

Lo smartwatch della società di Cupertino delega all'iPhone una buona parte dell'elaborazione dei dati. Agli sviluppatori sono state date le prime indicazioni.

Di Evaristo Meneguzzi



Telefono e orologio in simbiosi. L'accoppiata di Apple sarà vincente?

Apple Watch è il primo prodotto concepito da Apple dopo la scomparsa del carismatico fondatore, Steve Jobs. Rappresenta una sfida sia sotto il profilo tecnologico sia sotto quello strategico: sarà la cartina tornasole con cui il mercato e i commentatori valuteranno la capacità di innovazione della società di Cupertino. È un prodotto decisamente fuori dagli schemi, non solo nel design, ancillare a un dispositivo iOS, l'iPhone, e suo complemento.

Al momento non c'è una data ufficiale di presentazione se non il generico «inizio 2015» citato da Tim Cook durante la presentazione dell'anno scorso. Nell'attesa però Apple ha rilasciato sia il kit di sviluppo, sia la documentazione di base, per gli sviluppatori in modo da poter proporre una collezione di applicazioni indipendenti nello stesso giorno in cui l'orologio farà il suo debutto sul mercato. Da questa documentazione, provvisoria

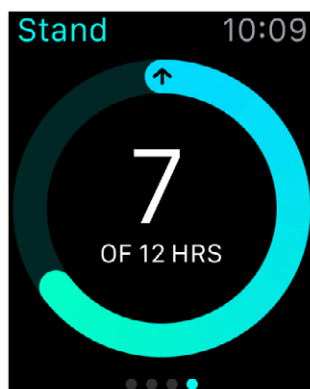
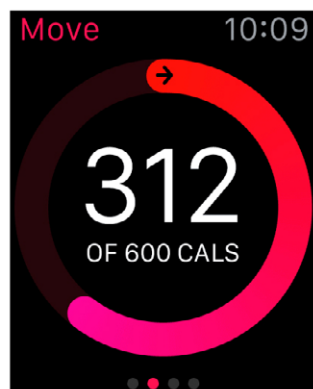
e incompleta se non nelle sue linee guida, si possono comunque valutare l'architettura dell'interfaccia utente del nuovo dispositivo, la sua interazione con l'utilizzatore e soprattutto (e ben più importante) come Apple indirizza gli sviluppatori, con un tocco gentile ed educato, nella direzione in cui intende far evolvere il suo Apple Watch.

Il WatchKit è il framework con cui gli sviluppatori dovranno interagire per la realizzazione di applicazioni che, come Apple sottolinea, saranno complementari ad iOS, senza l'intenzione di sostituirsi ad esse. «Le interazioni con le App iOS si misurano in minuti, quelle con WatchKit in secondi» afferma la guida all'interfaccia utente. La regola che si deduce quindi è chiara: utilizzo immediato, veloce, interfaccia semplicissima e con meno cambiamenti di stato possibili. I metodi di navigazione previsti all'interno di un app sono per ora solo due, quello

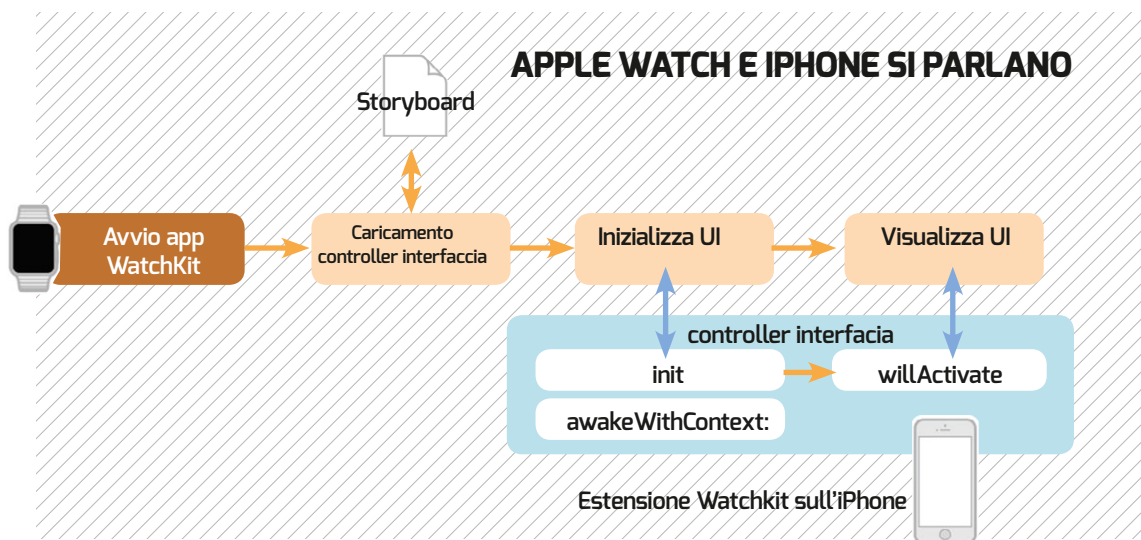
gerarchico e quello *page-based*. Nel primo l'utente seleziona via via le informazioni in successione, per esempio quando in un elenco si passa da una regione a una città e infine a una via.

Il modello *page-based* ha invece un'interazione basata sullo scorrimento orizzontale in seguito allo *swipe* con le dita sul quadrante dell'orologio. Al programmatore spetta creare gli elementi dell'interfaccia che consentono il ritorno sui propri passi nelle videate precedenti, in modo graficamente coerente con lo stile della propria app.

L'interazione dell'utente in entrambe le modalità avviene con gesture predeterminate nel framework, lo *swipe* appena citato (orizzontale e verticale), il *tap* singolo e doppio sugli elementi dell'interfaccia (bottoni, testi, righe, celle di tabella), il *force touch* (lo schermo è sensibile alla pressione esercitata). Questa azione, toccare e tenere premuto, consente allo sviluppatore di far apparire menù secondari o attivare azioni legate al contesto, una specie di "tasto destro" del mouse. Un'altra interazione, assolutamente esclusiva all'Apple Watch, avviene



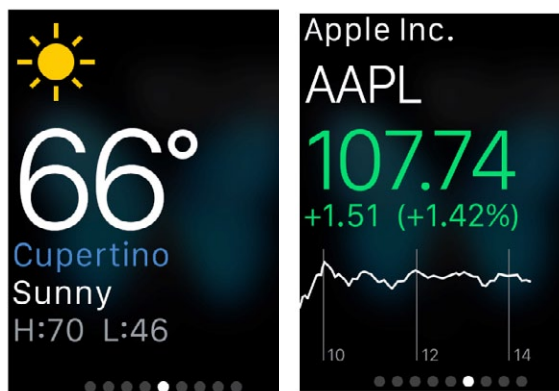
Un esempio di applicazione con scorrimento *page-based*.



attraverso quello che negli orologi tradizionali è il pulsante di ricarica che nel gergo degli sviluppatori Apple è stato battezzato *Digital Crown*. Secondo le specifiche è un controllo molto preciso, particolarmente fine nello spostamento e in grado di riconoscerle la velocità di accelerazione del suo movimento. Al momento le applicazioni di terze parti potranno utilizzare la corona solo per le operazioni di scrolling, mentre all'interno del sistema operativo potrebbe essere sfruttata per altre interazioni di cui non sono rese note le modalità.

Proprio per favorire la creazione di applicazioni dall'utilizzo veloce e istantaneo viene messa a disposizione degli sviluppatori una modalità chiamata *Glances*, in italiano si tradurrebbe come «sguardo». Il loro scopo è fornire all'utente in una singola videata informazioni precise e puntuali fornite dalle app preferite. Uno sguardo immediato su informazioni considerate importanti e critiche. Si può scorrere da una Glance all'altra, ma la singola app non consente scroll laterali e può solo fornire informazioni estrapolate da un altro programma. Le Glances sono basate su template ben definiti, disponibili in XCode, che differenziano la parte superiore e quella inferiore dello schermo per eventuali interazioni. Ciascuna Glance può essere associata a una sola azione: se l'utente

Due Glance:
piccole applicazioni
che evidenziano
un solo tipo di dato.



tocca lo schermo si passa all'applicazione "genitore" che fornisce i dati. La comunicazione tra app e Glance corrispondente avviene attraverso i protocolli di *Handoff* che hanno debuttato con Yosemite e iOS 8. In pratica le Glances sono compatti e funzionali widget monopagina che permettono di tenere sotto controllo i dati e visualizzarli con un semplice movimento del polso.

ANATOMIA DELL'APPLICAZIONE

Le app costruite per l'orologio di Apple sono composte da due parti, l'app vera e propria e un'estensione. L'app risiede sull'orologio (contiene solo le risorse grafiche ma non il codice eseguibile); l'estensione si installa sull'iPhone e ha al suo interno il codice e i dati per comunicare con l'interfaccia dell'app residente sull'Apple Watch.

Le due componenti lavorano in tandem: quando si interagisce sull'orologio viene mostrata dall'app la rappresentazione grafica del contesto completata con i dati richiesti alla controparte presente sull'iPhone, in modo trasparente all'utente. L'esempio citato nella documentazione di Apple è semplice: quando

si attiva una Glance, il codice dell'app chiede all'estensione sull'iPhone le informazioni da visualizzare come mostrato nello schema qui sopra.

Ogni videata (o, meglio, *scena*) ha un suo controller chiamato *WKInterfaceController* che svolge un compito simile ai view controller di iOS che gestiscono e presentano il contenuto sullo schermo in risposta alle azioni dell'utente. A differenza dei view controller, però, il *WKInterfaceController* non accede in modo diretto alle singole view ma delega questo compito al WatchKit.

Quale è, in breve, il ciclo di vita di un app sull'Apple Watch? L'utente tocca lo schermo e avvia dall'home screen l'app che desidera o attiva una Glance. Quest'azione lancia l'app sull'orologio e l'estensione sull'iPhone; i due colloquiano finché non sono individuati i dati necessari, inviati dal telefono all'orologio e poi visualizzati. L'estensione a questo punto viene messa in pausa fino alla richiesta successiva per ridurre i consumi energetici sul telefono.

Le applicazioni sull'orologio che per progettazione e scopo devono interagire in modo diretto con l'estensione (senza passare dal WatchKit) possono utilizzare API specifiche, in particolare il metodo *openParentApplication:reply* per chiedere e ottenere i dati necessari. Il WatchKit non supporta metodi di esecuzione in background ed è attivo solo finché l'utente interagisce con l'app corrispondente sull'orologio; l'estensione sull'iPhone può invece essere programmata per operare in background, elaborare o raccogliere dalla rete eventuali dati, rilevare la posizione dell'utente, da passare poi all'app sul telefono.