FLUTTER

Durante il primo incontro con il committente del progetto, in merito allo sviluppo dell’app mobile è stato consigliato al gruppo di preferire l’utilizzo del framework Flutter rispetto a linguaggi di programmazione nativi. Questa linea giuda è stata accolta conducendo una ricerca sugli aspetti positivi e negativi della tecnologia consigliata.

Flutter è un progetto open-source sviluppato da Google nel 2018 al fine di creare interfacce native per iOS e Android, usando Dart come unico linguaggio di programmazione.

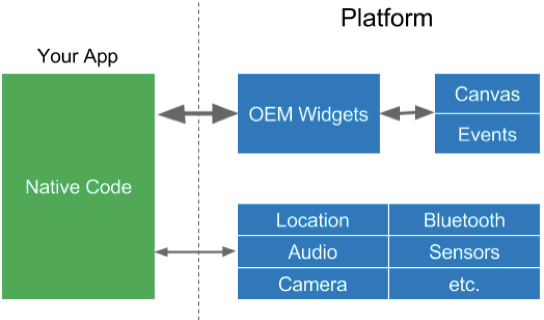
|  |  |
| --- | --- |
| PRO | CONTRO |
| Un unico linguaggio di programmazione e un unico codice sorgente per app iOS e Android | I molteplici widget rendono il codice molto nidificato |
| Il team di sviluppo può dedicare le sue forze allo sviluppo di un unico prodotto per entrambi i sistemi operativi ottimizzando i tempi | Dart è un linguaggio di programmazione nuovo e poco diffuso |
| Permette di raggiungere performance molto vicine a quelle di un app nativa |  |
| Tutti gli elementi dell’applicazione sono widget, è possibile crearne di complessi combinando insieme widget più semplici. Questo dà molte opportunità allo sviluppatore che può esprimere molta creatività |  |
| Sono presenti librerie molto ricche per l’interfaccia utente |  |
| Fase di sviluppo molto rapida con la funzionalità “hot reload” che non richiede di ricompilare il codice |  |

Sulla base di questi elementi è chiaro che i fattori positivi superano nettamente quelli negativi, perciò utilizzare Flutter risulta l’opzione migliore.

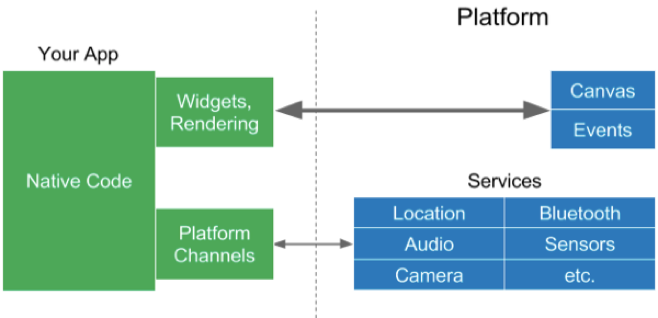
Tratti distintivi:

Flutter è diverso da tutti gli altri framework per lo sviluppo di app ibride perché non utilizza né i widget OEM (original equipment manufacturer) ovvero i componenti UI nativi del sistema, e neppure WebView cioè lo strumento che permette di aprire i link web all’interno dell’app stessa

Solitamente quando un’app nativa interagisce con la piattaforma per creare un widget, quest’ultimo viene scelto tra gli OEM widget disponibili, viene visualizzato all’interno di un canvas e gli eventi vengono passati ai widget che a loro volta comunicheranno con l’applicazione nativa.



Flutter invece sfrutta il proprio motore di rendering ad alte prestazioni per creare i widget



Librerie utili:

* web3dart: permette di interagire con la blockchain ethereum, si connette ad un nodo di ethereum per inviare transazioni, interagire con gli smart contracts ecc…

<https://pub.dev/packages/web3dart>

<https://medium.com/@dev_89267/develop-blockchain-applications-with-flutter-ethereum-59e846944127>

* qr\_flutter: consente di generare QR code attraverso widget
* qr\_code\_scanner: viene usata per permettere di scannerizzare i QR code grazie alla fotocamera