

Flutter

Framework-ul perfect pentru o aplicație de mobil?

Buhai Darius - 234.2

1 Context

O problemă comună întâmpinată în dezvoltarea aplicațiilor mobile este portabilitatea acestora (pe ce sisteme rulează? iOS, Android, Windows...), mai precis în câte limbaje de programare trebuie să rescriem aceeași aplicație pentru a putea fi instalată pe orice dispozitiv (Swift, Java, C...). Astfel multe startup-uri și business-uri mici au început să folosească **Cross-Platform Frameworks** cu ajutorul cărora pot dezvolta aplicații mobile pe majoritatea sistemelor de operare. Cele mai populare la momentul curent sunt:

1. Flutter (limbajul Dart)
2. React Native (Javascript)
3. Xamarin (C#)
4. Ionic (Javascript)

2 De ce Flutter?

Flutter este un framework realizat de Google ce oferă posibilitatea de a crea aplicații de iOS, Android, Windows, MacOS, Linux și Web (suportul pentru iOS și Android fiind cel mai stabil momentan) folosind un singur limbaj de programare. Spre deosebire de celelalte framework-uri disponibile, Flutter folosește limbajul **Dart** relativ ușor de învățat și utilizat (limbaj Orientat pe Obiecte).

Pe partea de UI, Flutter conține 2 librării principale (**Cupertino** - pentru iOS și **Materials** - pentru Android) cu ajutorul cărora se pot utiliza componente native pentru ambele sisteme de operare principale. De asemenea, un feature important prezent în Flutter este **Hot Reloading**, cu ajutorul căruia modificările de UI realizate în cod pot fi vizionate live pe mobil sau pe emulator.

De asemenea, în Flutter se pot instala librării dart printr-o singură comandă (pub get), market place-ul dart oferind unele dintre cele mai diversificate librării, precum google_maps_flutter, facebook_auth și multe altele.

Concluzie: Flutter reprezintă Framework-ul perfect atunci când se dorește crearea unei aplicații de mobil compatibilă cu cele mai importante sisteme de operare în cel mai scurt timp posibil.

3 Referințe utile

- [Flutter](#)
- [Librăria Dart](#)
- [Componente flutter](#)
- [Aplicații realizate în Flutter](#)