



# Scontro tra titani

Facciamo chiarezza sul mondo delle app

# SOMMARIO

Panoramica

Native

Web

PWA

Hybrid

Cross-Platform

Web App vs Others

Criteri per la scelta

Dilemma

Discussione





# Panoramica

Devo sviluppare un'applicazione:

- Qual è l'app che fa meglio per me?
- Quali sono le risorse di cui ho bisogno?

We are in a *Jungle of apps!*



# Classificazione delle applicazioni

## Native App





# Classificazione delle applicazioni

## **Native App** - Vantaggi

- Ottime prestazioni
- Ottimizzazione delle risorse disponibili
- Accesso alle funzionalità del dispositivo
- Accesso ai componenti hardware installati
- UI ottimizzata
  - Fluidità
  - Animazioni



# Classificazione delle applicazioni

## **Native App** - Svantaggi

- Ogni OS ha il suo linguaggio
- Cicli di vita differenti
- Non *portabilità*
- Più *team* di sviluppo
- *Simmetria* dei codici sorgenti
- Installazione e occupazione spazio memoria



# Classificazione delle applicazioni

Web App





# Classificazione delle applicazioni

## **Web App** - Vantaggi

- *“Write once, use everywhere”*
- Nessuna installazione, no occupazione memoria
- Aggiornati all’ultima versione
- Costi di sviluppo abbattuti
- Uso di linguaggi tipici del Web
- Portabilità (browser)



# Classificazione delle applicazioni

## **Web App** - Svantaggi

- Non sono delle vere e proprie applicazioni
- Sono siti Web resi *responsive*
- Pessima gestione dell'hardware
- Poca ottimizzazione
- Cattiva esperienza d'uso (browser)
- Ambito di applicazione limitato
- Web App = NOT(Native App)



Classificazione delle applicazioni



# Progressive Web App



# Classificazione delle applicazioni

## **PWA** - Vantaggi

- Categoria molto simile alle Web App
- *Notifiche push*
- Browser
- UX leggermente migliore
- Framework: **Angular** e **ReactJS**



# Classificazione delle applicazioni

## **PWA** - Svantaggi

- Medesimi svantaggi delle Web App
- [PWA vs. Web App](#)



# Classificazione delle applicazioni



## Hybrid



# Classificazione delle applicazioni

## **Hybrid** - Vantaggi

- Evoluzione delle Web App e delle PWA
- **Engine**
  - Migliore UI
  - Migliori prestazioni
  - Accesso maggiore all'hardware
- Stessi linguaggi del Web
- Portabilità
- Framework: **Ionic**



# Classificazione delle applicazioni

## **Hybrid** - Svantaggi

- L'applicazione deve essere pubblicata in uno *store*
- Rispetto alle *Native App*, la UX non è comparabile
  - *Engine* troppo astratto dall'architettura fisica



Classificazione delle applicazioni



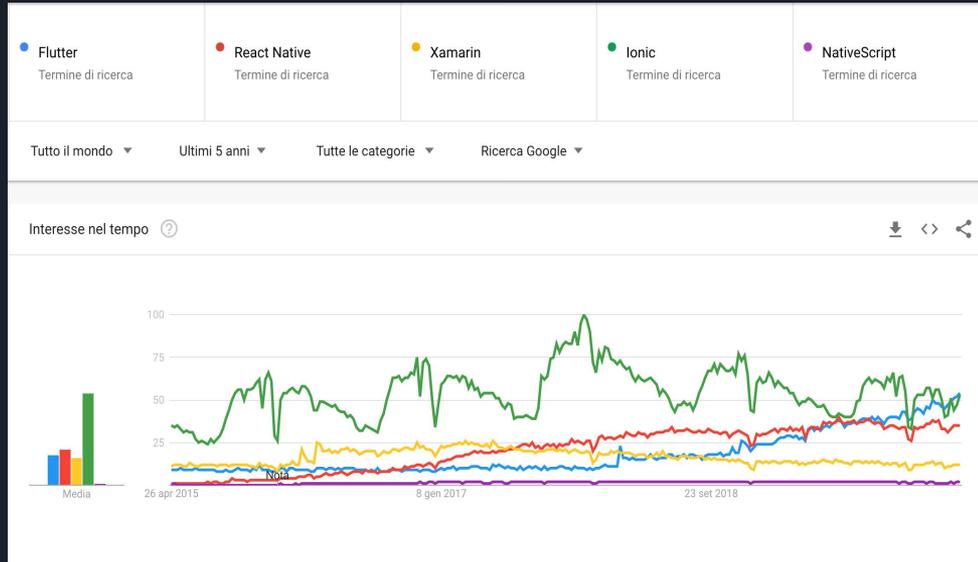
Cross-Platform



# Classificazione delle applicazioni **Cross-Platform** - Vantaggi

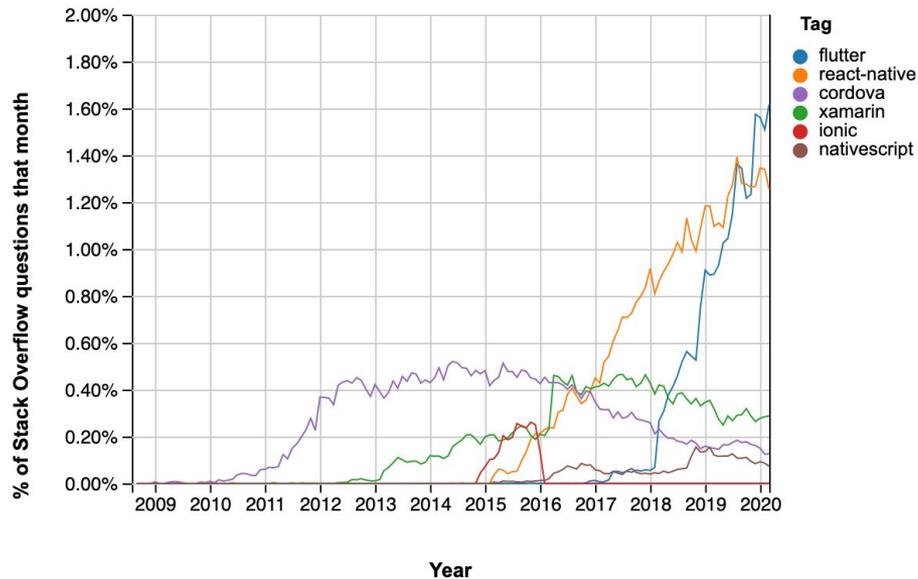
- Non sono app ibride!
- **Render Engine Nativo**
  - Migliori prestazioni, vicine alle *Native App*
  - Miglior accesso all'hardware
  - UI e UX vicine alle *Native App*
- Framework: **Flutter, React Native, Xamarin**

# Classificazione delle applicazioni Cross-Platform



- [Google Trends](#), Maggio 2020.

# Classificazione delle applicazioni Cross-Platform



- [StackOverflow](#), Maggio 2020.

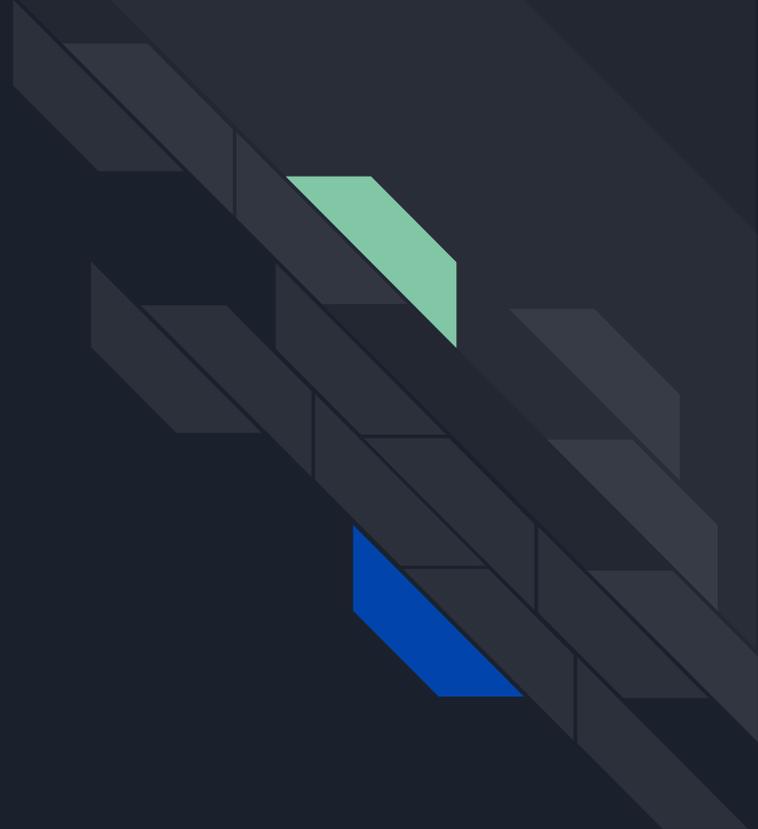


# Classificazione delle applicazioni **Cross-Platform** - Svantaggi

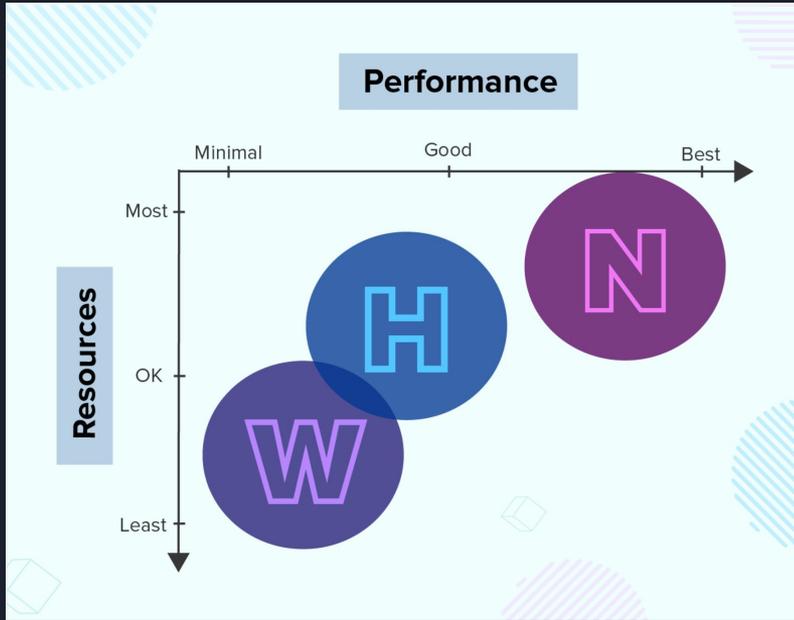
- L'applicazione deve essere pubblicata in uno *store*
- Possono essere utilizzati dei linguaggi non comuni al Web
  - Flutter → Dart
  - Xamarin → C#



# Web App vs Others



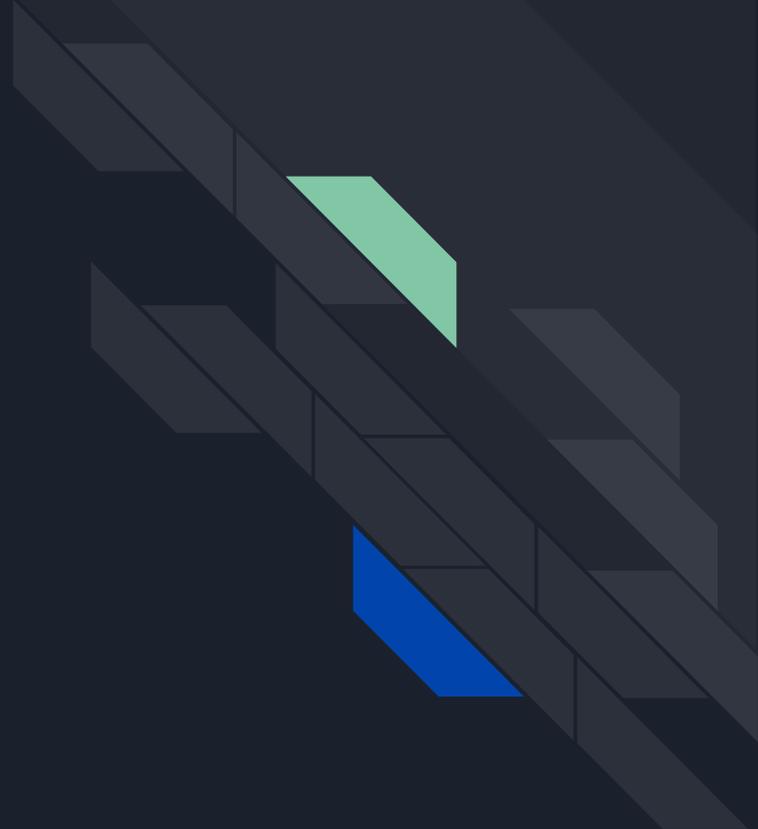
# Primo Confronto



Trade-off tra Web, Hybrid e Native



# Criteri per la scelta





# Risorse economiche

- **PMI o Startup** = poche risorse economiche
- Evitare il *nativo*, molto dispendioso
  - Assumere *devs* per ogni OS
  - i.e.: Un team per lo sviluppo Web, uno per Android e uno per iOS
  - Concettualmente, tanti sviluppatori per realizzare una singola app
- Consigliato l'approccio *platform-independent*
  - La scelta dipende dall'obiettivo che si vuole raggiungere



# Risorse economiche

- Se l'aspetto economico non è un problema, non è *necessario* andare sul *native*
- Lo stesso team che ha fatto il sito Web, può anche realizzare l'app
- Ottimizzare le risorse all'interno dell'azienda
  - Tempi ristretti per realizzazione e consegna
  - Evitare di assumere altri devs



# Time-to-market e reach

- **Time-to-market**
  - Tempistiche dettate dal mercato
  - Influenza la scelta
  - Vincere sul *competitor*
  - Raggiungere nel più breve tempo possibile la *massa critica*
  - Dopo questo obiettivo, si può valutare se passare al *nativo*
- **Reach**
  - Copertura del mercato con la propria app
  - Distribuire l'app su più store possibili
  - Approcci Web/ibridi sono favoriti



# Performance

- Dipende dal *target* e dall'obiettivo da raggiungere
- **Overhead** e accesso alle risorse del dispositivo

Totale accessibilità



Native



Cross



Hybrid



PWA

Poca accessibilità

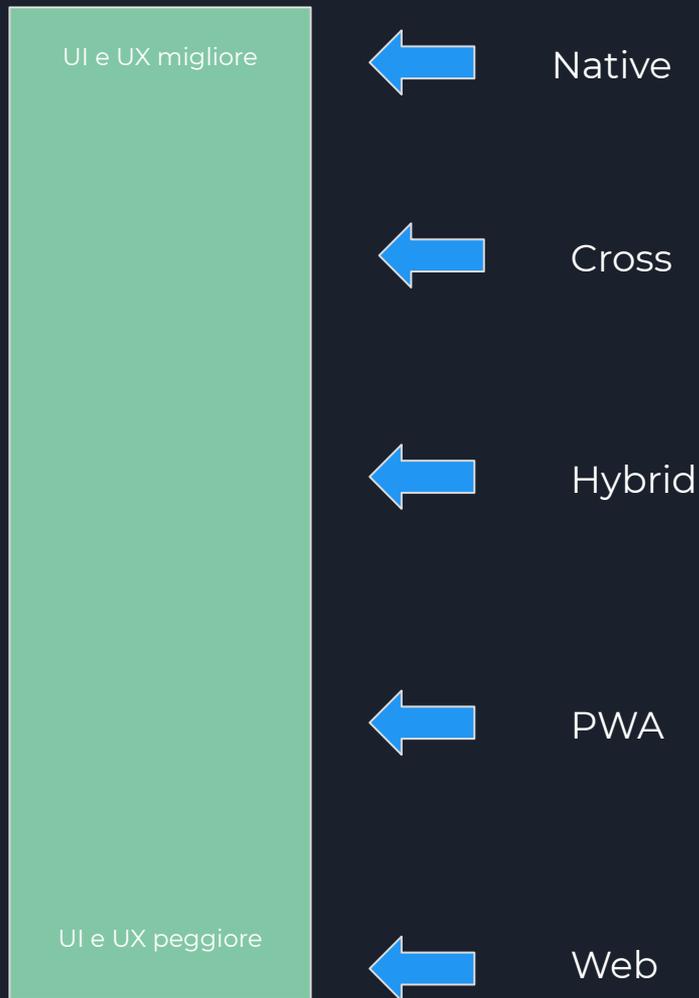


Web



# UI e UX

- App *native* offrono la UI e UX migliore
- Fluidità e animazioni





# Funzionalità

- Le app *native* hanno completo accesso alle funzionalità del dispositivo

Totale accesso



Native



Cross



Hybrid



PWA

Accesso limitato



Web



# Dilemma

- Ci sono vari fattori che vanno considerati
- Scegliere avendo un occhio di riguardo al futuro e al mercato
- Scegliere sulla base delle risorse disponibili (umane ed economiche)



Grazie!

A questo [link](#) troverete il capitolo 3 della tesi in cui ho approfondito le varie categorie delle applicazioni, discusse nella giornata di oggi.

Grazie ancora per la Vostra attenzione!

