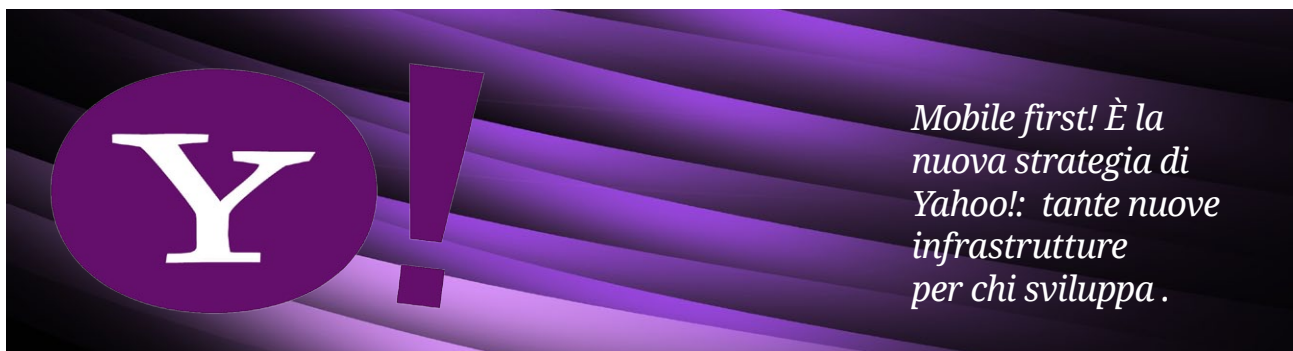


Sviluppo



Di Michele Costabile



Mobile first! È la nuova strategia di Yahoo!: tante nuove infrastrutture per chi sviluppa.

Un turbine di novità da Yahoo!

Nella seconda metà di febbraio, Marissa Mayer, il CEO di Yahoo ha annunciato, durante la prima Mobile Developer Conference organizzata dalla sua società, un cambiamento di strategia importante, sintetizzato in uno slogan: Mobile First. Le ragioni di questa scelta stanno in una slide molto interessante: l'utente medio di smartphone, secondo Mayer, passerebbe 177 minuti al giorno guardando il piccolo schermo. Del tempo dedicato al piccolo schermo, l'88% spetta alle applicazioni.

Non c'è da stupirsi, quindi, se l'investimento in pubblicità sui telefoni è cresciuto più che in qualunque altro settore, inoltre puntare solo sul

browser e sul web significa tralasciare la maggior parte delle occasioni di contatto.

Un altro dato che ci interessa particolarmente è il tasso di crescita della quota di mercato, diviso fra le principali applicazioni, per esempio Facebook e le altre. Secondo i dati pubblicati dal CEO di Flurry, le applicazioni principali, le prime 25, sono cresciute dell'uno per cento, mentre le applicazioni nel resto della classifica, hanno realizzato un incremento del ventuno per cento. Questo dato è estremamente interessante per gli sviluppatori indipendenti e le piccole software house. La mossa dell'azienda di Sunnyvale di fronte alla direzione che ha preso il

mercato è stata l'acquisizione per poco più di 200 milioni di dollari di Flurry, il cui nome significa turbine o tempesta improvvisa: un'azienda specializzata in strumenti per l'analisi di traffico e la distribuzione di annunci pubblicitari. Dopo l'acquisizione, il sito per sviluppatori di Yahoo ha cambiato radicalmente aspetto, conservando solo qualche vestigia degli strumenti e delle tecnologie più legate al web, per lasciare il posto alla suite di soluzioni per applicazioni mobili.

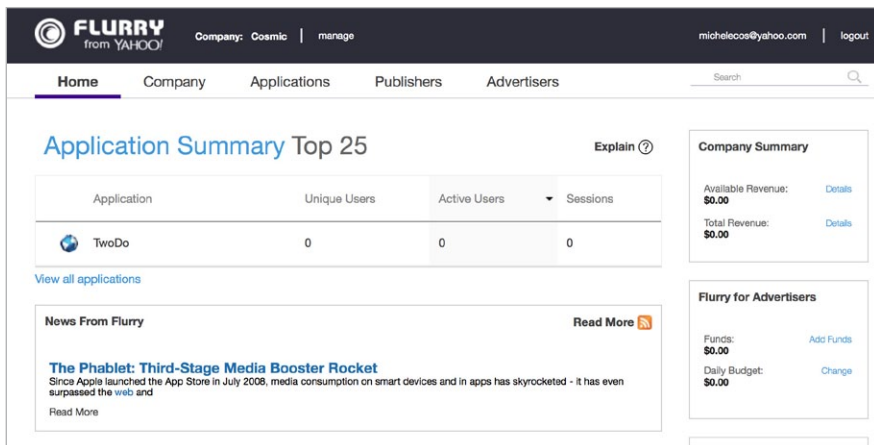
Per lo sviluppatore di applicazioni legate al mondo cellulare, questo significa che esiste un'altra piattaforma da prendere in considerazione per monetizzare le applicazioni e un'infrastruttura gratuita e ben strutturata, per analizzare il comportamento degli utenti.

UNA RAFFICA DI ANNUNCI

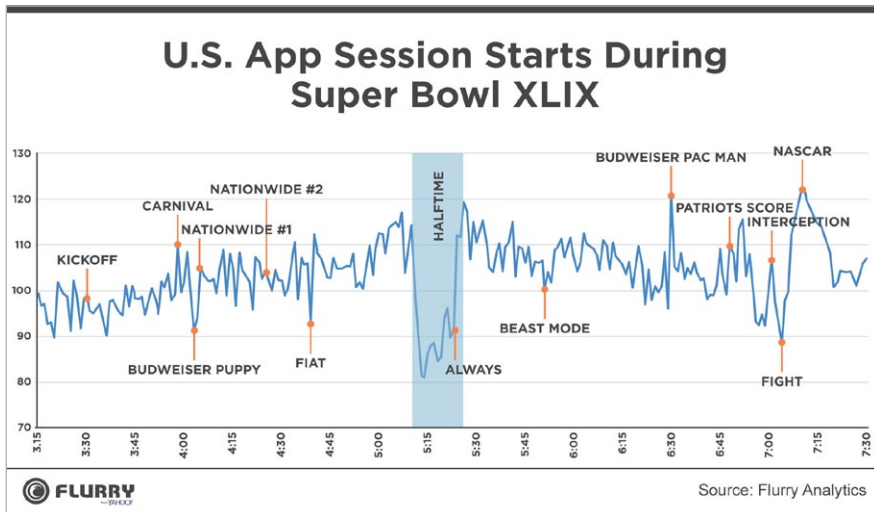
Il primo tassello di novità proposto agli sviluppatori è un'infrastruttura per raccogliere dati sull'uso dell'applicazione e sulle abitudini degli utenti accompagnato da un cruscotto per la visualizzazione e l'analisi che permette di aggregare e analizzare i dati generati dalle applicazioni in diversi modi in un'interfaccia web chiamata Analytics Explorer. Il servizio è gratuito e rimane tale a qualunque scala di impiego, secondo la documentazione su developer.yahoo.com.

Gli eventi sono generati dall'interno delle applicazioni, con codice simile





Il cruscotto di Flurry Analytics permette di fare interrogazioni su profilazione e demografia degli utenti e ha una colonna a destra che mostra il guadagno attraverso la pubblicità.



Marketing e sociologia si incontrano. Il grafico orario del lancio di applicazioni su cellulare nel Super Bowl, mostra un "fosso" durante l'esibizione di Katy Perry, un segno di attenzione.

al seguente:

```
FlurryAgent.  
logEvent("Article_Read");
```

e ogni applicazione può generare segnalazioni per un massimo di trecento eventi diversi. Dato che gli eventi sono definiti dal programma, si possono utilizzare per tracciare il livello a cui arrivano i giocatori in un gioco, il numero di volte in cui viene usata la pagina delle preferenze di un'applicazione o qual è la vista di default preferita. Il toolkit permette di tracciare la posizione geografica, il genere e l'età degli utenti. Lo strato di segnalazione degli eventi gestisce le condizioni di assenza di connettività. Anche un identificativo univoco degli utenti viene gestito dalla

strumentazione.

Fra gli esempi di domande a cui il toolkit fornisce una risposta, troviamo: quanti utenti che hanno letto il tutorial hanno finito i primi tre livelli in tre giorni? Oppure quanti utenti comprano un'espansione tre o quattro giorni dopo il primo lancio dell'applicazione? Questo genere di analisi di eventi scatenati da altri eventi viene chiamato *funnel*, cioè imbuto. Un video che mostra le funzionalità avanzate del sistema di analisi si trova all'indirizzo youtube.com/watch?v=al-T0szCbNM. La strumentazione per l'analisi dei dati può essere integrata in applicazioni iOS, Android e Html.

Un'altra funzione interessante, secondo la documentazione, consente di valutare dati come la frequenza di uso della

propria applicazione, in confronto con altre app della stessa categoria nello store di riferimento.

Yahoo App Publishing è una piattaforma di distribuzione di banner e video pubblicitari, molto facile da integrare nelle applicazioni Android o iOS. Lo sviluppatore può scegliere se integrare banner, pubblicità interstiziale all'interno di liste pubblicate dall'applicazione, o pubblicità video. Si può anche impostare Flurry come piattaforma mediata da altri sistemi di pubblicazione, come AdMob, McPub o DFP.

Yahoo App Marketing è un'infrastruttura che ha lo scopo di inviare la pubblicità di applicazioni a un insieme di potenziali acquirenti, scegliendo il bersaglio in base a un insieme di criteri, come età, sesso, locazione geografica, passione per le app.

Flurry Pulse consente di inviare dati sull'uso dell'applicazione a partner che si occupano di analizzare l'uso di applicazioni e stilare classifiche, come comScore. Anche questo servizio è gratuito.

Yahoo Search in Apps, infine, facilita l'integrazione di funzioni di ricerca all'interno delle applicazioni, naturalmente con il motore di Yahoo. La ricerca di Yahoo è gratuita a qualunque scala di impiego.

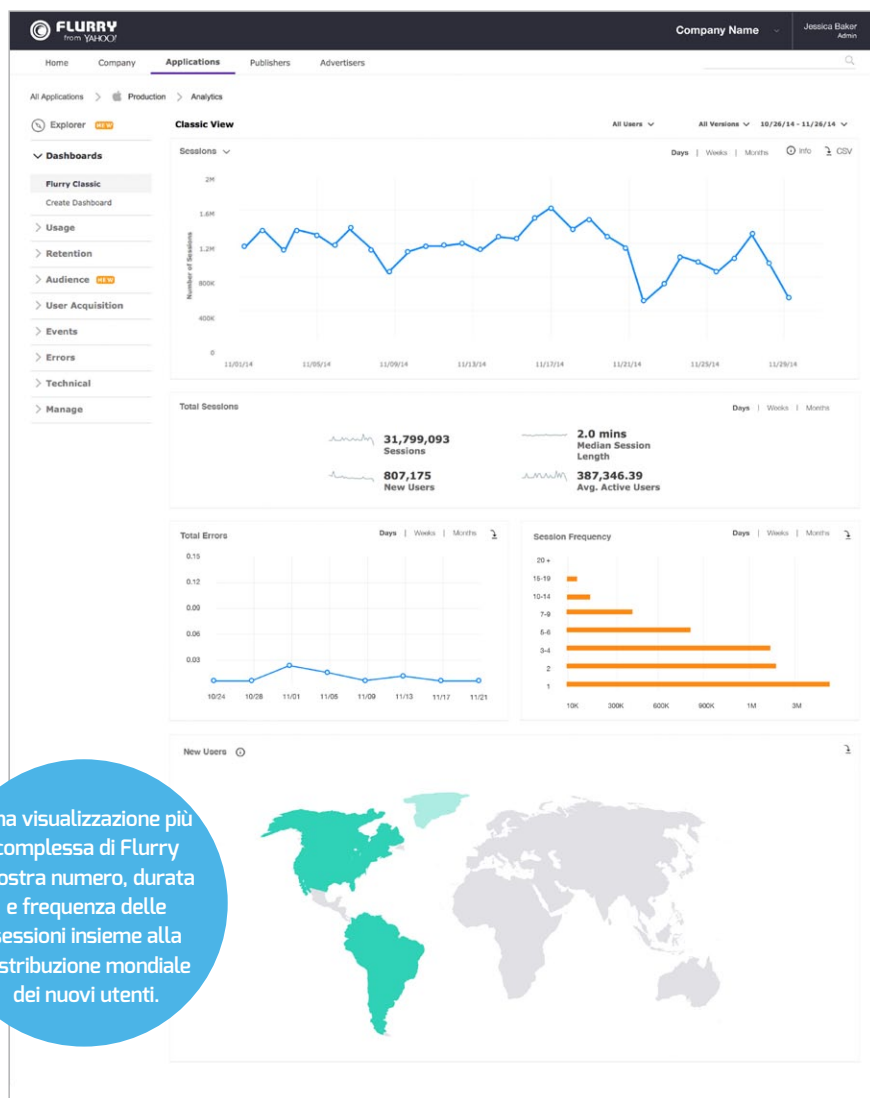
COME FUNZIONA L'INTEGRAZIONE

L'integrazione del toolkit in un'applicazione iOS richiede pochissime righe di codice:

```
#import "Flurry.h"  
  
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    //init Flurry  
    [Flurry startSession:@"YOUR_FLURRY_API_KEY"];  
    // Your code ...  
}
```

Lo stesso vale per Android

```
import com.flurry.android.  
FlurryAgent;  
//If you are writing an app with  
the minimum target set to API level  
14 or greater,  
//session handling is completely  
automatic
```



Una visualizzazione più complessa di Flurry mostra numero, durata e frequenza delle sessioni insieme alla distribuzione mondiale dei nuovi utenti.

```
//If you are shipping an app,
extend the Application class if you
are not already doing so:
public class MyApplication extends
Application {
    @Override
    public void onCreate() {
        super.onCreate();
        // init Flurry
        FlurryAgent.init(this,
YOUR_FLURRY_API_KEY);
        //....
    }
}
```

e per un'applicazione html

```
<script src="https://cdn.
flurry.com/js/flurry.js"></
script> <script>FlurryAgent.
startSession("YOUR_API_KEY");</
script>
```

In tutti questi casi, occorre sostituire

alla stringa campione, la propria chiave per l'uso delle interfacce applicative di Flurry.

Dopo avere stabilito la connessione, possiamo associare il tracciamento a un utente specifico, nel caso di un'applicazione html

```
FlurryAgent.setUserId(userId);
```

Così come possiamo associare una posizione geografica a questo episodio di uso dell'applicazione

```
FlurryAgent.setLocation(latitude,
longitude, accuracy);
```

E, infine, tracciare eventi

```
FlurryAgent.logEvent(eventName);
```

```
FlurryAgent.logEvent(eventName,
eventParameters);
```

INCORPORARE BANNER PUBBLICITARI

Ecco un esempio di codice che mostra come inserire banner pubblicitari in una UIViewController

```
#import "FlurryAdInterstitial.h"
#import "FlurryAdInterstitialDeleg
ate.h"

/**
 * Si possono integrare annunci in di-
versi modi, in questo caso presentiamo
* l'integrazione in una UIViewController
*/
@property FlurryAdInterstitial*
adInterstitial;
```

```
// the adSpaceName refers to the ad
space configured on dev.flurry.com
under Publishers tab then Inventory
/ Ad Spaces on the left nav
NSString *adSpaceName =
@"INTERSTITIAL_MAIN_VIEW";
```

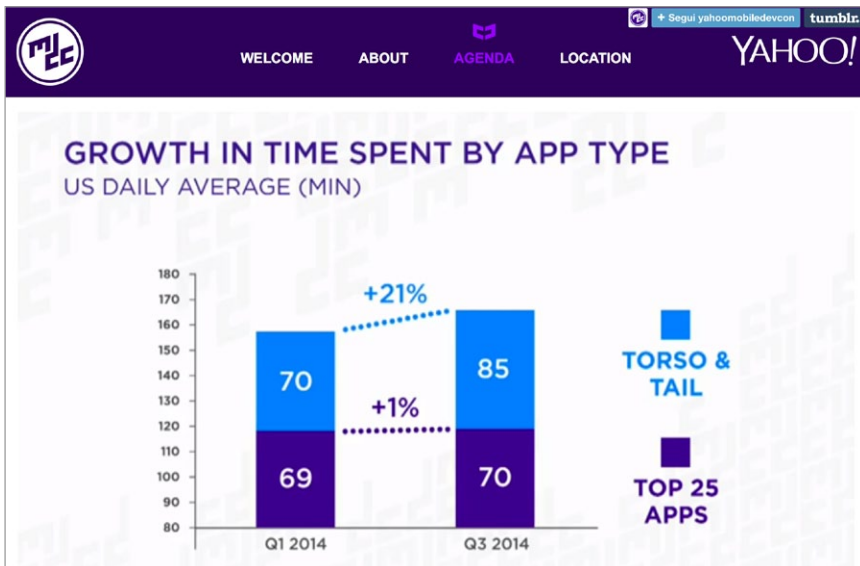
```
- (void)viewDidAppear:(BOOL)
animated {
    [super viewDidAppear:animated];

    // Fetch interstitial ads early
when a later display is likely. For
// example, at the beginning of
a level.
    self.adInterstitial =
[[FlurryAdInterstitial alloc]
initWithSpace:adSpaceName] ;
    self.adInterstitial.adDelegate
= self;
    [self.adInterstitial fetchAd];
}
```

```
/**
 * Un buon momento per integrare un
annuncio interstiziale è quando
* l'applicazione ha raggiunto una pausa
naturale, per esempio quando
* abbiamo completato un livello in un
gioco.
```

```
* In questo caso, l'evento a cui ci colle-
ghiamo è il clic di un pulsante.
```

```
*/
-(IBAction) showFullScreenAdClicked
Button:(id)sender {
    // Check if ad is ready. If so,
display the ad
    if ([self.adInterstitial ready]
== YES) {
        [self.adInterstitial
presentWithViewController:self];
    } else {
        // if the ad is not ready,
```



I grafici sono in minuti e mostrano il tempo dedicato allo smartphone in confronto al tempo dedicato alla TV (che spesso coincidono). I numeri sono impressionanti: quasi tre ore quotidiane di uso del telefono, di cui 82% va alle applicazioni e non al browser.

```
fetch the ad
self.adInterstitial =
[[FlurryAdInterstitial alloc]
initWithSpace:adSpace];
self.adInterstitial.
adDelegate = self;
[self.adInterstitial
fetchAd];
}
```

Ecco lo stesso esempio nel caso di Android

```
import com.flurry.android.
FlurryAgent;
import com.flurry.android.ads.
FlurryAdInterstitial;

public class
InterstitialAdsActivity extends
Activity {
// the adSpaceName refers to
the ad space configured on dev.
flurry.com under Publishers tab
// under left-hand nav
Inventory / Ad Spaces
private String adSpaceName =
"Takeover";
private FlurryAdInterstitial
mFlurryAdInterstitial = null;
private Button displayAd;
```

```
@Override
public void onStart() {
super.onStart();
//for ICE Cream Sandwich
and higher you can skip the
```

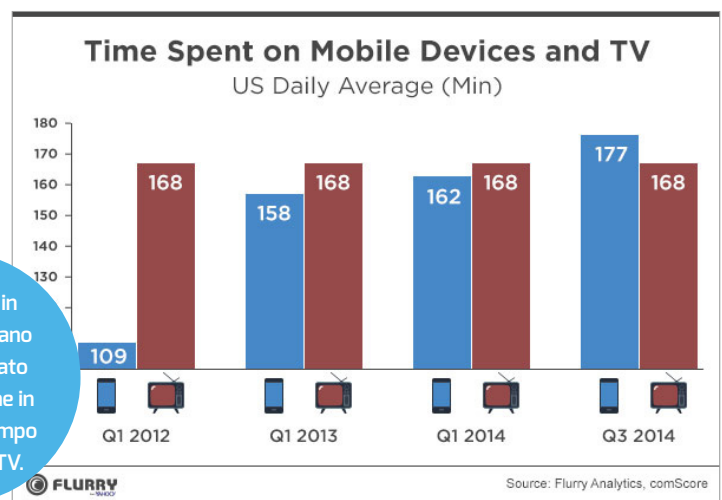
```
FlurryAgent.onStartSession;
FlurryAgent.
onStartSession(this, mApiKey);

displayAd = (Button)
findViewById(R.id.display);
mFlurryAdInterstitial =
new FlurryAdInterstitial(this,
mAdSpaceName);
mFlurryAdInterstitial.
fetchAd();
/**
* Invoke a takeover at
a natural pause in your app. For
example, when a
* level is completed, an
article is read or a button is
pressed. Here we
* mock the display of a
takeover when a button is pressed.
*/
```

```
displayAd.
setOnClickListener(new
OnClickListener() {
@Override
public void
onClick(View v) {
// fetch and
prepare ad for this ad space. won't
render one yet
if
(mFlurryAdInterstitial.isReady()) {
mFlurryAdInterstitial.displayAd();
} else {
mFlurryAdInterstitial = new
FlurryAdInterstitial(this,
mAdSpaceName);
mFlurryAdInterstitial.fetchAd();
}
});
@Override
public void onStop() {
super.onStop();
// con ICE Cream Sandwich e
successivi, si può saltare la
// FlurryAgent.
onEndSession;
FlurryAgent.
onEndSession(this);
}
```

USARE LA RICERCA

L'integrazione della ricerca di Yahoo in un'applicazione iOS è presentata su Github, all'indirizzo github.com/yahoo/searchsdk-ios



I grafici sono in minuti e mostrano il tempo dedicato allo smartphone in confronto al tempo dedicato alla TV.


```
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder,
UIApplicationDelegate {

    var window: UIWindow?

    func application(application:
UIApplication,
didFinishLaunchingWithOptions
launchOptions: [NSObject:
AnyObject]?) -> Bool {
        self.setupYahooSearchSDK()
        self.
setupStatusBarAppearance()
        return true
    }

    private func
setupYahooSearchSDK() {
        YSLSetting.setupWithAppId("
YourApplicationId");
    }

    func setupStatusBarAppearance()
{
        UIApplication.
sharedApplication().statusBarStyle
= .LightContent
    }
}
```

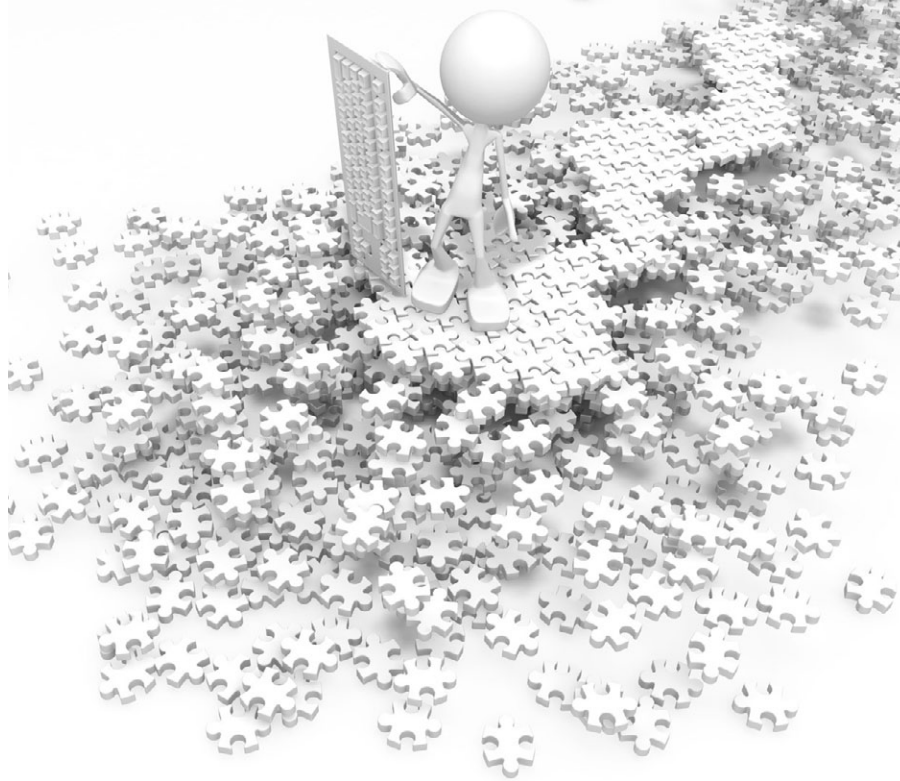
CONCLUSIONI

L'entrata in forze di Yahoo nel mondo della pubblicità all'interno delle applicazioni mobili crea nuove possibilità di scelta per gli sviluppatori.

I motivi che hanno spinto Yahoo a questo passo, sono spunto di riflessione per tutti coloro che vivono grazie alla creazione di applicazioni.

Yahoo un'azienda grande, costretta a essere dinamica dal calo di utili che ha subito in passato e dalla nuova direzione di Marissa Mayer, di provenienza da Google, vede un mercato interessante nelle applicazioni mobili e nella pubblicità su telefoni cellulari e, probabilmente, dovremmo vederci una fonte di potenziale guadagno anche noi.

Per inciso, segnaliamo che la sezione Blog e Insights del sito *web.flurry.com* è una miniera di dati interessanti per chiunque stia valutando il mercato delle applicazioni mobili. Alla fine dell'analisi, la risposta sintetica è buttiamoci. •

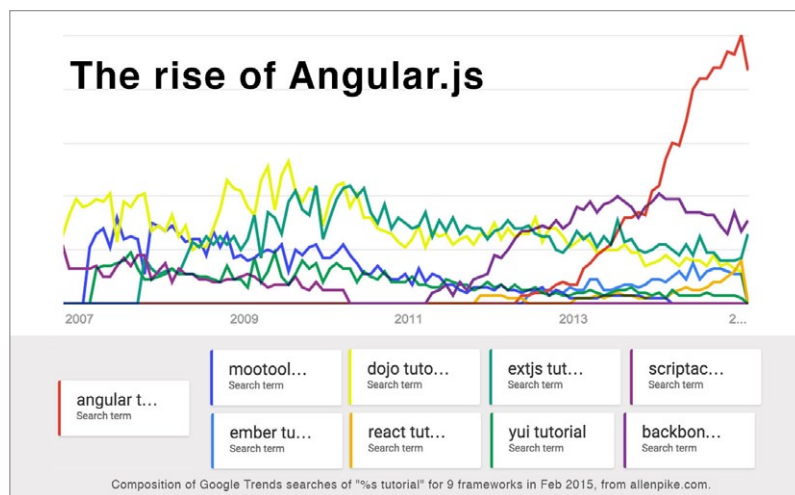


Aiuto, quanti framework

Un post di Allen Pike, sviluppatore Apple canadese e blogger, ci invita a riflettere su JavaScript e la sua dinamicità. L'articolo, pubblicato all'indirizzo allenpike.com/2015/javascript-framework-fatigue, parla della velocità di proliferazione dei framework JavaScript sul web. Si tratta di un problema che avevamo già incontrato quando, nel tempo in cui abbiamo letto il manuale di Prototype e Scriptaculous, si era già affermato come standard jQuery. Oggi i tempi sono più veloci e probabilmente la stella di un framework rischia di tramontare nel tempo in cui Amazon consegna il

Mai come ora gli sviluppatori hanno a disposizione librerie di funzioni pronte all'uso.

manuale. Qualcuno ha fatto una interessante collezione di framework sul sito *web.todomvc.com*, che ospita una sessantina di esemplari della stessa applicazione: una lista di cose da fare, realizzata con uno stile MVC o MV* per sottintendere tutte le possibili variazioni.



Il grafico pubblicato su allenpike.com, che mostra l'andamento temporale dell'interesse per diversi framework JavaScript di cui abbiamo parlato.

Il picco per Angular sembra indicare che ci sia un certo consenso fra gli sviluppatori

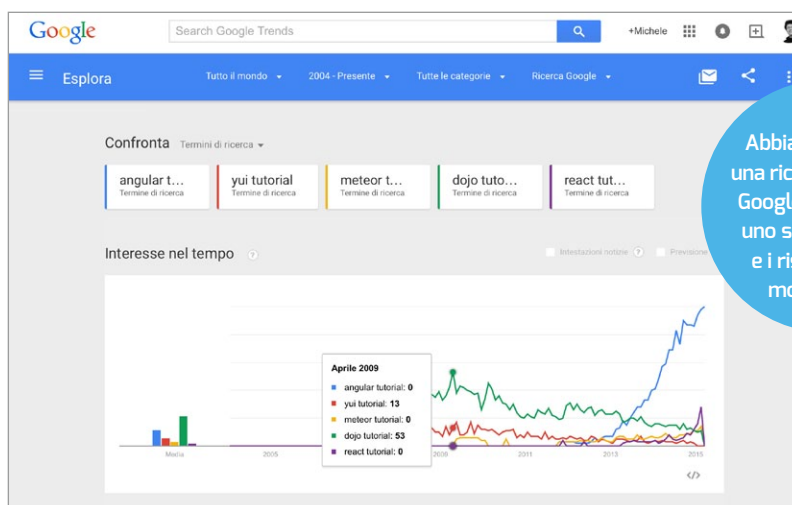
Di ogni applicazione sono disponibili i sorgenti, tutti open source e archiviati su Github, in modo che uno sviluppatore possa rendersi conto di come sono i dettagli di ogni applicazione. Esiste anche una versione di riferimento, scritta per riferimento in puro e semplice JavaScript, senza nessun framework, che totalizza un numero di righe di codice piuttosto elevato.

La versione di controllo scritta con jQuery senza nessun framework, invece, si ferma a 212 righe di codice, che non è poi molto, ma non è di facile lettura. Un framework è necessario. Può capitare di iniziare un progetto convinti di potersela cavare con un po' di codice e la manipolazione del dom standard, ma, non appena la complessità sale di poco, si rischia di trovarsi con sorgenti in crescita continua e sempre più difficili da tenere insieme.

JavaScript, per di più non è mai stato rinomato per una standardizzazione eccellente, un disegno pulito e prestazioni elevate, quindi meglio non spingerlo ai limiti.

Per contro, come abbiamo rilevato parecchie volte su queste pagine, JavaScript è un linguaggio simpatico e che può essere abusato in molti modi utili, fino a trattarlo come un lisp con le parentesi graffe (questa affermazione va intesa come un complimento). Proprio per questa dinamica adattabilità e per la innegabile produttività, i framework continuano a proliferare.

Le ragioni sono da cercare nei limiti del browser: troppo codice e l'applicazione rischia di annasparsi, specie sulle macchine più vecchie, poco codice ottimizzato e abbiamo un framework scattante,



Abbiamo provato una ricerca simile su Google Trends, con uno scope italiano e i risultati sono molto simili.

ma limitato nelle possibilità. Anche la modularizzazione del codice è un'arma a doppio taglio. Questo spiega perché spesso c'è chi si imbarca nella creazione di qualcosa di nuovo e, magari, riesce ad avere successo e popolarità, almeno fino al prossimo nato.

Questa situazione è del tutto diversa da quella che sperimentano gli sviluppatori di app per i quali la piattaforma spesso detta sia il linguaggio, sia il framework di riferimento. Sappiamo lo scalpore che si genera quando un produttore forza un cambio di pelle e liquido amniotico, per esempio quando Microsoft ha pensionato COM e Visual Basic per passare a .net e C#, così come quando Apple ha introdotto Swift.

Una soluzione originale e pragmatica al problema della scelta, Allen l'ha trovata in Google Trends, una funzione di ricerca di Google che permette di analizzare la frequenza di determinate stringhe di ricerca nel corso del tempo

e di visualizzare grafici che confrontano l'andamento temporale di diverse stringhe. Un modo di verificare l'adozione di nuovi framework è valutare con quale frequenza si ricercano articoli introduttivi a un determinato corpo di codice. Il grafico in figura mostra in colori diversi l'andamento temporale della ricerca di tutorial per un certo numero di framework e, con sorpresa, mostra un chiaro vincitore: Angular.js. Abbiamo ripetuto il test, con un numero minore di variabili e con un ambito regionale italiano, ottenendo un'indicazione simile, insieme a una conferma della crescita dell'interesse per Meteor e Cordova, due framework che affiancano lo sviluppo mobile al web.

Per confronto, abbiamo provato a cercare tracce di interesse per linguaggi nativi, come Objective-C e Swift, ma i grafici viaggiano su due scale diverse non confrontabili.



SEGUICI SU

