

# Programmazione di sistemi mobile

Alessandro Pioggia, Luca Rengo, Federico Brunelli, Leon Baiocchi

28 febbraio 2022

# Indice

# Capitolo 1

## Introduzione

Boh si fanno le app. Alcune domande comuni:

- Quanto tempo ci vuole? Boh si fanno le app;
- Serve la parte backend? Boh si fanno le app;
- Si può sviluppare il progetto con tecnologie ibride? Bah, boh si fanno le app;
- Si possono sviluppare giochi? Mandate mail, secondo le modalità descritte nella lezione, con la vostra idea... bla, bla, bla, si fanno le app.
- Useremo Java o Kotlin? Java, per fare l'app, anche se android ha dichiarato che lo sviluppo si focalizzerà su kotlin ma a noi piace guardare al futuro.

# Capitolo 2

## Mobile computing

Il mobile computing è la capacità di calcolo presente ovunque in ogni momento.

### 2.1 Fattori chiave

- Miniaturizzazione → Portabilità : Sviluppare componenti sempre più piccoli e potenti;
- Connettività : sviluppo di device collegati fra loro (attraverso bluetooth, wifi, ecc.)

### 2.2 5G

La next big thing, ovvero il più grande gomitolo mai creato.

### 2.3 Bluetooth low energy (Beacon)

Tecnologia che usa il bluetooth con uso ottimale di risorse.

### 2.4 RFID and NFC

Tecnologie simili, uno permette di scrivere, l'altro agisce solo in lettura.

### 2.5 Convergenza

I device sono praticamente in grado di fare tutto, come già si sa questo può portare sia vantaggi che svantaggi.

### 2.6 App

Siamo sommersi da App, è un problema, anche perchè noi siamo abituati ad utilizzarne davvero poche con continuità.

## 2.7 Attori principali mobile computing

- Mobile communication;
- Hardware mobile;
- Software mobile.

## 2.8 Sfide mobile

- Resource-constrained: i dispositivi, come sappiamo hanno una durata limitata della batteria;
- Connettività mobile;
- Sicurezza e privacy.

## 2.9 Sviluppo nativo e ibrido

Ibrido : sviluppo una volta e incastro nei due sistemi operativi.

Nativo : sviluppo unicamente per un sistema operativo.

---

```
//Hello.java
import javax.swing.JApplet;
import java.awt.Graphics;

public class Hello extends JApplet {
    public void paintComponent(Graphics g) {
        g.drawString("Hello, world!", 65, 95);
    }
}
```

---

# Capitolo 3

## Android

### 3.1 Introduzione

Android occupa l'80 per cento del mercato mobile, parliamo anche di altri device (non solo smartphone), ad esempio:

- wearable
- android tv
- android per le macchine
- android things (utilizzato per iot)

### 3.2 Qual è il sistema operativo più diffuso al mondo?

Il sistema operativo più diffuso, al 2019 è Android, dopo viene Windows.

### 3.3 Storia

Android è stato fondato nel 2003 da 4 amici : Andy Rubin, Rich Miner, Nick Sears e Chris White. Nel 2005 Google lo acquistò per 50 milioni di dollari. Google non solo lo compra, pensa bene di inserire nel sistema operativo il Kernel linux, trovata straordinaria! Lo sviluppo terminò il 5 novembre 2007. Il primo smartphone con Android viene fuori nel 2008, la prima versione accreditata è stata la 1.5 e ha preso il nome di cupcake. Fun fact : i nomi delle versioni (nomi di dolci) sono in ordine alfabetico. Nelle versioni successive ha dovuto cambiare, sia per mancanza di nomi che per questioni legate alle lamentele di determinati paesi.

### 3.4 Android 10

La versione 10 di android ha introdotto il 5G, ha migliorato la privacy e la security (es: randomizzazione del MAC address → {il nostro device ha un mac fisso, ma dal momento che si connette ne crea uno fasullo}, protezione del tracking, protezione dei dati degli utenti).

### 3.4.1 Maggiori novità

Grandi ottimizzazioni, ogni app gira in una sua sandbox, quindi le app sono staccate fra le altre, le comunicazioni fra app avvengono solo attraverso il sistema operativo, questo garantisce tanta sicurezza.

Thermal API: Miglioramenti legati alla temperatura, questa API, limita l'utilizzo di risorse.

## 3.5 Android 11

Rilasciata l'8 settembre 2020, novità principali: Lato privacy ha introdotto il one-time-permission (do il permesso solo una volta) e il permission auto-reset (se non uso l'app per un tot tempo toglie i permessi).

## 3.6 Android 12

C'è un completo restyling dell'interfaccia grafica, il motto è: More personal (personalizzabile), safe (privacy e sicurezza) and effortless (poco sforzo, hanno spinto molto su questo) than ever before. Questa versione si concentra davvero tanto sull'interfaccia grafica. Curato molto anche il tema dell'accessibilità, anche se ancora iOS è superiore. Android è open source... cool.

## 3.7 Android...In dettaglio!

- open source;
- sicuro.

## 3.8 Architettura

### 3.8.1 Kernel linux

Il fondamento della piattaforma Android è il kernel Linux, il quale comunica con l'hardware. Linux è portabile (facile da compilare nelle diverse architetture), sicuro, ha una gestione ottimale della memoria ed affidabile.

#### Sicurezza del kernel

Lato sicurezza si sfrutta interamente Linux (permessi, risorse, pid, boot, ecc.). Ogni applicazione, è in una sandbox ed ha un proprio UserID.

### 3.8.2 Hardware Abstraction Layer (HAL)

Siamo al livello superiore rispetto al Linux kernel e nasconde il vero device, astruendo tutto ciò che c'è a livello inferiore.

### 3.8.3 Android runtime

L'Android runtime rende possibile l'utilizzo delle sandbox, ogni app viene eseguita nel proprio processo e con la propria istanza di Android Runtime (ART). Questa è la parte ampiamente migliorata in Android 10. Qui viene gestita la virtualizzazione, dunque il garbage collector, debug, ecc... siamo in Java!

### 3.8.4 Native libraries

Le librerie Native sono scritte in C/C++. Serve comunicare con questa parte nativa nel caso in cui si debbano utilizzare dei sensori, altrimenti difficilmente capita.

### 3.8.5 Java API framework

As a Java hypebeast, this is very cool. Questo rappresenta il cuore di android.

### 3.8.6 System Apps

Queste sono le app preinstallate, es: camera, calendario, email, ecc. Solitamente vengono sfruttate dalle altre app, attraverso la comunicazione diretta.

## 3.9 Kotlin

Kotlin è un linguaggio molto promettente, leggibile. Ha differenze sostanziali con java, occorre del tempo. Netflix ad esempio è kotlin first.

## 3.10 Componenti fondamentali android

### 3.10.1 Activities

Tutto ciò che l'utente ha come possibilità per comunicare con la nostra App (è il suo punto di accesso con la nostra app). Quindi comprende : interfaccia, touch, ecc...

### 3.10.2 Services

I servizi sono tipo le activity, la differenza è che non sono visibili all'utente, sono tutte le robe che vanno in background (tipo spotify). Dunque **non** fornisce un'interfaccia utente.

#### Background services

Sono servizi che eseguono in background senza che l'utente ne sia direttamente a conoscenza come in esecuzione.

#### Broadcast receivers

Mandano messaggi ad altre app, in broadcast. E' un modo per scambiarsi messaggi fra le diverse app.

### 3.10.3 Content providers

Ci permette di capire dove memorizzeremo i dati, Android ha un proprio database interno.

### 3.10.4 Intent

E' tutto quello che ci permette di comunicare con altre app, è molto semplice ed intuitivo. Ci permetterà di esporre servizi anche agli altri.



# Capitolo 4

## lab01

I dati venivano scambiati attraverso XML, tutte le informazioni dentro a res sono in questo formato. Android studio generalmente li genera, dal momento che dà a disposizione la preview grafica.

### 4.1 Manifest

Il manifest gestisce i permessi di accesso (localizzazione, ecc.), è obbligatorio e deve assolutamente chiamarsi manifest.xml.

### 4.2 Progetto

#### 4.2.1 Gradle

Android studio sfrutta Gradle per il building, è basato su apache ant e apache maven e ci permette di gestire le dipendenze.

#### 4.2.2 Moduli

Gestione diversa dell'informazione in funzione del device utilizzato (smartphone o android tv o altre robe). Ogni modulo deve contenere:

- manifest
- java
- res