

Flutter vs React Native

Azael Garcia Rufer

March 24, 2025

Contents

Contents	1
1 Introduzione	2
2 Flutter	2
2.1 Cos'è Flutter?	2
2.2 Caratteristiche principali	2
2.3 Vantaggi e Svantaggi	2
3 React Native	2
3.1 Cos'è React Native?	2
3.2 Caratteristiche principali	2
3.3 Vantaggi e Svantaggi	3
4 Confronto Diretto	3
5 Flutter vs React Native – Qual è il migliore?	3
5.1 Perché scegliere Flutter	3
5.2 Quando scegliere Flutter	3
5.3 Perché scegliere React Native	4
5.4 Quando usare React Native	4
6 Conclusione	4

Introduzione

Negli ultimi anni, lo sviluppo di applicazioni mobili si è evoluto grazie a framework cross-platform come **Flutter** (Google) e **React Native** (Meta/Facebook). Questi strumenti permettono di creare app per Android e iOS con un'unica codebase, semplificando il processo di sviluppo.

Questa relazione si propone di:

- Presentare Flutter e React Native, analizzandone vantaggi e svantaggi.
- Confrontarli in termini di performance, architettura e usabilità.
- Fornire indicazioni pratiche su quale scegliere in base alle esigenze del progetto.

Flutter

2.1 Cos'è Flutter?

Flutter è un framework open-source sviluppato da **Google** nel 2017. Utilizza il linguaggio **Dart** e un proprio motore di rendering chiamato **Skia**.

2.2 Caratteristiche principali

- **Rendering Proprio:** Non usa i componenti nativi, ma disegna la UI con il proprio motore grafico.
- **Hot Reload:** Permette aggiornamenti istantanei senza riavviare l'app.
- **UI Personalizzabile:** Supporta Material Design (Android) e Cupertino (iOS).
- **Alta Performance:** Compilazione nativa (AOT - Ahead Of Time).

2.3 Vantaggi e Svantaggi

Vantaggi:

- Prestazioni elevate e animazioni fluide.
- Interfaccia coerente tra iOS e Android.
- Supporto per Web, Desktop e Mobile.

Svantaggi:

- App più pesanti rispetto a React Native.
- Dart meno diffuso rispetto a JavaScript.

React Native

3.1 Cos'è React Native?

React Native è un framework sviluppato da **Meta (Facebook)** nel 2015. Utilizza **JavaScript** e il framework React per creare app mobili sfruttando componenti nativi.

3.2 Caratteristiche principali

- **Basato su JavaScript:** Facile da apprendere per chi conosce il web.
- **Componenti Nativi:** Usa gli elementi della UI propri di iOS e Android.
- **Hot Reload:** Permette di testare rapidamente le modifiche.

3.3 Vantaggi e Svantaggi

Vantaggi:

- Ampia community e molte librerie disponibili.
- Facile integrazione con codice nativo.
- JavaScript è molto popolare tra gli sviluppatori.

Svantaggi:

- Performance inferiore rispetto a Flutter (a causa del bridge JavaScript-to-Native).
- Problemi di compatibilità tra versioni di iOS e Android.

Confronto Diretto

Caratteristica	Flutter	React Native
Linguaggio	Dart	JavaScript (React)
Rendering UI	Skia (personalizzato)	Componenti nativi
Performance	Alta (compilazione AOT)	Buona, ma dipende dal bridge
Facilità di apprendimento	Media	Facile (JavaScript è popolare)
Community	In crescita	Più ampia
Ecosistema	In espansione	Maturo e consolidato

Table 1: Confronto tra Flutter e React Native

Flutter vs React Native – Qual è il migliore?

La scelta tra Flutter e React Native dipende da vari fattori come i requisiti del progetto, le competenze del team di sviluppo e le necessità di performance.

Flutter, che utilizza Dart, offre alte prestazioni ma ha una base di sviluppatori più piccola e un supporto IDE limitato. React Native, che sfrutta JavaScript, ha un supporto comunitario più ampio e uno sviluppo più veloce grazie al linguaggio familiare.

Alla fine, la decisione dovrebbe essere basata sulle necessità specifiche del progetto e sulle preferenze del team di sviluppo.

5.1 Perché scegliere Flutter

- Usa il linguaggio Dart, che compila più velocemente di JavaScript, garantendo alte prestazioni.
- Dart ha una diffusione inferiore rispetto a JavaScript, rendendolo meno familiare agli sviluppatori.
- Supporto IDE limitato per Dart a causa della sua minore popolarità.
- Consente uno sviluppo più rapido ma porta a file di dimensioni maggiori.
- Offre funzionalità di test integrate per facilitare i test.

5.2 Quando scegliere Flutter

Flutter può essere la scelta ideale quando:

- La tua app ha bisogno di design UI altamente personalizzati e perfetti a livello di pixel. Flutter ha un'architettura basata su widget e funziona senza fare affidamento su componenti specifici della piattaforma.
- La tua app ha una grafica pesante e animazioni complesse e richiede alte prestazioni. Flutter può supportare questo tramite il proprio motore di rendering Skia.
- Vuoi che la tua app abbia un aspetto identico su dispositivi Android e iOS, quindi Flutter è la scelta giusta.

5.3 Perché scegliere React Native

- Sfrutta JavaScript, che ha un supporto comunitario più ampio ed è più facile da imparare per gli sviluppatori.
- Prestazioni più lente a causa del bridge JavaScript-to-Native.
- Risparmia tempo per gli sviluppatori grazie all'uso di un linguaggio familiare.
- Richiede framework di test di terze parti come Detox per i test.
- Più ampiamente adottato e supportato da una comunità più grande.

5.4 Quando usare React Native

Conviene utilizzare React Native quando:

- Vuoi riutilizzare i componenti della tua app desktop o sito web per una mobile app.
- Se il tuo team è più esperto in JavaScript o React, React Native dovrebbe essere la scelta ovvia, poiché utilizza lo stesso linguaggio e gli stessi concetti.
- Se desideri ridurre il consumo di memoria, in particolare sui dispositivi Android, React Native è una buona scelta, poiché il motore Hermes aiuta a ottimizzare l'uso della memoria.
- Quando la tua app dovrà fare un uso frequente di moduli nativi come fotocamera, GPS, ecc., React Native può essere scelto grazie alla sua robusta integrazione con le API native.

Conclusione

Entrambi i framework sono ottimi strumenti per lo sviluppo mobile. La scelta dipende dalle esigenze del progetto e dalle competenze del team.

Se cerchi performance elevate e UI coerente → scegli Flutter.

Se vuoi flessibilità e facilità di sviluppo con JavaScript → scegli React Native.