

Riferimenti per Flutter

1. Cosa è Flutter

Flutter è un framework open source di Google per la creazione di bellissime applicazioni multi piattaforma, per Android, iOS, Web, Windows, Mac e Linux, compilate in modo nativo da un'unica base di codice. A seconda della piattaforma scelta le applicazioni sono compilate per Intel, Arm o Javascript (per la versione web)

2. Piattaforme supportate e Multi-Platform

Attraverso l'utilizzo di Dart come linguaggio di programmazione è possibile scrivere in un solo linguaggio, e poi fare il deploy della propria applicazione sulle varie piattaforme supportate arrivando anche a condividere quasi tutto il sorgente che diventa così più facile da mantenere, sviluppare, ottenere una consistenza e parità di funzionalità tra tutte le piattaforme, diminuire i tempi e i costi di sviluppo e di rilascio di nuove funzionalità, fornire assistenza e bugfix più rapidamente

- Mobile: iOS and Android apps
- Web
- Desktop: Windows, MacOS, Linux
- Embedded devices

3. Chi lo utilizza?

▼ Flutter apps in production

BMW: Scalare lo sviluppo del prodotto incentrato sul cliente al BMW Group con Flutter

Google Pay: Diventare globale in Google Pay con Flutter

eBay: Deliziare gli ingegneri di eBay con Flutter

Nubank: Scalare con integrità a Nubank con Flutter

Rive: Rive ha riscritto il suo potente strumento di animazione interamente in Flutter per consentire agli sviluppatori di creare bellissime illustrazioni multipiattaforma

Supernova: una piattaforma di design system, ha usato Flutter per costruire la propria web app collaborativa per designer e sviluppatori

Toyota: Miglioramento dei sistemi di infotainment alla Toyota con Flutter

Superlist: Ripensare la gestione dei compiti e dei progetti con Flutter

Tonal: Garantire la parità di funzionalità in Tonal con Flutter

Crédit Agricole: Crédit Agricole mette i clienti al primo posto con Flutter

4. Success metrics

▼ Qualche numero

Google Pay: riduzione del 70% degli sforzi ingegneristici

Crowdsource: riduzione del 50% delle dimensioni del codice

eBay: 100% degli sviluppatori dichiara di preferirlo a iOS o Android

Nubank: 30% di miglioramento del tasso di successo delle merge rispetto a iOS

Push: 20% di riduzione degli sforzi di manutenzione

eBay: 70% degli sviluppatori ritiene che lo sviluppo sia due volte più veloce rispetto alle soluzioni native

eBay: 98.3% di codice condiviso

Tencent: 33% di riduzione dello sforzo di sviluppo

Google Pay: 35% di riduzione delle linee di codice

Tencent: 90% di codice multiplatforma

Nubank: tasso di merge del 600% più veloce

Push: 100% di parità di funzionalità

Tencent: 80% di aumento dell'efficienza del debug

Crowdsource: aumento del 100% della velocità di sviluppo

ByteDance: 33% di aumento della produttività

5. Marketshare e confronto con altri tool

a. vs React Native (Flutter for React Native developers)

b. vs Xamarin (Flutter for Xamarin.Forms developers)

c. vs Web (Flutter for web developers)

d. vs Android (Flutter for Android developers)

e. vs iOS (Flutter for iOS developers)

6. Parole chiave del mondo Flutter

- a. Widget
 - i. Introduction to widgets
 - ii. Flutter widget index
 - iii. Widget catalog
 - iv. Tutti i 215 widget: Every Flutter Widget Explained!
- b. Hot Reload: La funzione **hot reload** di Flutter ti aiuta a sperimentare, creare interfacce utente, aggiungere funzionalità e correggere bug in modo rapido e semplice. L'**hot reload** funziona iniettando file di codice sorgente aggiornati nella Dart Virtual Machine (VM) in esecuzione. Dopo che la VM ha aggiornato le classi con le nuove versioni di campi e funzioni, il framework Flutter ricostruisce automaticamente l'albero dei widget, consentendo di visualizzare rapidamente gli effetti delle modifiche
- c. Skia
 - i. Architectural layers
 - ii. Rendering and layout: È utile iniziare pensando al funzionamento delle applicazioni Android tradizionali. Quando si disegna, si chiama prima il codice Java del framework Android. Le librerie di sistema di Android forniscono i componenti responsabili del disegno su un oggetto Canvas, che Android può poi renderizzare con Skia, un motore grafico scritto in C/C++ che chiama la CPU o la GPU per completare il disegno sul dispositivo. I framework multiplatforma funzionano in genere creando un livello di astrazione sulle librerie UI native di Android e iOS, cercando di attenuare le incongruenze della rappresentazione di ciascuna piattaforma. Il codice dell'applicazione è spesso scritto in un linguaggio interpretato come JavaScript, che a sua volta deve interagire con le librerie di sistema Android basate su Java o iOS basate su Objective-C per visualizzare l'interfaccia utente. Tutto questo aggiunge un overhead che può essere significativo, in particolare quando c'è molta interazione tra l'interfaccia utente e la logica dell'applicazione. Al contrario, Flutter riduce al minimo queste astrazioni, aggirando le librerie di widget UI di sistema a favore del proprio set di widget. Il codice Dart che dipinge le immagini di Flutter è compilato in codice nativo, che utilizza Skia per il rendering. Flutter incorpora anche la propria copia di Skia come parte del motore, consentendo allo sviluppatore di aggiornare la propria applicazione con gli ultimi miglioramenti delle prestazioni anche se il

telefono non è stato aggiornato con una nuova versione di Android. Lo stesso vale per Flutter su altre piattaforme native, come iOS, Windows o macOS

d. Dart

i. Bootstrap into Dart

ii. Why did Flutter choose to use Dart?

e. IDE: Puoi creare app con Flutter utilizzando qualsiasi editor di testo combinato con gli strumenti da riga di comando di Flutter. Tuttavia, ti consigliamo di utilizzare uno dei nostri plugin per l'editor per un'esperienza ancora migliore. Questi plug-in forniscono il completamento del codice, l'evidenziazione della sintassi, l'assistenza alla modifica dei widget, il supporto per l'esecuzione e il debug e altro ancora

i. VS Code

ii. Android Studio, IntelliJ

7. Documentazione

a. Learn Flutter any way you want

8. Package e pub.dev

a. Flutter Favorite program: identificare i pacchetti e i plugin che dovrete prendere in considerazione per la realizzazione della vostra applicazione. Questa non è una garanzia di qualità o di idoneità alla vostra situazione specifica: dovrete sempre valutare voi stessi i pacchetti e i plugin per il vostro progetto

9. Building a Flutter business

a. Ads in your Flutter app: L'SDK Google Mobile Ads per Flutter funziona sia con AdMob che con Ad Manager. Disponibili anche una varietà di formati di annunci, inclusi banner, interstitial, video con premio e annunci nativi per guadagnare.

b. In-app purchases in Flutter: Integra servizi per offrire contenuti aggiuntivi nella tua app come servizi premium, beni digitali e abbonamenti.

c. Accepting payments: Semplifica l'integrazione con più fornitori di servizi di pagamento come Google e Apple Pay con il plug-in Pay per Flutter.

- d. RevenueCat: RevenueCat fornisce un backend per gli abbonamenti e un wrapper attorno a StoreKit di Apple, Google Play Billing e ai pagamenti basati sul web per semplificare la gestione degli acquisti in-app e la centralizzazione dei dati degli abbonati. Non è necessaria alcuna manutenzione del server o codice backend.
- e. Stripe: L'SDK di Stripe Flutter vi permette di creare esperienze di pagamento piacevoli nelle vostre app native per Android e iOS utilizzando Flutter. Forniamo schermate ed elementi dell'interfaccia utente potenti e personalizzabili che possono essere utilizzati immediatamente per raccogliere i dati di pagamento degli utenti.
- f. Video:
 - i. Monetizing apps with Flutter
 - ii. Flutter Update: App Monetization

10. Adaptive e Responsive

- a. Adaptive: adattamento di una app per l'esecuzione su diversi tipi di dispositivi, come dispositivi mobili e desktop, richiede la gestione dell'input da mouse e tastiera, nonché l'input tattile. Significa anche che ci sono aspettative diverse sulla densità visiva dell'app, su come funziona la selezione dei componenti (menu a cascata rispetto ai bottom sheets), utilizzando funzionalità specifiche della piattaforma (come le finestre di primo livello) e altro ancora.
- b. Responsive: la app ha un layout ottimizzato per le dimensioni dello schermo su cui viene visualizzata. Spesso questo significa ridisporre l'interfaccia utente se l'utente ridimensiona la finestra o cambia l'orientamento del dispositivo. Ciò è particolarmente necessario quando la stessa app può essere eseguita su una varietà di dispositivi, da un orologio, un telefono, un tablet, un laptop o un computer desktop.

▼ Risorse

[Designing truly adaptive user interfaces](#)

[Platform-specific behaviors and adaptations](#)

[Using Flutter to build a native-looking desktop app for macOS and Windows](#)

[Adaptive and Responsive! Simple practical guide](#)

11. Windows, MacOS e Linux

- a. [Desktop support for Flutter](#)
 - b. [Building Windows apps with Flutter](#)
 - c. [Building macOS apps with Flutter](#)
 - d. [Building Linux apps with Flutter](#)
12. [CI/CD of course](#)
- a. [Codemagic](#), [Bitrise](#), [Appcircle](#)
13. **DEMO**
14. Riferimenti e risorse online
- a. [Documentazione ufficiale](#)
 - b. [Getting Started](#)
 - c. [Widget catalog](#)
 - d. [Cookbook](#)
 - e. [Samples](#)
 - f. [FAQ](#)
 - g. Canali YouTube
 - i. [Flutter](#)
 - ii. [Code With Andrea](#)
 - iii. [Fudeo - Corsi online in italiano su Flutter](#)
 - iv. [Reso Coder](#)
 - v. [RobertBrunhage](#)
 - vi. [Codepur](#)
 - h. Blog - Siti
 - i. [Code with Andrea](#)
 - ii. [Fudeo](#)
 - iii. [Robert Brunhage](#)
 - iv. [Reso Coder](#)
 - v. [Flutter and Other Experiments](#)