

# ICT | SOFTWARE



## Una scelta vincente

DI ALBERTO BONACINA\*

**Q**uali sono le sfide al giorno d'oggi nello sviluppo di un software? Su quali piattaforme è meglio puntare? Dove sono i nostri potenziali utenti e come possiamo raggiungerli? C'è una tecnologia che ci può aiutare in tutto questo? Queste sono alcune delle domande che ogni giorno ingegneri e sviluppatori si fanno quando vogliono iniziare un nuovo progetto di sviluppo o aprirsi ad un nuovo

mercato e a cui cercheremo di dare una risposta.

### CONTESTO DI RIFERIMENTO

Il mercato a cui ci avviciniamo è mai come in questo momento frammentato rispetto a quello che poteva essere solo 10 anni fa. Il mobile non era così diffuso, smartphone e tablet cominciavano ad entrare nella nostra quotidianità, il desktop, nella maggior parte dei casi Windows, era la piattaforma principale per gli utenti e il web si stava affermando come la nuo-

va piattaforma su cui puntare ma che non aveva ancora prestazioni e supporto hardware come al giorno d'oggi.

I numeri ci dicono invece che oggi il mobile ha superato il desktop in market share con percentuali che si attestano ad un 65% contro il 35%, con picchi di oltre l'80-90% nei paesi in via di sviluppo dove lo smartphone è l'unico dispositivo posseduto per accedere ad app e servizi, il web è una piattaforma ormai più che matura e che permette di arrivare a chiunque nel

mondo con prestazioni che non erano immaginabili fino a qualche tempo fa. In questo scenario sviluppare un'applicazione o servizio che non tenga conto di tutte le piattaforme utilizzate dagli utenti ci può precludere una buona fetta di mercato, che in certi ambiti può essere anche considerevole.

Il metodo prevalente per sviluppare applicazioni su diverse piattaforme come Android e iOS per dispositivi mobili, il Web, Windows, Mac (e Linux) per computer desktop, è stato tradizionalmente

conosciuto come "approccio nativo". In questo approccio, ogni applicazione viene sviluppata specificamente per il sistema operativo di destinazione, utilizzando il linguaggio di programmazione, gli strumenti di sviluppo, le convenzioni e i processi propri di quel sistema. Tuttavia, questa pratica conduce inevitabilmente alla duplicazione del codice, degli sforzi di sviluppo e coinvolge team di sviluppatori distinti, comportando un aumento correlato dei costi e dei tempi di sviluppo. In aggiun-

### Flutter e lo sviluppo multiplatforma



**TECNOLOGIA  
SOLUZIONI  
PRODOTTI  
PRESTAZIONI**



## SAFE BUILDING

### FRCM CRM SYSTEM

Una gamma completa di sistemi certificati con CVT per il rinforzo strutturale e antisismico.

Il gruppo **G&P intech** vanta una storia unica e riconosciuta nel panorama nazionale delle **tecnologie antisismiche** e di **ristrutturazione per la ricostruzione post eventi calamitosi** e per la prevenzione del costruito esistente. Un partner ideale per competenza, affidabilità, innovazione, per Pubbliche Amministrazioni, progettisti, imprese di costruzioni. Un servizio tecnico di consulenza ingegneristica e assistenza in cantiere su tutto il territorio nazionale.

[www.gpintech.com](http://www.gpintech.com)



ta, c'è il costo inevitabile della sincronizzazione tra tutti questi progetti paralleli, che devono progredire simultaneamente per evitare ritardi su una piattaforma e garantire agli utenti l'accesso a tutte le funzionalità dell'applicazione, indipendentemente dalla piattaforma utilizzata.

Per affrontare tali problematiche nel corso degli anni, sono state proposte diverse soluzioni che consentono di scrivere il codice sorgente una sola volta utilizzando un singolo linguaggio di programmazione e successivamente eseguirlo su diverse piattaforme, seguendo il principio del "write once and run everywhere". Un esempio significativo nel contesto delle applicazioni lato server è rappresentato storicamente da Java, che, grazie alla sua Java Virtual Machine (JVM), permette lo sviluppo di applicazioni in grado di funzionare su qualsiasi sistema operativo dotato della JVM, senza la necessità di preoccuparsi del sistema operativo specifico in uso. Questo approccio quindi porta a scrivere applicazioni non pensando al sistema operativo su cui viene lanciata perché si utilizza sempre un layer di astrazione tra il sistema operativo e la nostra applicazione. Vedremo nei prossimi paragrafi un'altra soluzione software che consente di scrivere applicazioni multipiattaforma lato client, quindi ad interfaccia grafica, per i sistemi operativi introdotti precedentemente, scrivendo il codice sorgente in un solo linguaggio di programmazione.

#### VANTAGGI DELLO SVILUPPO MULTIPIATTAFORMA CON FLUTTER

Flutter, framework open source di Google, consente agli Ingegneri del Software di scrivere applicazioni di alta qualità, compilate in modo nativo per tutte le 6 piattaforme citate precedentemente: Android, iOS, Web, Windows, macOS e Linux scrivendo il codice una sola volta, in linguaggio Dart, ottimizzando gli sforzi e snellendo il processo di sviluppo, riducendo il time-to-market e garantendo un'esperienza utente coerente su tutti i dispositivi.

Siamo consapevoli che il tempo rappresenta la risorsa più limitata in ogni progetto e che è essenziale cercare di ridurlo al minimo pur mantenendo elevata la qualità del prodotto. Flutter, con le sue caratteristiche, si propone di ottimizzare la gestione dello sviluppo delle applicazioni, traducendosi direttamente in minori costi e tempi di produzione del software. Flutter aumenta l'efficienza consentendo la scrittura di una singola base di codice che offre diversi vantaggi: elimina la duplicazione di codice e sforzi nello sviluppo di funzionalità specifiche, previene la disparità di funzionalità tra diverse piattaforme, evitando che una piattaforma sia più avanti nello sviluppo rispetto a un'altra. Inoltre, impedisce discrepanze visive nell'applicazione su diverse piattaforme, consentendo agli utenti di utilizzare il software su qualsiasi dispositivo senza compromettere la loro esperienza. Infine, favorisce

la collaborazione tra sviluppatori, elimina la necessità di più team di sviluppo e la necessità di switchare continuamente tra ambienti di sviluppo e riducendo gli inefficaci "context-switch", migliorando così la produttività complessiva.

#### SFIDE E CONSIDERAZIONI

Come si dice sempre - non è sempre tutto rose e fiori - e naturalmente la creazione di un'applicazione con un unico codice sorgente destinato a diverse piattaforme può presentare delle sfide, se queste non vengono affrontate con la giusta strategia, potrebbero emergere problemi significativi durante il processo di sviluppo.

Un aspetto iniziale da considerare è legato alle dimensioni dei dispositivi che rappresentano queste piattaforme: prevalentemente smartphone per Android e iOS, e in gran parte PC per Windows, Linux e macOS. Questa distinzione è fondamentale poiché orienta la progettazione dell'interazione e dell'interfaccia utente in modo diverso, tenendo conto delle dimensioni dello schermo sui quali i nostri utenti utilizzano l'applicazione. Non da ultimo, va menzionato lo "strumento" con cui interagiscono, principalmente attraverso le dita nel caso degli smartphone, mentre tramite mouse e tastiera nel contesto dei PC.

Un secondo aspetto da prendere in considerazione riguarda l'eterogeneità dell'hardware con cui dobbiamo interagire. Nei dispositivi smartphone, l'hardware è integrato e comprende una varietà considerevole di sensori. Al contrario, nelle piattaforme desktop, l'hardware è composto da un insieme di periferiche collegate al computer. Come affrontare quindi tutto questo portando a casa il risultato migliore? Un primo approccio potrebbe consistere nello sviluppo di un'applicazione dedicata a piattaforme affini, come Android e iOS, e un'altra dedicata alle piattaforme desktop. In questo modo, sarebbe possibile effettuare scelte mirate basate su fattori quali le dimensioni dello schermo, la mobilità degli utenti o le specifiche dell'hardware disponibile. Alternativamente, si potrebbe adottare un approccio unificato, creando un'applicazione con un'interfaccia utente (UI) flessibile che si adatta alle dimensioni dello schermo in esecuzione, seguendo il concetto di "responsive design" noto nel campo dei siti web. In questo scenario, alcune funzionalità potrebbero essere abilitate solo su determinate piattaforme, ottimizzando così l'utilizzo dell'hardware disponibile.

Ecco un esempio semplificato che rende l'idea: nella versione mobile dell'applicazione, consentire la cattura di foto e video tramite la fotocamera del dispositivo, mentre nella versione desktop offrire la possibilità di effettuare successivamente modifiche più avanzate e precise utilizzando strumenti dedicati.

#### PROGETTI ED ESPERIENZA SUL CAMPO

Ecco alcuni esempi di applicazioni



sviluppate in cui sono stato direttamente coinvolto spaziano dalla gestione magazzino alla gestione di carte fedeltà fino a un POC (proof of concept) per una grande realtà italiana nella logistica e spedizione. Queste 3 applicazioni sono state sviluppate direttamente con Flutter integrando degli SDK (Software Development Kit) per interfacciarsi direttamente e in modo nativo con l'hardware a disposizione andando a creare un collegamento efficiente tra hardware e software.

Nella prima applicazione è stato integrato un modulo per la lettura di codici a barre e barcode per permettere la gestione magazzino direttamente da smartphone iOS senza la necessità di acquisto di device industriali dedicati; la

particolarità di questo progetto è legata al fatto che l'intero sviluppo è stato fatto su piattaforma Windows e Linux creando le rispettive applicazioni native e solo nell'ultima fase di testing sul campo è stato utilizzato un Mac con un device iOS.

Per gestire le tessere fedeltà è stata sviluppata una applicazione Web, fruibile sia da smartphone che da desktop, con la quale è possibile scansionare il codice presente sulle tessere e associarle al proprio profilo utente. Per effettuare la scansione si può utilizzare la fotocamera del dispositivo sia con Android che con iOS, o la webcam del PC nel caso di visualizzazione su Desktop, il tutto utilizzando la stessa base di codice per sfruttare l'hardware pre-

sente sulle diverse piattaforme. Infine per il POC è stata sviluppata una applicazione Android che può essere utilizzata sia su smartphone che su smartwatch, che si interfaccia con un hardware dedicato per le operazioni di picking in magazzino, e che modifica la sua visualizzazione e le sue funzionalità a seconda della dimensione dello schermo del device su cui viene eseguita; anche in questo caso la base di codice è la stessa e lo studio della UI è stato realizzato visualizzando l'applicazione con diversi fattori di forma come applicazione nativa Linux.

#### CONCLUSIONI

Gestire progetti di sviluppo rappresenta un compito complesso, poiché coinvolge una vasta gamma di tecnologie, standard e aspettative degli utenti riguardo alle funzionalità e all'usabilità di una soluzione software specifica. Nel corso del tempo, abbiamo osservato l'evoluzione di approcci diversi per affrontare la sfida dello sviluppo. Inizialmente, si è iniziato con lo sviluppo nativo, ma ormai sembra consolidarsi l'ambiente multipiattaforma in cui Flutter appare come lo stato dell'arte più promettente.

\*PRESIDENTE COMMISSIONE ICT ORDINE INGEGNERI DI BERGAMO

**concrete**  
structural engineering software

**Sismicad 12**  
un passo avanti

**Sismicad**  
Tante funzionalità un unico software

Scopri tutte le offerte su [www.concrete.it](http://www.concrete.it)

**Sismicad**  
Sostenitore  
Ingegneria Strutturale Italiana