

## Cross Platform vs. Native Mobile Programming

### Sommario

Cross Platform vs. Native Mobile Programming.....	1
Introduzione .....	1
Native Mobile Programming.....	2
Meccanismo delle app native .....	2
Linguaggi di programmazione native.....	2
Ambienti di sviluppo native .....	3
Cross-platform:.....	4
Tipi di applicazioni cross-platform.....	4
Linguaggi di programmazione cross-platform.....	5
Ambienti di sviluppo cross-platform.....	6
Itis App: esempio di applicazione cross-platform .....	7
Sitografia .....	8

### Introduzione

Per creare un'applicazione bisogna tenere conto di molte variabili che entrano in gioco e che vanno ad incidere su tempi, costi e delle sue prestazioni.

Il mercato delle applicazioni mobili sta acquisendo sempre più importanza. Nel 2008 questo mercato non esisteva, mentre 8 anni dopo valeva già 1300 miliardi di dollari. Le stime vedono questo mercato in crescita nei prossimi anni e probabilmente il suo valore si aggirerà intorno ai 6350 miliardi di dollari rendendolo il terzo più grande. Gli app store di Apple e Android sono un ecosistema di circa 6,5 milioni di app in cui cinesi e statunitensi dominano dal punto di vista dei guadagni. Nel nord Europa le competenze tecniche e tecnologiche (come le connessioni 4G) si sono diffuse in maniera meno frammentata e più veloce rispetto al sud Europa e questo ha consentito un maggiore sviluppo, rendendo le aziende protagoniste di questo mercato.

Il nostro obiettivo è quello di fare chiarezza sulla tipologia con cui sia meglio approcciarsi alla programmazione mobile poiché lo sviluppo app è un'attività complessa in termini di competenze e, per questo, occorre fare un confronto fra chi commissiona e chi sviluppa per capire quale sia la soluzione migliore, scegliendo tra due tipologie di app esistenti: le native e le cross-platform.

## ***Native Mobile Programming***

Le app native sono una categoria di applicazioni create utilizzando la tipologia nativa che raggruppano un insieme di istruzioni informatiche progettate con lo scopo di rendere possibile un servizio o una serie di servizi o strumenti ritenuti utili o desiderabili dall'utente, creata appositamente per uno specifico sistema operativo utilizzando linguaggi specifici (es. Java per Android e Swift per iOS).

I vantaggi dell'utilizzo della programmazione native sono l'utilizzo di un solo linguaggio di programmazione, la maggiore visibilità negli app store del sistema operativo scelto, la maggiore professionalità offerta dall'app e la possibilità di avere prestazioni elevate.

Gli svantaggi di questa programmazione sono la compatibilità limitata ad un solo sistema operativo, i maggiori costi di produzione, la difficoltà nella gestione e la necessità di una continua lavorazione.

### **Meccanismo delle app native**

Le app, vanno ad ampliare le capacità native del dispositivo implementate all'interno del sistema operativo. L'interazione diretta con le API messe a disposizione dal costruttore del sistema operativo garantirà accesso a tutte le funzionalità del dispositivo. Inoltre permetterà prestazioni ottimali e migliorerà moltissimo l'usabilità del dispositivo.

### **Linguaggi di programmazione native**

#### ***Android SDK***

Lo sviluppo di software Android è il processo attraverso cui vengono costruite nuove applicazioni per il sistema operativo Android. Le app sono in genere sviluppate in linguaggio Java utilizzando il kit di sviluppo software Android (SDK), nonostante siano disponibili anche altri ambienti di sviluppo.

Android SDK include un grande insieme di strumenti di sviluppo. Le piattaforme per lo sviluppo attualmente supportate sono tutti i computer che eseguono Linux, macOS 10.5.8 o versioni successive e Windows XP o successivo. A partire da marzo 2015, l'SDK non è disponibile su Android in sé, ma lo sviluppo del software rimane possibile utilizzando applicazioni Android specializzate.

Fino alla fine del 2014 l'ambiente di sviluppo integrato ufficialmente supportato è stato Eclipse utilizzando gli Android Development Tools Plugin, anche se ad esempio gli ambienti IntelliJ IDEA e NetBeans supportano pienamente lo sviluppo di Android. A partire dal 2015 Android Studio, realizzato da Google e alimentato da IntelliJ, è l'IDE ufficiale di Android.

I miglioramenti che vengono via via apportati all'SDK vanno di pari passo con l'evoluzione del sistema operativo Android. L'SDK supporta anche le versioni precedenti del Sistema Operativo nel caso in cui gli sviluppatori delle applicazioni necessitassero di eseguire le loro applicazioni su dispositivi che supportano versioni Android più vecchie. Gli strumenti di sviluppo sono tutti scaricabili, pertanto dopo aver scaricato l'ultima versione dell'ambiente di sviluppo e della piattaforma, quelle meno recenti possono essere scaricate per eseguire ad esempio dei test di compatibilità.

Una qualsiasi applicazione Android ha estensione .apk e viene memorizzata nella cartella /data/app del dispositivo.

## ***Swift***

Swift è un linguaggio di programmazione orientata agli oggetti per sistemi macOS, iOS, watchOS, tvOS e Linux, che è stato presentato da Apple nel 2014. Swift è stato concepito per coesistere con il linguaggio Objective-C (C orientato agli oggetti), tipico degli sviluppi per i sistemi operativi Apple, attraverso la semplificazione della scrittura del codice e alla maggiore resistenza agli errori all'interno del codice.

Swift è stato iniziato a sviluppare nel 2010. Il linguaggio di programmazione prende spunti da Objective-C, Python, C# e, citando il creatore Chris Lattner, "da troppi altri per elencarli tutti".

La prima versione di Swift è stata rilasciata il 9 settembre 2014, preceduta da alcune versioni beta distribuite per gli sviluppatori Apple.

Molto recentemente, il 25 marzo 2019, è stata pubblicata la sua versione 5.0 sotto licenza Apache 2.0 per i sistemi operativi Apple e Linux.

## **Ambienti di sviluppo native**

### ***Android Studio***

È un ambiente di sviluppo integrato per lo sviluppo all'interno della piattaforma Android. È relativamente recente essendo stato annunciato per la prima volta il 16 maggio 2013 e lanciato ufficialmente nel dicembre 2014.

Android Studio viene distribuito gratuitamente sotto licenza Apache 2.0.

L'ambiente di sviluppo supporta linguaggi di programmazione IntelliJ e Clion, come Java e C++, e i linguaggi con estensioni, come Go. Mentre IntelliJ, su cui è basato Android Studio, supporta tutte le versioni Java rilasciate e Java 12, non è chiaro quale versione di Java sia supportata da Android Studio. La documentazione ufficiale menziona il supporto parziale di Java 8, ma consente l'utilizzo delle nuove funzionalità linguistiche di Java 12.

### ***Cocoa***

È un ambiente di sviluppo nativo per lo sviluppo di applicazioni che girano sui sistemi operativi macOS e iOS. Le applicazioni create tramite Cocoa vengono sviluppate con l'utilizzo degli strumenti di sviluppo messi a disposizione direttamente da Apple, XCode e Interface Builder.

Cocoa utilizza come linguaggi di programmazione Objective-C e Swift, tipici della programmazione per sistemi operativi Apple, ma tramite l'utilizzo di vari tools è reso possibile anche l'utilizzo di altri linguaggi come ad esempio Python, Object Pascal o AppleScript.

Questo ambiente di sviluppo è la deriva da ambienti di sviluppo come NeXTStep e OneStep, sviluppati da NeXT negli anni '80 e '90. Dopo l'acquisto da parte di Apple nel '96 è stato messo in secondo piano, per poi essere utilizzato come base di OneStep per librerie e supporto binario. Con la razionalizzazione di Steve Jobs è stato interrotto per poi essere concesso in licenza a terze parti.

## **Cross-platform:**

Le app cross platform sono un tipo di applicazioni create per essere compatibili con tutte le tipologie di sistema operativo utilizzando un linguaggio di programmazione universale.

I vantaggi della programmazione cross platform sono la compatibilità con sistemi operativi differenti, l'utilizzo di linguaggi universali (es. html5), i costi minori e lo sviluppo in minor tempo.

Gli svantaggi di questa programmazione sono la minore visibilità negli app store, la maggiore esposizione a bug e crash e il bisogno di una costante manutenzione.

## **Tipi di applicazioni cross-platform**

### ***Applicazioni ibride***

Le applicazioni ibride sono sviluppate utilizzando framework web puri per dispositivi mobile, i quali attraverso l'utilizzo di linguaggi web come HTML, JavaScript e CSS permettono di costruire l'interfaccia utente e la logica dell'applicazione, in combinazione con framework che rendono possibile l'accesso alle funzionalità native dei dispositivi su cui le applicazioni girano. Ad esempio, rendono possibile l'accesso a funzionalità hardware come il GPS e la fotocamera attraverso delle chiamate in JavaScript. Un esempio di framework che può accedere a queste funzionalità è Apache ***Cordova***.

### ***Applicazioni con codice interpretato***

Le applicazioni a codice interpretato sono invece sviluppate utilizzando framework che non utilizzano il motore integrato per interpretare il codice JavaScript, bensì compilano il codice sorgente in codice macchina, eseguibile quindi dal dispositivo. Alcuni di questi framework, utilizzano linguaggi web per produrre l'interfaccia utente dell'applicazione, mentre la logica viene sviluppata in linguaggi di programmazione differenti a seconda dell'utilizzo. Un esempio di questi framework è ***Xamarin***.

### ***Applicazioni tradotte***

Questo tipo di applicazioni sono create sviluppando inizialmente un'applicazione con tecnologia nativa per un'unica piattaforma. Dopodiché, tramite l'utilizzo di alcuni tool, il codice sorgente viene tradotto in un linguaggio utilizzabile da parte di un'altra piattaforma.

## Linguaggi di programmazione cross-platform

### *HTML*

HTML è un linguaggio utilizzato per la strutturazione delle pagine web. Ad ottobre 2014 è stata pubblicata l'ultima versione del linguaggio: HTML 5.

Lo sviluppo di questa versione venne avviato dal gruppo di lavoro WHATWG, che pose come obiettivo quello di progettare delle specifiche per lo sviluppo di applicazioni web, focalizzandosi su miglioramenti e aggiunte ad HTML e alle tecnologie correlate.

Inizialmente in contrasto con il World Wide Web Consortium, il quale ha poi riconosciuto valido il nuovo linguaggio, annunciando di creare un apposito gruppo per la standardizzazione dell'HTML5.

Dal 2007 il WHATWG ha collaborato con il W3C in tale processo di standardizzazione, per poi decidere nel 2012 di separarsi dal W3C, creando di fatto due versioni dell'HTML5: la versione definita come "HTML Living Standard" e quindi in continua evoluzione, e quella del W3C che è un'unica versione corrispondente ad uno "snapshot" del Living Standard.

La prima versione dello standard è stata pubblicata dal W3C come Recommendation il 28 ottobre 2014, mentre la versione HTML 5.1 è stata pubblicata il 1° novembre 2016.

Le novità introdotte da HTML5 rispetto al suo predecessore (HTML4) sono finalizzate soprattutto a migliorare il disaccoppiamento fra struttura, caratteristiche definite dalle direttive di stile, e contenuti di una pagina web, definiti dal testo vero e proprio.

Inoltre l'HTML5 prevede il supporto per la memorizzazione locale di grandi quantità di dati scaricati dal web browser, per consentire l'utilizzo di applicazioni basate su web (come per esempio le caselle di posta di Google o altri servizi analoghi) anche in assenza di collegamento a Internet.

### *JavaScript*

JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi. Comunemente utilizzato nella programmazione Web lato client, per la creazione, in siti web e applicazioni web, di effetti dinamici ed interattivi tramite apposite funzioni di script invocate da eventi innescati a loro volta dall'utente stesso sulla pagina web in uso.

Tali funzioni di script possono essere opportunamente inserite in file HTML, in pagine JSP o in appositi file separati con estensione .js richiamati poi per l'esecuzione dell'evento.

Ultimamente il suo campo di utilizzo è stato esteso alla programmazione cross-platform, con le quali è possibile creare app per più sistemi operativi utilizzando un unico codice sorgente basato appunto su JavaScript, HTML e CSS.

## **Ambienti di sviluppo cross-platform**

### ***Cordova***

Apache Cordova è un framework di sviluppo mobile open source utilizzato per la creazione di applicazioni per i sistemi operativi Windows, iOS e Android. Consente di utilizzare tecnologie web standard come HTML5, CSS3 e JavaScript per lo sviluppo multipiattaforma, evitando l'utilizzo di un linguaggio di sviluppo nativo per ogni singola piattaforma mobile.

Le applicazioni generate da questo framework non possono né considerarsi puramente native, né basate completamente sul web. Infatti Le applicazioni vengono eseguite all'interno di wrapper destinati a ciascuna piattaforma, basandosi su associazioni API conformi agli standard per accedere ai vari sensori, ai dati e allo stato della rete di ogni dispositivo mobile.

Visual Studio mette a disposizione i tools di Apache Cordova necessari per lo sviluppo delle varie applicazioni, estendendo l'ambiente di sviluppo ad Android, iOS e Windows.

### ***Sviluppo del file.apk***

Per sviluppare il file.apk in Cordova bisogna seguire queste fasi:

- Crea il file keystore per firmare l'apk.
- Crea il file delle proprietà.
- Imposta versione, codice versione e nome pacchetto.
- Versione Android di Google

### ***Xamarin***

"Xamarin" è un'azienda produttrice di software statunitense con sede a San Francisco. Fondata nel maggio 2011 da ingegneri provenienti da infrastrutture di linguaggio comune e di Common Language Specifications (spesso chiamato Microsoft .NET).

Utilizzando un codice basato principalmente su C#, gli sviluppatori possono usare gli strumenti messi a disposizione da Xamarin, per scrivere applicazioni di tipo nativo con sistema operativo Android, iOS e Windows con interfacce utenti native e condividere inoltre il codice su diverse piattaforme.

"Xamarin.Android" e "Xamarin.iOS" sono due componenti di xamarin che rendono possibile uno sviluppo pseudo-nativo per i sistemi operativi Android, iOS e Windows, utilizzando Visual Studio o Xamarin Studio, dove gli sviluppatori scrivono il codice che verrà poi condiviso e compilato da nativo nelle varie piattaforme in cui verrà installata l'applicazione.

### ***Sviluppo del file.apk***

Per sviluppare il file.apk su Xamarin bisogna seguire le seguenti fasi:

- Specificare l'icona dell'app
- Specificare la versione dell'app
- Compattare il file.apk
- Proteggere l'applicazione
- Impostare le proprietà per la creazione dei pacchetti
- Compilare
- Archiviare per la pubblicazione

## Itis App: esempio di applicazione cross-platform

### *Cos'è Itis App?*

Itis App è un esempio di applicazione cross-platform di tipo ibrido, sviluppata con l'intento di dare un esempio pratico delle tematiche trattate e di facilitare l'accesso a specifiche aree del sito, come circolari e orari, utili a tutti i membri della scuola.

### *Realizzazione e funzione*

L'app, essendo un'applicazione di tipo ibrido, viene realizzata con l'utilizzo dei linguaggi HTML5 e CSS3 e sviluppata tramite i tools di Apache Cordova messi a disposizione da Visual Studio.

L'uso del linguaggio HTML5 permette di collegarsi direttamente alle zone del sito richieste con l'utilizzo di tag href che reindirizzano all'url specificato al momento del tocco.

Il linguaggio CSS3, invece, consente la possibilità di creare uno stile grafico di vario per la presentazione dell'interfaccia utente, rendendo il tutto più bello e affascinante da vedere.

In allegato viene illustrato una parte del codice html che svolge i collegamenti alle varie zone disponibili.

```

index.css  index.html  ×
14         <link rel="stylesheet" type="text/css" href="css/index.css">
15
16         <title>ITIS APP</title>
17     </head>
18
19     <body>
20
21         <div id="titolo">
22             
23         </div>
24
25         <nav>
26             <ul>
27                 <li>
28                     <a>COMUNICAZIONI</a>
29                     <ul>
30                         <li><a href="https://dati.itis.pr.it/avvisi/avvisialunni/alunni/ListaDocumenti.shtm">AVVISI STUDENTI</a></li>
31                         <li><a href="https://dati.itis.pr.it/avvisi/avvisidocenti/docenti/ListaDocumenti.shtm">AVVISI DOCENTI</a></li>
32                         <li><a href="https://dati.itis.pr.it/avvisi/avvisiata/ata/ListaDocumenti.shtm">AVVISI PERSONALE ATA</a></li>
33                     </ul>
34                 </li>
35
36                 <li>
37                     <a>ORARI</a>
38                     <ul>
39                         <li><a href="https://dati.itis.pr.it/orariodiurno/">ORARIO DIURNO</a></li>
40                         <li><a href="https://dati.itis.pr.it/orario%20serale/">ORARIO SERALE</a></li>
41                     </ul>
42                 </li>
43             </ul>
44         </nav>
45

```

## **Sitografia**

*Utilizzando i seguenti link, è possibile ricercare nel dettaglio le risorse da noi utilizzate per lo sviluppo di questo approfondimento.*

### ***Sviluppo Crossplatform vs native:***

- <https://www.cellulare-magazine.it/lo-sviluppo-di-un-app-mobile-approccio-nativo-vs-cross-platform/>
- <https://medium.com/@andrei.klubnikin/cross-platform-vs-native-mobile-app-development-choosing-the-right-dev-tools-for-your-app-project-47d0abafee81>

### ***Xamarin + Cordova:***

- <https://www.clarity-ventures.com/services/custom-development/xamarin-vs-titanium-vs-phonegap-vs-cordova-a-comparison>
- <https://cordova.apache.org/docs/en/4.0.0/guide/overview/index.html>
- <https://www.codemag.com/article/1703031/Xamarin-versus-Cordova>

### ***Caratteristiche Crossplatform:***

- [https://en.wikipedia.org/wiki/Cross-platform\\_software](https://en.wikipedia.org/wiki/Cross-platform_software)

### ***Sviluppo app native:***

- <https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>

### ***App mobile:***

- <https://www.corriere.it/dataroom-milena-gabanelli/apple-google-mercato-app-2021-terza-economia-mondiale-smartphone/c7e5f46c-f26e-11e8-9ee1-95c4f8c44f3b-va.shtml>
- <https://www.mobileworld.it/2018/07/17/mercato-app-continua-crescere-34-miliardi-dollari-6-mesi-2018-185133/>

### ***Ambiente di sviluppo android studio:***

- [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)

### ***Linguaggio e ambiente di sviluppo per swift:***

- [https://it.wikipedia.org/wiki/Swift\\_\(linguaggio\\_di\\_programmazione\)](https://it.wikipedia.org/wiki/Swift_(linguaggio_di_programmazione))

### ***Altro:***

- <https://skelia.com/articles/general-overview-of-cross-platform-languages/>

### ***Cocoa:***

- [https://it.wikipedia.org/wiki/Cocoa\\_\(software\)](https://it.wikipedia.org/wiki/Cocoa_(software))

### ***HTML 5:***

- <https://it.wikipedia.org/wiki/HTML5>

### ***JavaScript:***

- <https://it.wikipedia.org/wiki/JavaScript>

### ***Cordova:***

- [https://it.wikipedia.org/wiki/Apache\\_Cordova](https://it.wikipedia.org/wiki/Apache_Cordova)

### ***Xamarin:***

- <https://it.wikipedia.org/wiki/Xamarin>