

Sviluppo mobile: primi passi con iOs

Esaminiamo i principali ambienti per tablet e smartphone partendo dall'ecosistema di Apple.



Il mercato delle applicazioni mobili cresce continuamente e diventa sempre più interessante. Di conseguenza numerosi sviluppatori passano sulle nuove piattaforme - o sono tentati di farlo - per trovare mercato. Nuove piattaforme implicano differenti abitudini, strumenti di sviluppo, contratti di distribuzione e così via, per questo iniziamo con questo numero una piccola serie, esaminando gli ambienti più diffusi, per aiutare gli sviluppatori a scegliere la propria strada.

Considerazioni di strategia

Le piattaforme mobili sono diverse e richiedono strumenti di sviluppo e linguaggi specifici. Nel frattempo, le applicazioni Html hanno acquistato in potenza e programmabilità, quindi appare conveniente sviluppare usando Html5 ed essere agnostici rispetto all'ambiente hardware. Prima di partire con l'analisi delle piattaforme native, conviene valutare a fondo questa scelta strategica e le situazioni in cui è opportuna. Vediamo prima i motivi a favore. In primo luogo, le prestazioni delle Cpu nei cellulari sono sempre più elevate, la velocità degli interpreti Javascript in tutti i browser è cresciuta a livelli impensabili alcuni anni fa e il modello di programmazione di un

browser Html5 è ricco e sfruttabile, con funzioni multimediali soddisfacenti e persino accesso ai sistemi di localizzazione Gps. C'è ovviamente un «però», almeno secondo le società che hanno iniziato con app multiplatforma, per tornare sui propri passi e scegliere lo sviluppo nativo. L'esempio più importante è Facebook, che da quando è passato a una app nativa è migliorato parecchio in funzionalità. Zuckerberg ha detto «scommettere tutto su Html5 è stato uno degli errori strategici principali che abbiamo fatto, se non il maggiore». Di recente un'altra startup ha fatto inversione di marcia e c'è un articolo che discute i motivi della scelta con equanimità (tnw.co/13ccfxZ), che merita una lettura approfondita.

Esaminando ciò che troviamo nei report sul web e quello che raccogliamo dalle nostre conoscenze, si conclude che la soluzione Html5 permette di iniziare

più velocemente, ma porta a potenziali limitazioni nello sviluppo quando l'applicazione si fa più complessa e a scontrarsi con problemi di prestazioni. È un fatto che lo sviluppo di un'applicazione fatta da migliaia di righe di codice Javascript è meno strutturato e assistito, rispetto allo sviluppo nativo e i problemi di gestione diventano sempre più avvertibili, man mano che l'applicazione cresce.

Insomma, lo sviluppo Html5 «richiederebbe costi e tempi maggiori rispetto a un'applicazione nativa», e lavorare con Html5 e lo stesso livello di prestazioni e funzionalità, richiederebbe più competenza, lavoro e attenzione.

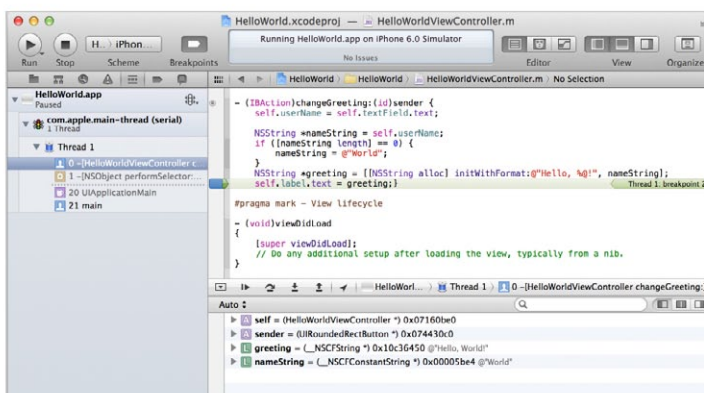
Prendiamo per buone queste indicazioni e scegliamo la strada nativa, considerando anche che giocare secondo le regole aiuta a entrare negli Store, specie se, come nel caso di Apple, ci sono guardie all'ingresso.

L'ambiente di sviluppo

Per creare applicazioni iOs, lo strumento di riferimento è XCode, l'ambiente di sviluppo di Apple, elegante,



Si può sviluppare un'applicazione e provarla nel simulatore senza spendere un centesimo.



lineare, ricco di funzioni come una macchina tedesca, con la pulizia dell'interfaccia che ci si aspetta da un prodotto Apple e una curiosa somiglianza dell'interfaccia con iTunes, che sta probabilmente a suggerire che lo sviluppo di applicazioni può essere ricreativo.

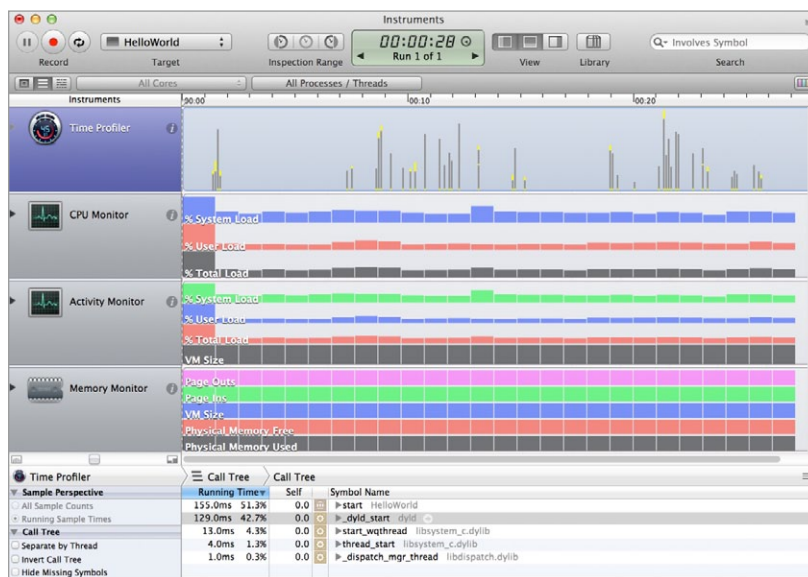
Come altre applicazioni Apple anche XCode è una miniera di funzionalità, nascoste dietro un'apparente austera semplicità, non un cruscotto d'aereo irto di pulsanti, come Eclipse.

Diamo per scontate tutte le funzioni che fanno un ambiente integrato di sviluppo dei nostri tempi, come un editor molto evoluto con avanzate opzioni di autocompletamento. Segnaliamo, fuori dalla norma, la sua singolare capacità di analisi del codice a run time.

Si possono raccogliere dati di profilazione in modo automatico e visualizzare una traccia del carico sulla Cpu in funzione del tempo. Un clic su un picco di elaborazione ci può portare nel codice in esecuzione al momento. Insomma, chi non si tira indietro può arrivare in pochi tentativi a mettere le mani nel punto giusto per rimuovere un collo di bottiglia.

Si possono confrontare tracce di carico di elaborazione diverse, come se fossero tracce audio, per paragonare gli effetti ottenuti modificando il codice e naturalmente è possibile registrare un'interazione con programma e replicarla, in modo da poter avere dati ripetibili.

Dato che OS X è Unix, parecchie delle abitudini consolidate di uno sviluppatore Unix incontrano il supporto dell'editor. Non ci sono problemi, quindi, per interfacciarsi con un repository git per l'archiviazione dei



XCode ha capacità di analisi delle prestazioni fuori dal comune. Ecco l'analisi della traccia di un'esecuzione rispetto a diversi indicatori, effettuata con Instruments.

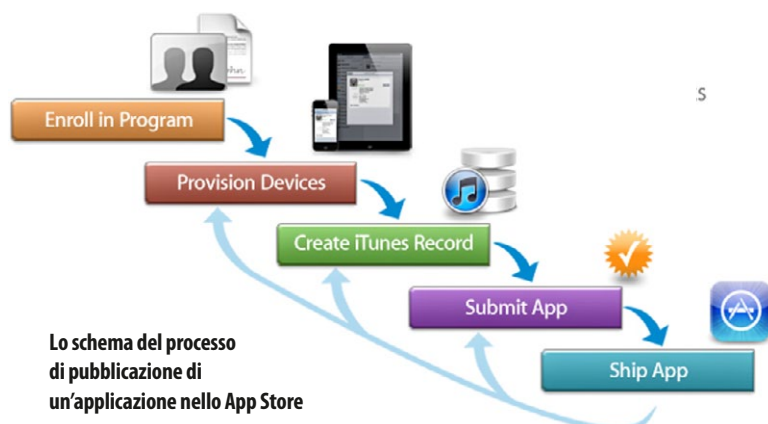
sorgenti, si può attivare l'editor da una shell e si riesce a lavorare facilmente con codice open source sviluppato con editor e riga di comando.

Naturalmente, la costruzione interattiva di interfacce e il collegamento visuale di eventi di interfaccia con sezioni di codice sono gestiti in modo eccellente, rivelando la discendenza Smalltalk dell'ambiente di sviluppo. Dallo strumento per la creazione di interfacce è possibile trascinare eventi visuali e collegarli a sezioni di codice in modo strettamente visuale. Diversi anni fa IBM aveva un ambiente di sviluppo Java in cui era altrettanto piacevole creare un'interfaccia utente, VisualAge, che è stato soppiantato da Eclipse. Chiudiamo questa breve

introduzione segnalando un video (bit.ly/Ytniv0) che mostra la creazione di un'interfaccia utente ridotta all'osso nei tre ambienti di sviluppo principali per sistemi operativi mobili: Visual Studio, XCode e Eclipse. Visual Studio è il vincitore in questo confronto, ma c'è più di «hello world» nel mondo dello sviluppo, quindi il risultato non va preso come un giudizio definitivo. Sinteticamente, a noi pare che ci siano molte ragioni per amare lo sviluppo con XCode, a cui se ne aggiunge una pragmatica: XCode si scarica gratuitamente dall'App Store di Apple.

Il kit di sviluppo per telefoni cellulari contiene un simulatore, che permette di sviluppare e testare applicazioni anche senza avere un telefono a disposizione.

COME SI PUBBLICA SULL'APP STORE



I linguaggi

La soluzione principe delle piattaforme Apple è Objective C, una versione estesa del C creata più o meno durante gli anni in cui Bjarne Stroustrup stava lavorando al C with classes, che poi diventò il C++.

Il C++ ha preso la strada di Simula, implementando proprietà e metodi come variabili e funzioni, che sono risolti dal linker in indirizzi assoluti.

Objective C ha preso la strada di Smalltalk, realizzando i metodi come messaggi ricevuti da un oggetto e gestiti attivando funzioni locali.

Sotto il cofano c'è sempre il linguaggio C, quindi le classi Objective C hanno uno strato di interfaccia che si occupa di ricevere messaggi e li traduce nell'attivazione del codice binario corrispondente, in modo vagamente simile al Dcom su Windows e a Rmi in Java.

La mancanza di un accoppiamento diretto fra oggetti, risolto dal linker, richiede un po' più tempo nell'elaborazione dell'invio di un messaggio, ma implica, per esempio, che un oggetto possa ricevere un messaggio che non è in grado di gestire senza che succeda nulla di male, se pure non accade nulla di positivo.

Per il resto, l'efficienza è quella che ci si aspetta dal C, ossia il massimo che si possa pretendere da un linguaggio di programmazione, anche se il prezzo è che la competenza richiesta per lavorare con una App è maggiore. I programmatori validi in C, infatti, sono giustamente considerati un sottoinsieme proprio dei programmatori abili in Java o C#.

Come in C e C++, una classe è definita da uno header file, che contiene la sua interfaccia, e da una implementazione, che contiene il corpo dei metodi. L'invio di messaggi a un oggetto segue una sintassi peculiare, che impiega parentesi quadre. In questo esempio, un oggetto chiamato *myArray* riceve un messaggio *insertObject* che ha un parametro,

anObject, e un secondo parametro di nome *atIndex*, che vale 0.

```
[myArray insertObject:anObject
atIndex:0];
```

Il senso è «chiedi all'oggetto *myArray* di inserire l'oggetto *anObject* in posizione 0».

I messaggi possono essere nidificati, per esempio l'oggetto da inserire può arrivare dalla risposta di un altro oggetto, per esempio

```
[myArray
insertObject:[myAppObject
objectToInsert] atIndex:0];
```

Questa nidificazione permette catene di messaggi molto espressive e ha una piacevole reminiscenza Lisp.

Le istruzioni di invio dei messaggi sono miscelate all'interno di codice C.

XCode permette di usare anche il C++ e di mischiare questo linguaggio a quello nativo, anche se lo strato di interfaccia utente deve usare il meccanismo dei messaggi per interfacciarsi con la libreria grafica. Al di là di quanto si possa dire in queste brevi annotazioni, Objective C è facilmente alla portata di un programmatore C, C++ o Java, una volta accettata la sua caratteristica peculiare. Ci sono linguaggi alternativi

che permettono di scrivere per iOS, ma Objective C è il modo ortodosso in cui le cose si fanno, le altre strade sono minoritarie e facilmente deprecate. Da un punto di vista soggettivo, una volta passata la prima barriera, Objective C è un linguaggio simpatico: agile, performante e pragmatico.

Le librerie di supporto

La libreria per lo sviluppo di programmi di iOS è largamente in comune con quella di OS X ed è ricca almeno quanto le omologhe Java o .net. I nuovi arrivati devono mettere in conto la curva di apprendimento. La documentazione online inclusa in XCode è eccellente, ci sono libri elettronici gratuiti su iBooks e un mare di articoli ed esempi di codice su developer.apple.com.

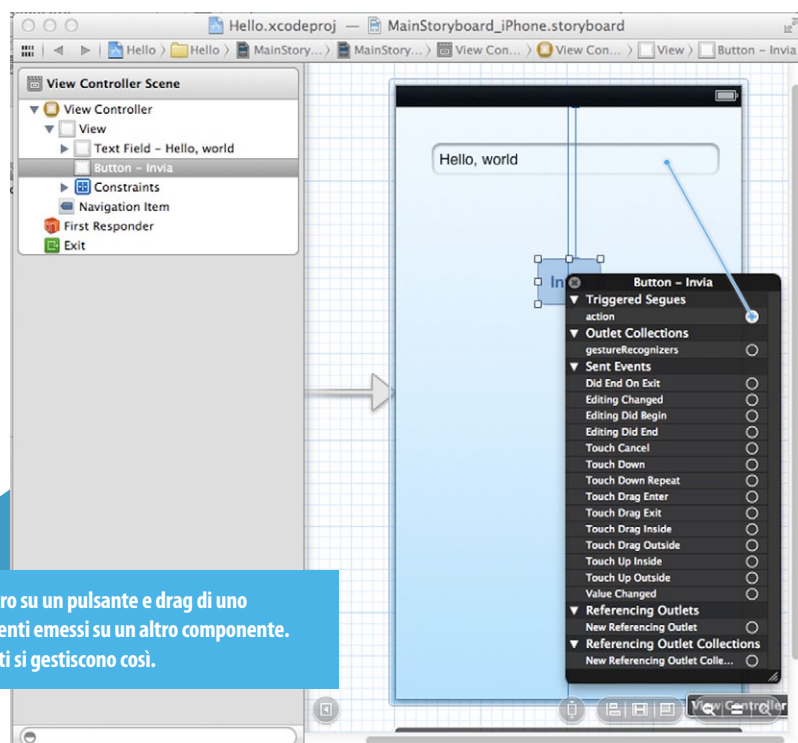
È anche interessante notare che la tag *xcode* ha una volta e mezzo i follower della tag *visual-studio* su stackoverflow.com, un sito per sviluppatori neutro rispetto alla piattaforma. Difficile dire se questo esprime una tendenza del mercato, ma è un dato interessante. Va anche notato che gli abbonati alla tag *android* sono circa sei volte di più.

L'insieme delle classi di supporto è suddiviso in quattro parti o Framework: *Cocoa*, che gestisce l'interfaccia utente, *Media Layer*, che raccoglie tutte le funzioni multimediali, *Core Services Layer*, che gestisce servizi vari, dalla rubrica indirizzi al Gps, infine *Core OS Layer* gestisce i servizi di base del sistema, come la rete, il Bluetooth, la sicurezza e il sistema operativo.

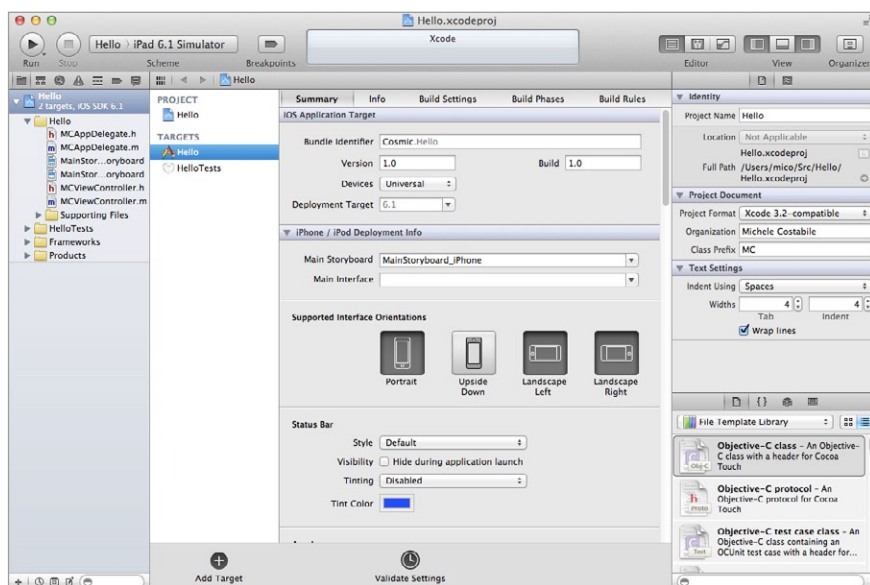
Programmi per sviluppatori

Lo sviluppo di applicazioni per i telefoni cellulari richiede un account per sviluppatori, per provare le applicazioni su un device fisico. Se abbiamo completato lo sviluppo di un'applicazione con l'emulatore ed è ora di provare su un vero device - o meglio, su un campionario di iPhone, iPod e iPad che rispecchi i potenziali clienti - diventa necessario avere un'abilitazione per cominciare a connettere smartphone e tablet al sistema di sviluppo.

Qui c'è il primo pedaggio sulla strada dello sviluppatore: l'iscrizione al Developer Program costa \$99 all'anno, che comprendono il diritto a due richieste di supporto e alla distribuzione dell'applicazione attraverso il negozio Apple. Esiste anche un programma Enterprise,



Clic destro su un pulsante e drag di uno degli eventi emessi su un altro componente. Gli eventi si gestiscono così.



La pagina con le principali proprietà di un'applicazione, sempre all'interno di XCode.

che consente la distribuzione di app su un canale interno, ma non sullo Store, che costa \$299 ed è pensato per applicazioni destinate esclusivamente per uso aziendale. Quasi cento dollari selezionano gli sviluppatori che pensano di avere un ritorno dalla loro applicazione da quelli che lo fanno solo per vedere cosa succede, ma sono una barriera psicologica, che forse non dispiace a Apple, interessata a custodire la fama di un App Store con applicazioni di qualità migliore. Tutto quello che serve sapere per iniziare a creare applicazioni si trova su developer.apple.com, ecco per comodità una URL abbreviata del percorso su cui Apple accoglie le nuove leve: bit.ly/165PbyA.

Iniziare a vendere

Una volta completata un'applicazione, l'account per sviluppatori consente di spedirla al negozio Apple. Il percorso completo è illustrato con molta chiarezza in una pagina del sito per sviluppatori di Cupertino (bit.ly/XvFeVn). La confezione e la consegna dell'applicazione si fanno comodamente dall'interno dell'ambiente di sviluppo XCode.

Il primo scoglio è la creazione del record dedicato all'applicazione nel database dello store.

Per crearlo occorre dare una data prevista di consegna, che può essere comunque modificata; una breve

descrizione che i clienti vedranno sul negozio; un'icona in grande formato (quello canonico è di 512x512 pixel), almeno uno screenshot dell'applicazione e altri dati minori, come il numero di versione. Sul portale si crea un certificato pubblico che, accoppiato con quello privato dello sviluppatore, permette la firma digitale dell'applicazione.

Infine, sempre con XCode, si impacchetta l'applicazione per la consegna e la si spedisce al negozio online impostando la data di disponibilità, che può essere immediata o anche spostata nel futuro. L'applicazione non va in linea subito: lo staff di Apple verifica che siano rispettati i metodi di produzione e le linee guida, sia con strumenti di profilazione automatici, sia direttamente. La vita dell'applicazione continua, dopo la prima pubblicazione, su iTunes Connect, che mette a disposizione dello sviluppatore i dati di marketing, per esempio il numero di download, i giudizi degli utenti e il report dei guadagni.

Allo sviluppatore arrivano anche i log degli errori fatali dell'applicazione, spesso accompagnati da uno stack trace. XCode può utilizzare l'archivio di distribuzione dell'applicazione per risolvere gli stack trace e aiutare a trovare il punto esatto del codice in cui il problema si è determinato. La riga finale, la più importante, è che l'App Store riconosce mensilmente il 70% dei guadagni, senza altri costi, oltre all'abbonamento annuale. In cambio lo sviluppatore riceve visibilità e distribuzione in tutto il mondo. •

PC PROFESSIONALE IN DIFESA DEL CONSUMATORE



✓ Canon MG-2150
Primo prezzo, adatta solo per chi stampa poco

È l'entry level della gamma Piama di Canon, ha un prezzo di listino di 50,83 euro. Adatta per chi vuol spendere il minimo possibile ma stampa poco, poiché utilizza solo due cartucce (una per il nero e l'altra con i tre colori) che, in versione standard, raggiungono le 180 copie. Nella medesima pagina del volantino c'è l'offerta di un set di cartucce per questo modello, conveniente perché include due cartucce del nero per un prezzo di poco superiore alla normale combinazione nero + colore. La MG-2150 può utilizzare anche cartucce ad alta capacità PG-440XL (nero da 600 copie) e CL-541XL (colore da 400 copie): costano 29,85 euro (prezzo Canon) ciascuna. (mm)

✓ Multifunzione HP Photosmart AIO 5510
Mfp completa e versatile

Si tratta di un buon prodotto, con cartucce separate che riducono gli sprechi di inchiostro: display touch screen, Wi-Fi, funzionalità ePrint per la stampa via e-mail da qualsiasi dispositivo connesso a Internet e compatibilità con Apple AirPrint. Ottima funzionalità anche in modalità autonoma e stampa di contenuti Web formattati mediante App disponibili gratuitamente, già installate sull'Mfp o scaricabili dal sito HP. Il prezzo è quello di listino, online si possono trovare facilmente offerte migliori. (mm)



Non perdere su www.pcprofessionale.it i nostri commenti sulla qualità delle offerte proposte sui volantini delle grandi catene di distribuzione

