Programmazione Dispositivi Mobili

Sara Angeretti

2024/2025

Indice

Capitolo 1

Introduzione al corso

1.1 Obiettivi del corso

Il corso ha come obiettivo acquisire:

- **Conoscenze** (principi di buona programmazione) relative al mondo dello sviluppo mobile
- **Competenze** sullo sviluppo Android

Ma perché è stato scelto proprio Android? Comporta diversi vantaggi rispetto ad altri sistemi operativi come iOs:

- più open source
- essendo più open source conosciuto meglio dai docenti che sono quindi più in grado di insegnare e correggere
- Android, nel caso uno voglia poi accedere allo Store e pubblicare un'app, prevede una tassa di iscrizione di ~25$ **una tantum**, mentre per iOs è di 100$ ma penso sia *annuale*.

Ci aspettiamo che alla fine del corso siate in grado di:

- Sviluppare un'applicazione "from scratch" che segua l'**architettura** di riferimento Android
  - alla fine se abbiamo rispettato o no l'architettura presentata a lezione è quello che guardano di più della nostra app, se non è bellissima o funzionante al 101% importa meno
- Comprendere il funzionamento di applicazioni Android

1.2 Il corso in pillole

1. Introduzione alla progettazione e allo sviluppo di applicazioni mobili
2. Linee guida sull'architettura dell'app

3. Sviluppo di un'app in Java
   Per il nostro progetto possiamo usare Java o Kotlin, la teoria rimane la stessa, ma a lezione useremo solo Java. Questo perché è già stato presentato ed usato in altri due corsi e quindi conosciuto meglio da docenti e studenti. Inoltre, è previsto (penso in entrambi i linguaggi) l'uso di lambda functions, più elegante e funzionale in Kotlin, però buono anche in Java. Infine, Java risulta più conveniente per l'uso di librerie esterne di Kotlin, che è più giovane e meno conosciuto e quindi ha meno librerie disponibili.
   Eventualmente, sul sito Google ci sono disponibili diversi tutorial gratuiti (video) per imparare Kotlin e per la migrazione del mio progetto da Kotlin a Java.
   L'app deve essere robusta. La robustezza si basa sull'autonomia dalla connessione di rete. Il concetto "offline-first" è molto importante, prima di tutto l'app deve funzionare senza connessione. Inoltre, deve essere anche evolvibile. Oltre ai suoi componenti funzionali (ovvero le sue funzionalità) di base, ci sono determinate funzionalità che devono essere mantenute ed evolute/ampliate nel tempo.
   La nostra app deve essere:

   3.1 Compliant con l'architettura di riferimento
   La cosa importante della nostra app non è l'estetica o se funziona bene, ma **come** l'architettura presentata a lezione viene sviluppata.

   3.2 Con UI

   3.3 Che accede alla rete per i dati (API esterne)

   3.4 Che fa persistenza (locale + remoto)
   Ovvero deve funzionare *localmente* e salvare localmente i dati (importante per il concetto di "*offline-first*"). Però deve anche salvare *in remoto* i dati, ma deve rispettare la *cross-device synchronization.* Importante per quanto riguarda la cross-device synchronization, ovvero deve funzionare su diversi dispositivi con stato sincronizzato.

   3.5 Che usa Firebase
   Firebase è un framework di Google che offre una serie di servizi per lo sviluppo di applicazioni mobili.