

Sviluppo



Di Michele Costabile



Aggiornamento per l'ambiente di sviluppo: si tratta della modifica più importante all'ultima edizione della soluzione Microsoft.

Visual Studio 2013: update 2

L'ultimo aggiornamento introduce diverse novità, ma la più importante, quella che dà il colore all'evento, è un passo importante verso la convergenza delle piattaforme di sviluppo Microsoft per telefono, tablet e personal computer.

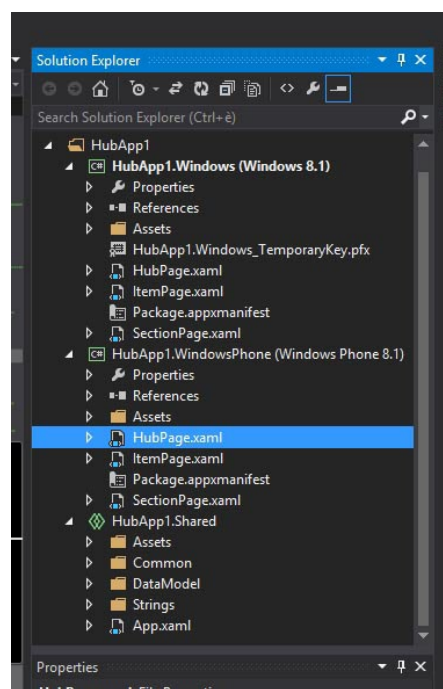
Redmond, come si sa, ha scommesso - forse azzardatamente - su una convergenza massiccia del software per telefono, tablet e computer.

Abbiamo visto la creazione di un'interfaccia utente originale, con diversi particolari che ci piacciono, con la release 7.5 di Windows Phone, nome in codice *Mango*. Impostazione visuale e interattiva del sistema operativo, che Windows 8 ha portato sul Pc.

Microsoft, successivamente, ha impostato la sua linea di sistemi operativi, per telefono, tablet e computer, scegliendo il punto di divisione all'altezza del tablet. Ovvero: le versioni di sistema operativo sono due, come per Apple, e quella in comune è fra tablet e computer. Apple, invece, ha optato per tenere telefono e tablet in un cesto, con iOS, e il computer in un altro, con OS X.

Non c'è dubbio sul fatto che sia stata una mossa coraggiosa e azzardata, una scelta con cui Microsoft si è anche fatta del male. In primo luogo, perché il desktop tradizionale non è potuto scomparire, per non perdere la compatibilità con le applicazioni del passato, creando confusione e la strana impressioni di lavorare con due sistemi operativi pinzati insieme in qualche modo. In secondo luogo, perché uno schema di interazione stile tablet, con una singola applicazione a tutto schermo fa infuriare gli utenti avanzati, quelli che hanno diverse finestre aperte e sanno come usarle, il che spiega come mai Apple non si schiodi ancora dal modello di interazione per cui va giustamente famosa, il più adatto agli ampi monitor dei suoi computer da tavolo.

Per gli sviluppatori c'è l'indubbio vantaggio di poter considerare



Una **Universal App** contiene tre progetti, uno che ha per target Windows 8.1, uno per Windows Phone 8.1 e uno con le parti comuni.

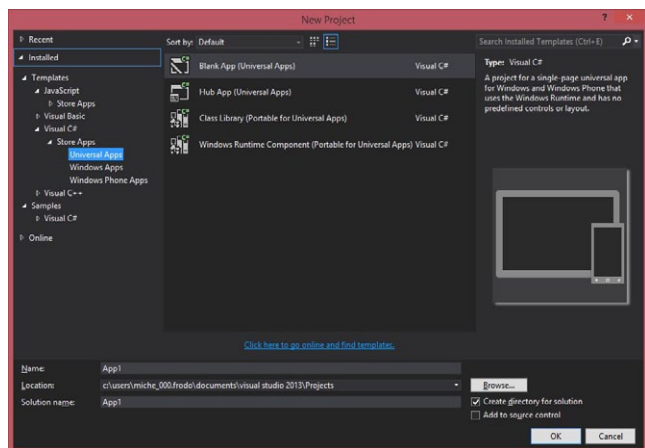


Gli sviluppatori potranno unificare le proprie applicazioni per tablet e per desktop sotto il nuovo modello di interfaccia.

l'unificazione di applicazioni per il tablet e il Pc, un vantaggio sminuito solo dal fatto che non sembra ancora che il nuovo modello di interfaccia si sia veramente fatto strada nelle richieste degli utenti e nelle proposte degli sviluppatori. La tendenza, comunque, non si ferma: Windows 9 dovrebbe spingere l'unificazione della piattaforma ancora più in là.

In attesa del prossimo sistema operativo, la nuova versione di Visual Studio presenta le app universali, pronte per girare su ogni piattaforma.

Fra i modelli di applicazione che si possono creare con Visual Studio 2013 update 2, c'è anche la categoria Universal App.



UNIVERSAL APP

Abbiamo sperimentato il nuovo modello di app universale con Visual Studio Express 2013 Update 2 RC, la versione gratuita dell'ambiente di sviluppo, che si può scaricare da www.microsoft.com/en-us/download/details.aspx?id=42305. Ricordiamo che conviene aggiornare la macchina di test a Windows 8.1 prima di procedere.

Fra le applicazioni proposte dalla finestra di dialogo dei nuovi progetti è disponibile una collezione di modelli di applicazione universale. Abbiamo scelto un'applicazione di tipo Hub, che ci pare quella più flessibile.

La soluzione creata dal wizard, contiene un progetto per applicazioni Windows 8.1, uno per applicazioni Windows Phone 8.1 e un progetto condiviso. Manca un progetto per applicazioni destinate al desktop classico di Windows, mentre è stata promessa la compatibilità anche con Xbox One al *Microsoft Build*, la conferenza per sviluppatori che si

è tenuta agli inizi di aprile. Il progetto condiviso contiene tutta la logica di navigazione dell'applicazione, espressa dalla classe RelayCommand, ben familiare a chi ha sviluppato con Wpf e Xaml, e tutte le classi derivate contenute in una classe NavigationHelper. Alcune sezioni del codice sono specifiche per un sistema operativo, abbiamo trovato diverse sezioni protette da un `#if WINDOWS_PHONE_APP`.

L'approccio con codice condizionale ha senso perché si mantengono semplici le interfacce, non si carica la struttura delle classi di eccessivi pattern e - probabilmente - si anticipa un futuro in cui una maggiore integrazione fra le piattaforme richiederà meno specializzazione.

Nell'editor c'è un elenco a discesa che permette di stabilire il contesto in cui si sta editando un file, mettendo a fuoco una piattaforma specifica. Questo si riflette nel codice mettendo in grigio le aree di codice escluse dalla compilazione condizionale. Altri elementi a

fattor comune sono un dizionario osservabile, ObservableDictionary, che consente l'invio di eventi all'interfaccia utente, un SuspensionManager, che gestisce il ciclo di vita dell'applicazione, un DataModel, che incapsula l'accesso ai dati e un elenco di stringhe pronto per l'internazionalizzazione.

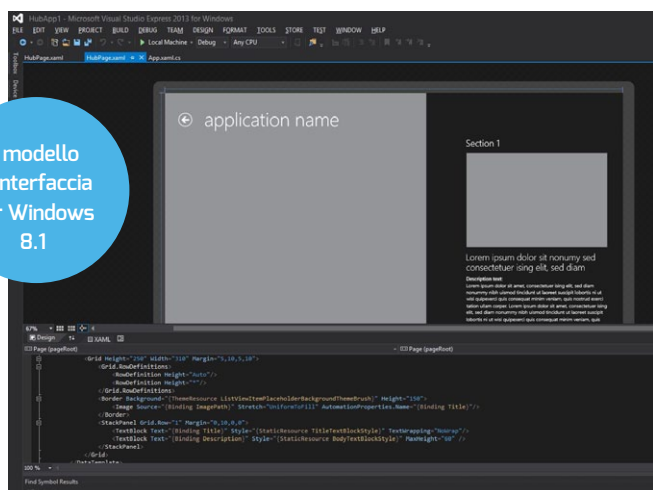
Nei progetti specifici per piattaforma, invece, troviamo la pagina principale, le pagine di sezione e quelle di dettaglio per la lista di oggetti per cui sono pensate le applicazioni hub. Naturalmente, le pagine di interfaccia sono ottimizzate per fattori di forma diversi e hanno layout per dimensioni radicalmente diverse dello schermo.

Il modello ci è parso solido e ben ponderato e crediamo sia efficace nella sua missione di essere di aiuto e ispirazione per gli sviluppatori. Ovviamente, conviene continuare a tenere la logica applicativa nella parte condivisa del codice espandendo lo scheletro di comandi e eventi e legandolo alla logica di business.

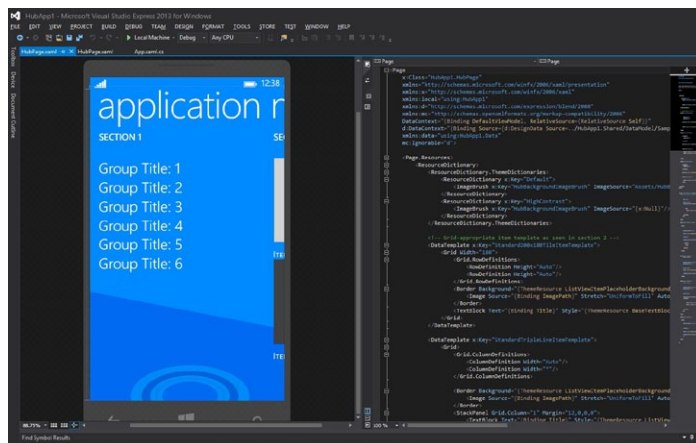
E ORA C# DIVENTA NATIVO

Un'altra novità dell'ultima release di Visual Studio è il supporto, in beta, per una funzionalità chiamata .net native, che migliorerà del 60% il tempo di avvio delle applicazioni Windows Store. È uno stadio di compilazione che trasforma in codice nativo il codice intermedio di un'applicazione scritta in C#. L'ambiente di compilazione e link produce una versione nativa delle funzioni del framework. Il risultato è un eseguibile binario che non ha dipendenze da .net, pur godendo dell'affidabilità propria del codice managed. Quello che è interessante è il momento in cui il codice viene compilato nativamente. Gli sviluppatori continueranno a creare il codice usando la macchina virtuale e a inviare allo Store pacchetti in linguaggio intermedio Msil. La compilazione avverrà al momento di scaricare l'applicazione e installarla, quindi in modo trasparente allo sviluppatore.





Così appare, invece, il modello di interfaccia per Windows Phone 8.1



UNIVERSALIZZARE UN PROGETTO ESISTENTE

Fra le voci di menu contestuale di un progetto creato in precedenza, per esempio destinato a Windows 81, appare una voce "Add Windows Phone 8.1". Un clic e Visual Studio provvede a creare un progetto parallelo per l'altro sistema operativo e un progetto condiviso. Se si aggiunge il supporto per Windows Phone 8.1 a un'applicazione desktop, avremo la creazione di uno scheletro completo di interfaccia per il telefono, ma solo per quello che riguarda la struttura applicativa, cioè l'insieme dei file necessari per un'interfaccia utente, come App.xaml e MainPage.xaml, oltre a tutta l'infrastruttura di Properties, References e Assets. L'interfaccia creata automaticamente è vuota e starà a noi trasportare il codice Xaml di descrizione dell'interfaccia con i dovuti aggiustamenti. Parimenti vuoto è il progetto condiviso, ma basta spostare col mouse le parti condivisibili - per esempio le cartelle Common e DataModel - e cancellarle dal progetto originario, per ristrutturare l'applicazione senza perdere un secondo in porting. Visual Studio provvede a aggiornare i riferimenti e possiamo eseguire il codice originario senza alcun intervento.

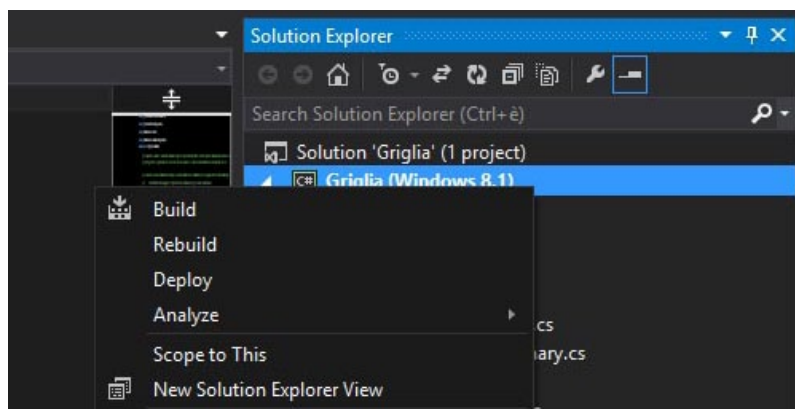
CONCLUSIONI

Non ci soffermiamo qui su altre novità importanti, ma di nicchia, come una prima release di .net native (di cui abbiamo parlato nel box nella pagina precedente), una complessa tecnologia di compilazione trasparente che permette di utilizzare C++ con la

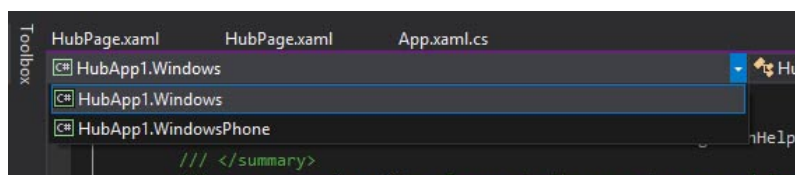
stessa produttività del C# ; tralasciamo anche la versione 1.0 di TypeScript, un linguaggio derivato da JavaScript con l'aggiunta della tipizzazione delle variabili e l'aggiornamento di Team Foundation Server.

L'unificazione della piattaforma Windows è un passo importante, da cui gli sviluppatori hanno indubbiamente molto da guadagnare. Microsoft ha anche compiuto sforzi notevoli per preparare un facile percorso di evoluzione verso Windows Phone 8.1 per chi sta già sviluppando applicazioni destinate a Windows Store. Per chi ancora è

riluttante ad abbandonare il fido Visual Basic 6, suona decisamente la campana, così come suona da un pezzo per Windows XP. Chi sviluppa con Windows Forms può ancora contare sul supporto per la sua nicchia, che secondo Microsoft ha un futuro assicurato nella versione 9 di Windows e, forse, anche nelle successive edizioni. Occorre però considerare se vale la pena di superare la curva di apprendimento e cominciare a traslocare in un territorio più ampio, dove il numero di device che si possono raggiungere si moltiplica per qualche ordine di grandezza. •



Su un progetto Windows 8.1 creato in precedenza appare una nuova voce per l'aggiunta automatica del supporto Windows Phone 8.1,



L'editor di Visual Studio 2013 Update 2 permette di impostare il contesto per cui si compila, le parti non rilevanti del codice vengono immediatamente mostrate in grigio.