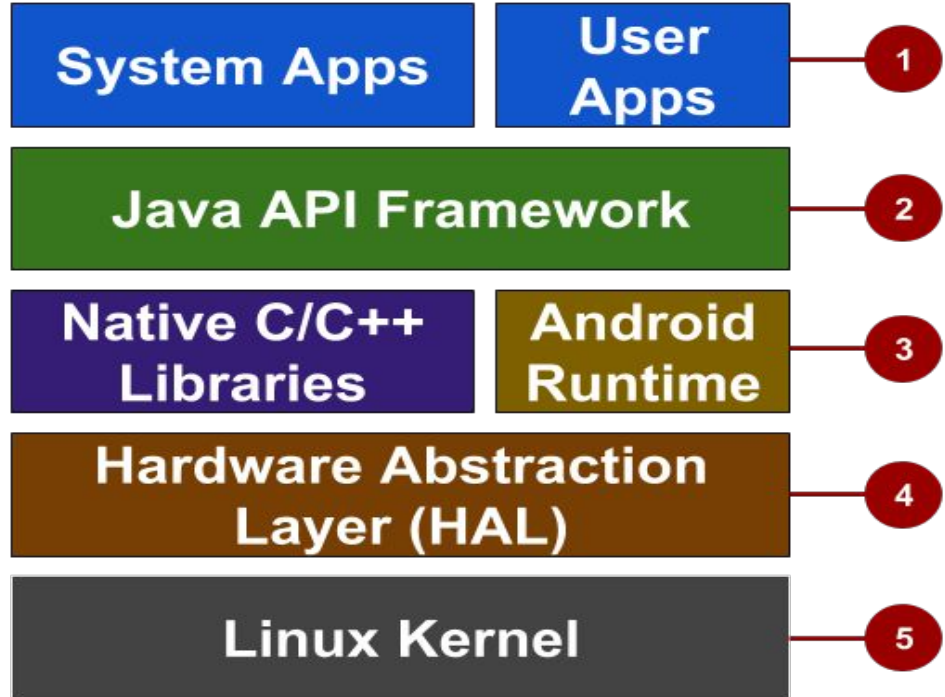
- App store ufficiale per Android
- Servizio di distribuzione ufficiale gestito da Google



Architettura della piattaforma Android

Android stack

1. App di sistema ed app utente
2. Android OS API in Java e Kotlin
3. Espone APIs native; esegue le app
4. Espone funzionalità hardware del device
5. Linux Kernel



App di sistema ed utente

- Le app di sistema non hanno uno status speciale
- forniscono funzionalità chiave agli sviluppatori

Esempio: La nostra app può usare un'app di sistema per inviare un SMS o visualizzare una pagina web

Java/Kotlin API Framework

Tutte le funzionalità di Android OS sono disponibili tramite API scritte in Java e Kotlin

- Gerarchia di classi per costruire l'interfaccia grafica di un'app
- Gestione delle notifiche
- Gestore del ciclo di vita delle attività e la navigazione tra di esse

Android Runtime

Ogni app viene eseguita come processo separato all'interno di una istanza dell'Android RunTime.

Librerie C/C++

- Un insieme di librerie C/C++ forniscono accesso alle componenti e servizi nativi del sistema Android (es: gestione dati e dispositivi, comunicazione, grafica,...)

Hardware Abstraction Layer (HAL)

- Interfacce standard che espongono funzionalità dei dispositivi hardware come librerie

Esempi: fotocamera, GPS, modulo bluetooth

Linux Kernel

- Gestione di basso livello dei processi, dei thread e della memoria
- Funzionalità di sicurezza
- Driver

Versioni di Android

Versioni di Android



Codename	Version	Released
-	1.0	2008
Petit Four, Cupcake	1.1, 1.5	2009
Donut, Eclair	1.6, 2.0	2009
Froyo	2.2	2010
Gingerbread	2.3	2010

Dal 2009 iniziano
i nomi di dessert



Versioni di Android



Codename	Version	Released	API Level
<i>Honeycomb</i> (la prima anche su tablet)	3.0 - 3.2.6	Feb 2011	11 - 13
<i>Ice Cream Sandwich</i>	4.0 - 4.0.4	Oct 2011	14 - 15
<i>Jelly Bean</i>	4.1 - 4.3.1	July 2012	16 - 18
<i>KitKat</i>	4.4 - 4.4.4	Oct 2013	19 - 20
<i>Lollipop</i> (la prima a 64 bit)	5.0 - 5.1.1	Nov 2014	21 - 22

Versioni di Android



Codename	Version	Released	API Level
<i>Marshmallow</i>	6.0 - 6.0.1	Oct 2015	23
<i>Nougat</i>	7.0 - 7.1	Sept 2016	24 - 25
<i>Oreo</i>	8.0 - 8.1	Sept 2017	26 - 27
<i>Pie</i>	9.0	Aug 2018	28

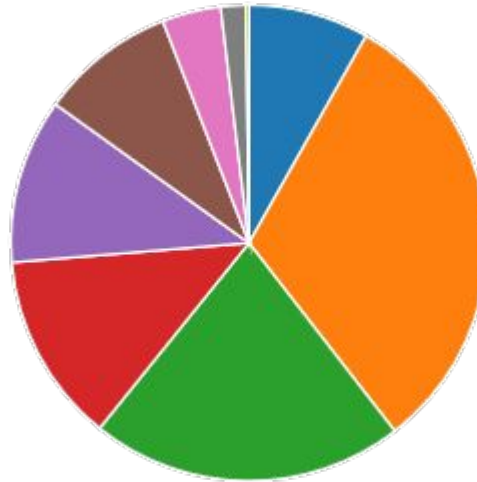
Versioni di Android



Codename	Version	Released	API Level
Android 10 Quince tart	10.0	Sep 2019	29
Android 11 Red velvet cake	11.0	Sep 2020	30
Android 12 Snow cone	12.0	Oct 2021	31 - 32*
Android 13 Tiramisù	13.0	*	33*

Per approfondire: https://it.wikipedia.org/wiki/Versioni_di_Android

Versioni di Android



Diffusione Android (10 aprile 2020)

Android 10	(8,2%)
Pie	(31,3%)
Oreo	(21,3%)
Nougat	(12,9%)
Marshmallow	(11,2%)
Lollipop	(9,2%)
KitKat	(4%)
JellyBean	(1,7%)
Ice Cream Sandwich	(0,2%)

Versioni di Android



Android 13 Developer Preview

An early look at the next version of Android, for testing, development, and feedback. Try it today on a supported Google Pixel device!

Get started



Developer Preview 1

Developer Preview 1 is now available, with the latest features and changes to try with your apps. Install a system image and update the tools to get started. During this phase we're looking for your feedback, so please let us know what you think! Visit the [feedback page](#) to report an issue or submit a feature request. The earlier we get your feedback, the more we can include in the final release.



Program overview

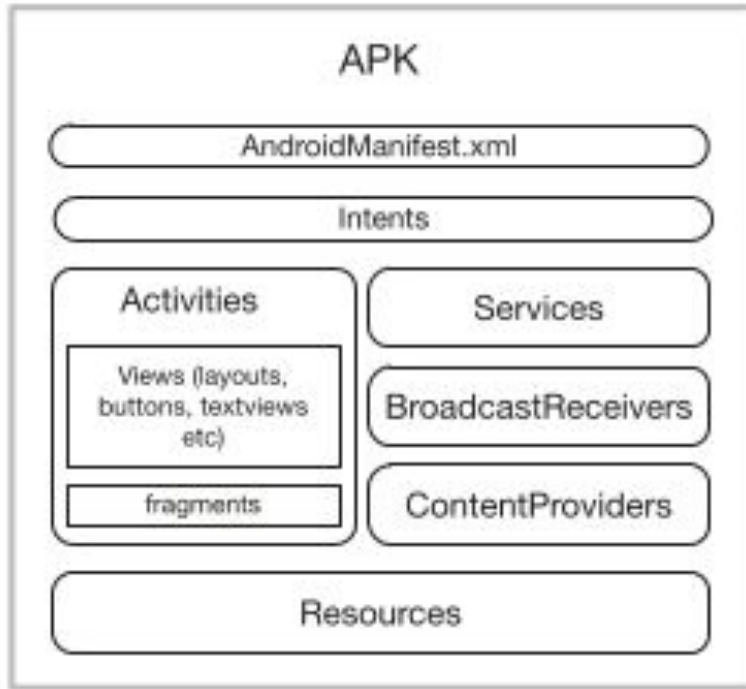
[\[Link\]](#)

Anatomia di un'app

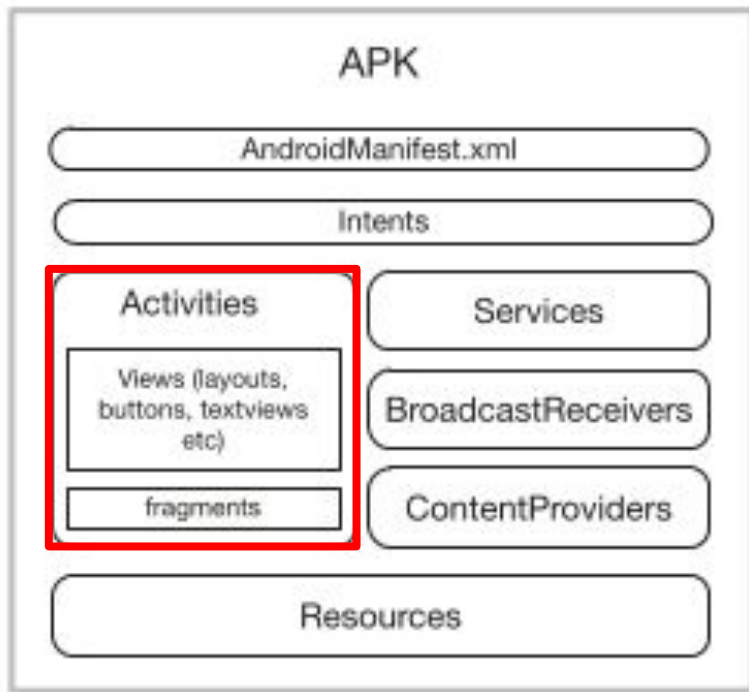
Cos'è un'app Android?

- Un insieme di una o più schermate interattive
 - Scritta usando Kotlin/Java e XML
 - Usa l'Android Software Development Kit (SDK)
 - Usa librerie Android e l'Android Application Framework
 - Viene eseguita dall'Android Runtime virtual machine (ART)

Cosa c'è in un'app?

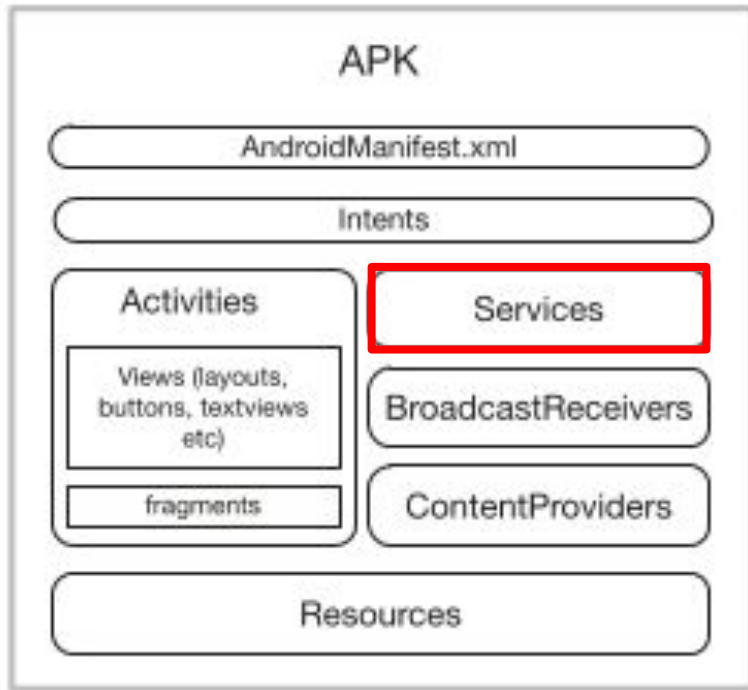


Cosa c'è in un'app?



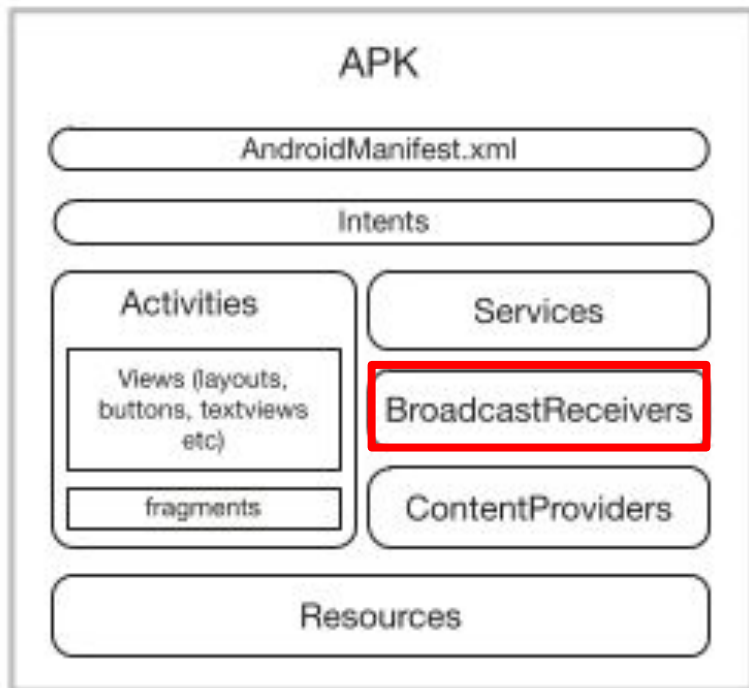
- **Activity:** la componente che utilizziamo per costruire ciò che l'utente vede e che può utilizzare (ad esempio la schermata che mostra una mail)
- Contiene Views (per mostrare contenuto a schermo) e Fragments (simile ad un'Activity ma più piccola)

Cosa c'è in un'app?



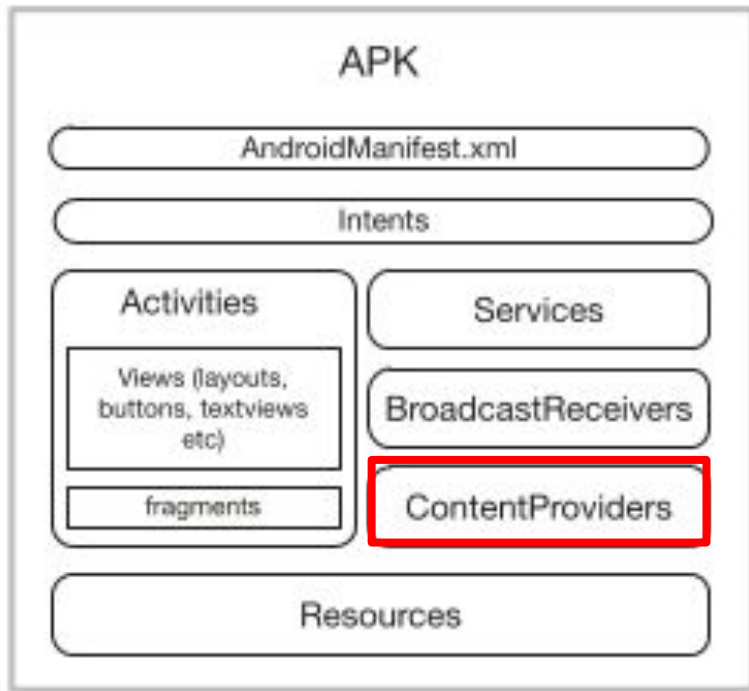
- **Service:** una classe che consente di eseguire istruzioni in background (ad esempio per effettuare il download di un file da Internet)

Cosa c'è in un'app?



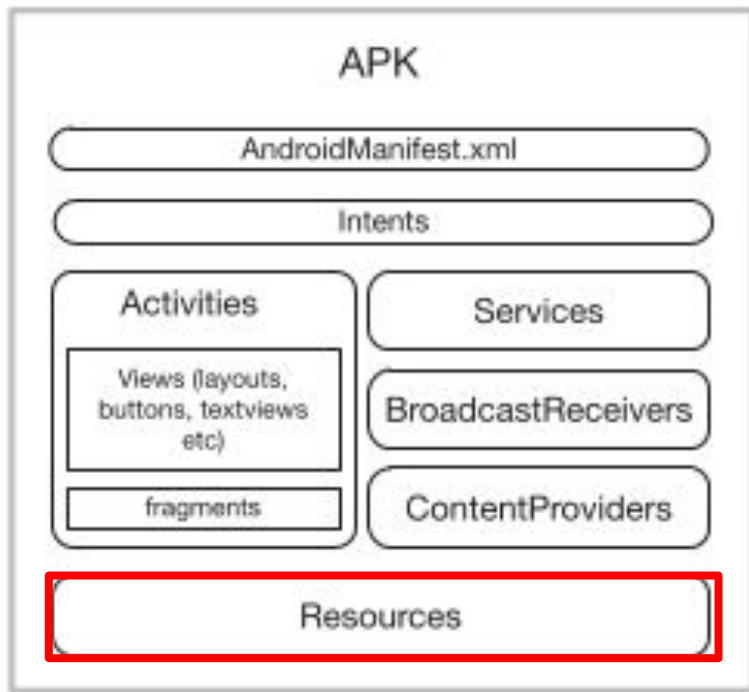
- **BroadcastReceiver:** consente all'applicazione di ricevere messaggi dal runtime Android oppure da altre applicazioni (ad esempio utile per mostrare un messaggio quando la batteria scende al di sotto del 10%)

Cosa c'è in un'app?



- **ContentProvider**: utilizzato per applicazioni che scambiano dati con altre, gestisce l'accesso ad una sorta di repository centralizzato
- Utile per consentire alle app di accedere ai dati di altre senza dover eseguire query SQL (un esempio è l'app "Contacts")

Cosa c'è in un'app?



- **Resources:** l'insieme dei contenuti statici o aggiuntivi di cui un'app ha bisogno per funzionare:
 - File di layout
 - Immagini
 - File audio
 - Stringhe usate nell'interfaccia
 - Icona dell'app

Cosa c'è in un'app?



- **AndroidManifest**: un file manifesto in XML, per dichiarare una serie di proprietà:

- Il nome dell'app
- quali Activity esistono e quale è l'entry point dell'app
- altre componenti
- cosa può fare (permessi)
- Uso di librerie esterne
- versione di Android

Cosa c'è in un'app?

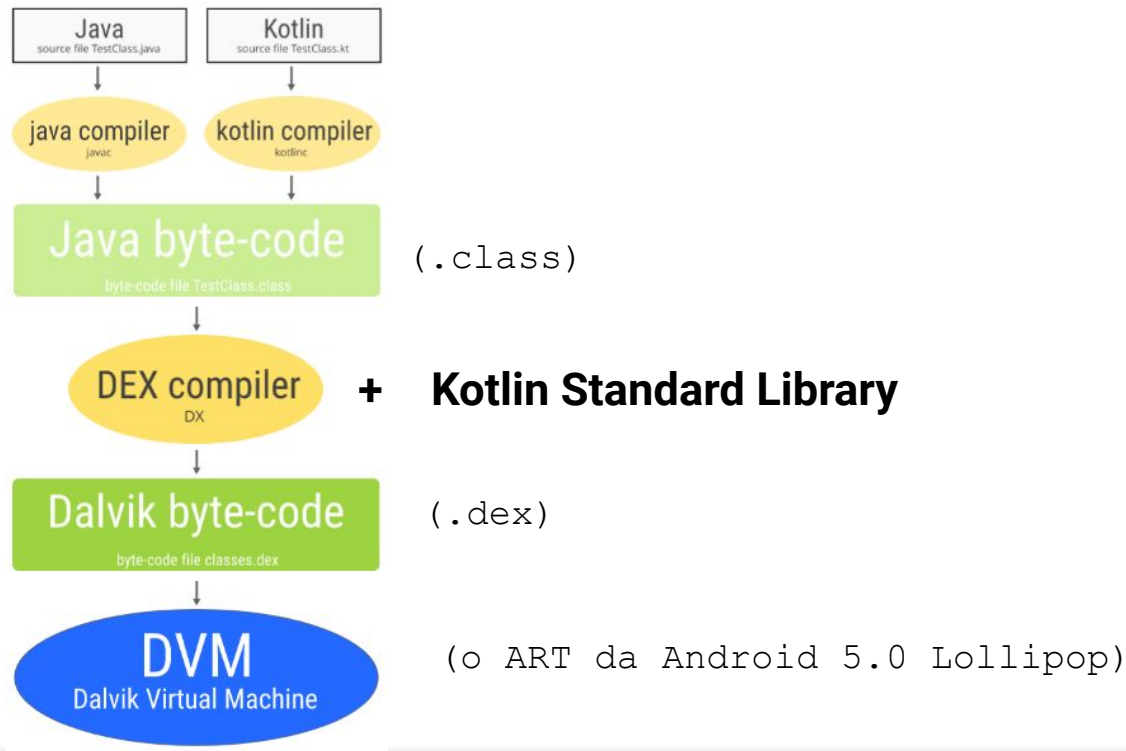


- **Intents:** un meccanismo per attivare altre activity all'interno dell'app oppure app esterne

Cosa c'è in un'app? In sintesi

- Approccio *loose-coupling*: un'app è un' <<insieme di componenti>> tenute insieme da un file manifest
- Interattività *user-centric*: ciascuna componente può essere attivata inviandole un messaggio
- L'intera piattaforma è sviluppata in modo che ciascuna componente possa interagire con molte altre, senza che sia stato previsto a priori
 - es: un'app può usare l'app Instagram o Telegram per condividere un contenuto senza conoscerne i dettagli di funzionamento

Compilazione di un'app (Kotlin version)



Compilazione di un'app (Kotlin version)

- Android non lavora con la JVM (Java Virtual Machine)
- Due VM progettate per Android:
 - DVM: Dalvik Virtual Machine
 - ART: Android Run Time

Compilazione di un'app (Kotlin version)

- Dalvik Virtual Machine:
 - ottimizzata per sfruttare la poca memoria dei dispositivi
 - consente di far girare diverse istanze della macchina virtuale contemporaneamente, nascondendo al SO la gestione della memoria e dei thread
 - Compilazione JIT (Just In Time), o traduzione dinamica: effettuata durante l'esecuzione (buon compromesso tra portabilità e velocità di compilazione)

Compilazione di un'app (Kotlin version)

- Android RunTime:
 - Runtime system introdotto in Android 4.4 che sostituisce DVM da Android 5.0
 - Approccio AOT (Ahead Of Time): l'intera compilazione avviene durante l'installazione, con vantaggi in termini di prestazioni e gestione delle risorse (con un leggero incremento di tempi di installazione)

Compilazione di un'app (Kotlin version)

- Perché Android non usa JVM?
- Risposta breve: perché lo sviluppo mobile ha requisiti di sicurezza, performance e affidabilità differenti

Compilazione di un'app (Kotlin version)

DVM	JVM
<ul style="list-style-type: none">• È basata su registri e progettata per operare con poca memoria a disposizione• Usa il suo proprio byte-code ed esegue file “.dex”. Da Android 2.2 Dalvik ha incorporato un suo proprio compilatore JIT.• È stata progettata in modo da far girare istanze multiple di VM in modo efficiente. Ogni applicazione è eseguita su una VM diversa.• Supporta solo il sistema operativo Android.• Esiste un constant pool per ogni applicazione.• L'eseguibile è il APK.	<ul style="list-style-type: none">• È basata su stack• Usa Java byte-code ed esegue file “.class” su un compilatore JIT• Una singola istanza di JVM è condivisa tra le applicazioni.• Supporta diversi sistemi operativi.• Esiste un <u>constant pool</u> per ciascuna classe.• L'eseguibile è il JAR.

Sfide nello sviluppo di un'app

- Schermi di dimensioni e risoluzione diverse
- Performance: rendere l'esecuzione reattiva e fluida
- Sicurezza del codice e dei dati
- Compatibilità: eseguibile anche su versioni più vecchie
- Marketing: comprendere il mercato e gli utenti

Approfondimenti

- [La storia di Android](#)
- [Introduzione ad Android](#)
- [Architettura della piattaforma](#)
- [UI Overview](#)
- [Versioni della piattaforma](#)
- [Supporto a differenti versioni della piattaforma](#)
- [Android Studio User's Guide](#)