

# Introduzione ad Android

Android:

Sistema operativo open source Linux based; composto da più livelli per supportare device differenti.

Ciascun livello utilizza i servizi dello strato sottostante e fornisce servizi al livello superiore.



La scelta del sistema operativo Linux è dovuta alla semplicità, alla sicurezza e affidabilità.

Sopra il layer costituito dal kernel di Linux c'è il livello che contiene l'insieme di librerie native, realizzate in C e C++, che rappresentano il core vero e proprio di Android.

Le app sono scritte in Java; il layout in Xml; la compilazione può avvenire tramite Android studio ma ci sono anche altri strumenti.

Per poter verificare il funzionamento dell'app, si può provare direttamente sul dispositivo oppure, si può emulare un dispositivo android tramite AVD di Android studio.

AVD= Android Virtual Device

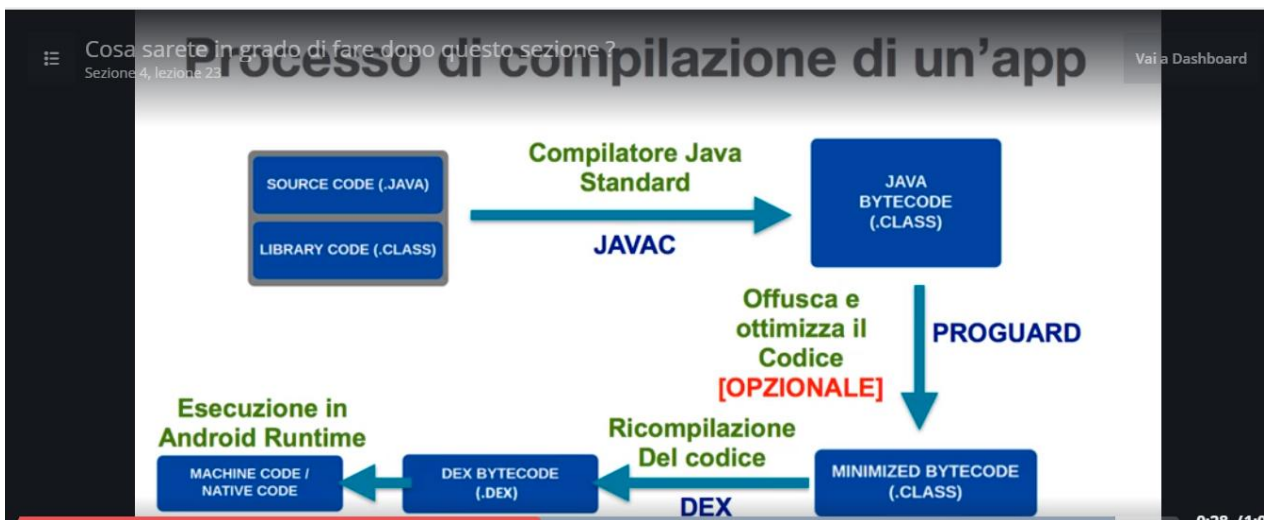
In Tools sotto la voce AVD Manager di Android studio

Per programmare una app di android è importante avere:

- Java JDK , Java Development Kit, che contiene tutti gli strumenti necessari a sviluppare nel linguaggio Java. Questo strumento è indispensabile in quanto Android si programma in Java, e senza un JDK non è possibile compilare il codice.
- un apposito kit di sviluppo (SDK), che sia completo di emulatore, librerie e documentazione; utile per poter vedere realizzate le app
- un IDE (Eclipse) con relativo add-on denominato Android Development Tools for Eclipse (ADT), utile per lavorare in un ambiente integrato.

La compilazione è doppia:

- I. viene fatta una compilazione java che genera un byte code class
- II. una ottimizzazione del codice
- III. una ricompilazione che genera un .apk



Blocchi costitutivi di un'applicazione:

Un **Activity** è un'interfaccia utente. La realizzazione di Activity è il punto di partenza di ogni progetto Android, vi è il componente con cui l'utente ha il contatto più diretto. È costituita da pulsanti, caselle, label...

Un **Service** svolge un ruolo interamente in background senza bisogno di interazione diretta con l'utente; i service spesso preparano i dati che le activity devono mostrare all'utente permettendo una reattività maggiore nel momento della visualizzazione.

Un **Content Provider** nasce con lo scopo della condivisione di dati tra applicazioni. La sua finalità richiama quel principio di sicurezza dell'applicazione.

Un **Broadcast Receiver** è un componente che reagisce ad un messaggi a di sistema – appunto in *broadcast* – con cui Android notifica l'avvenimento di un determinato evento, ad esempio l'arrivo di un SMS o di una chiamata o sollecita l'esecuzione di azioni o il livello di batteria. Questi componenti come si può immaginare sono particolarmente utili per la gestione istantanea di determinate circostanze speciali.

L'**architettura di progetto** così impostata è costituita da un certo numero di file e cartelle.

Tutti sono importanti ma gli elementi tra i quali il programmatore dovrà sapersi muovere al più presto con scioltezza sono:

- la **cartella *src*** che conterrà tutto il codice Java che scriveremo;
- la **cartella *res*** in cui risiederanno le cosiddette risorse dell'applicazione per la maggior parte configurate in XML ma non solo;
- il file ***AndroidManifest.xml*** anch'esso in XML che custodirà configurazioni e ruoli dei componenti della nostra app.

### **Ciclo di vita di una activity**

Active o running: è visibile e riceve i dati in input

Paused: è parzialmente visibile e non riceve dati in input

Stopped: non è visibile ma è in esecuzione

Destroyed: è rimossa dalla memoria

### **Sensori:**

- ambientali
- di posizione
- di movimento

I sensori in generale sono: accelerometro, giroscopio, magnetometro, sensore di prossimità, sensore di luminosità, sensore di temperatura e umidità, barometro, pedometro, sensore di impronte.