



# Presentazione 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 usa?

### ▼ Flutter apps in production

BMW: Scaling customer-centric product development at BMW Group with Flutter

Google Pay: Going global at Google Pay with Flutter.

eBay: Delighting engineers at eBay with Flutter

Nubank: Scaling with integrity at Nubank with Flutter

Rive: Rive rewrote their powerful animation tool entirely in Flutter to enable developers to create beautiful, multi-platform illustrations

Supernova: a design system platform, used Flutter to build their collaborative web app for designers and developers

Toyota: Improving infotainment systems at Toyota with Flutter

Superlist: Rethinking task and project management with Flutter

Tonal: Ensuring feature parity at Tonal with Flutter

Crédit Agricole: Crédit Agricole puts customers first with Flutter

#### 4. Success metrics

##### ▼ Qualche numero

Google Pay: 70% reduction in engineering effort

Crowdsource: 50% reduction in code size

eBay: 100% developers say they prefer over iOS or Android

Nubank: 30% better merge success rate compared to iOS

Push: 20% reduction in maintenance effort

eBay: 70% developers believe development is 2x faster than native solutions

eBay: 98.3% shared code

Tencent: 33% reduction in development effort

Google Pay: 35% reduction in lines of code

Tencent: 90% multiplatform code

Nubank: 600% faster merge rate

Push: 100% feature parity

Tencent: 80% increased debugging efficiency

Crowdscore: 100% increase in development speed

ByteDance: 33% increase in productivity

#### 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
- 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: It's useful to start by thinking about how traditional Android apps work. When drawing, you first call the Java code of the Android framework. The Android system libraries provide the components responsible for drawing themselves to a Canvas object, which Android can then render with Skia, a graphics engine written in C/C++ that calls the CPU or GPU to complete the drawing on the device. Cross-platform frameworks typically work by creating an abstraction layer over the underlying native Android and iOS UI libraries, attempting to smooth out the inconsistencies of each platform representation. App code is often written in an interpreted language like JavaScript, which must in turn interact with the Java-based Android or Objective-C-based iOS system libraries to display UI. All this adds overhead that can be significant, particularly where there is a lot of interaction between the UI and the app logic. By contrast, Flutter minimizes those abstractions, bypassing the system UI widget libraries in favor of its own widget set. The Dart code that paints Flutter's visuals is compiled into native code, which uses Skia for rendering. Flutter also embeds its own copy of Skia as part of the engine, allowing the developer to upgrade their app to stay updated with the latest performance improvements even if the phone hasn't been updated with a new Android version. The same is true for Flutter on other native platforms, such as iOS, Windows, or 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: identify packages and plugins that you should first consider when building your app. This is not a guarantee of quality or suitability to your particular situation—you should always perform your own evaluation of packages and plugins for your project.

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 provides a subscription backend and wrapper around Apple's StoreKit, Google Play Billing, and web-based payments to make managing in-app purchases and centralizing subscriber data simple. No server maintenance or backend code required.

- e. Stripe: The Stripe Flutter SDK allows you to build delightful payment experiences in your native Android and iOS apps using Flutter. We provide powerful and customizable UI screens and elements that can be used out-of-the-box to collect your users' payment details.

## 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 adaptative user interfaces](#)

[Platform-specific behaviors and adaptations](#)

## 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](#)