



## Una nuvola molto solida, da Microsoft

*La nuova release dell'ambiente Visual Studio introduce un cambio prospettico: lo sviluppo attraverso il browser.*

**A**rriva la versione 2013 di Visual Studio, una novità attesa e importante, perché si tratta dell'ambiente in cui, molto probabilmente, una buona parte dei lettori di questa rubrica passa la sua giornata.

La versione di quest'anno include, come ci aspettiamo dalla software house di Redmond, una quantità di miglioramenti notevole, oltre alla compatibilità con le versioni più recenti dei sistemi operativi Microsoft e la release più recente di .net.

Discuteremo brevemente di questi miglioramenti, in qualche modo in linea con quello che ci si aspetta da un ambiente operativo famoso per la soddisfazione degli utenti, e che Microsoft ha giustamente curato quanto serve per attirare sviluppatori.

Non entreremo, però nei dettagli tipici di una prova di prodotto, perché abbiamo a che fare con una rivoluzione importante, che non si può confinare nella lista di modifiche e miglioramenti a una soluzione che conosciamo.

Nel 2014 Visual Studio introduce una rivoluzione legata al cloud e all'uso della rete, che avrà conseguenze importanti nel modo in cui si lavora e richiederà decisioni da parte di chiunque gestisca un gruppo di sviluppatori, per non parlare dei team di sviluppatori aggregati spontaneamente in qualche progetto cooperativo, sia o meno open source. Nasce Visual Studio online, un ecosistema nel cloud di assoluto interesse, che permette di esternalizzare parte della struttura di sviluppo consentendo a chi è a corto di risorse di razionalizzare le procedure, abbracciare metodologie collaudate, migliorare la qualità del software e proteggere il proprio lavoro nel migliore dei modi, senza investire in infrastrutture di networking e senza

caricarsi della gestione di una rete di sviluppo avanzata.

### Cosa cambia nella nuova versione

Visual Studio 2013 ha un aspetto minimalista e rifugge da sfumature, effetti tridimensionali e decorazioni, come aveva anticipato diverso tempo fa Blend, e in linea con l'estetica di Windows 8.

Questa nuova versione incorpora molte funzioni che sono state sperimentate e distribuite come power tools opzionali. Compiliamo una breve lista delle novità più interessanti. *Generate from usage* permette di creare lo scheletro di una funzione o una classe a partire dal suo uso, una feature che piacerà particolarmente agli sviluppatori che preferiscono partire dai test, o progettare a partire dagli esempi.

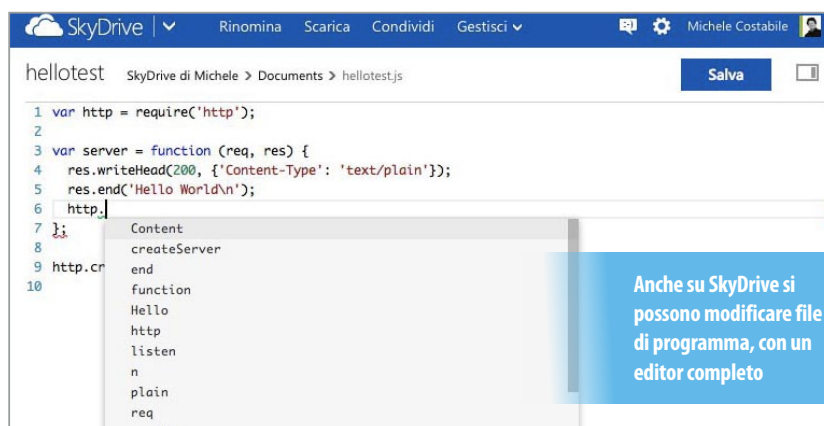
Semplicemente, quando si scrive codice che fa riferimento a un metodo o una classe che non esiste, Visual Studio marca, come al solito, con una sottolineatura gli identificatori inesistenti. Nel menu contestuale, troveremo la possibilità di

creare il codice mancante senza interrompere il flusso di lavoro.

Questa è una delle piccole funzionalità che ci piacciono molto, dato che spesso, in questi tempi di sviluppo agile e scadenze pressanti, l'esigenza di creare o arricchire un elemento di codice non arriva sempre da un'analisi a priori, ma mentre si sta facendo qualcosa d'altro. La barra di scorrimento laterale può essere personalizzata in diversi modi utili, per dare un'indicazione visuale di dove stanno, nel nostro codice, le cose che cerchiamo. Possiamo visualizzare, per esempio, tutte le occorrenze di una stringa, una funzione, che abbiamo sempre apprezzato in Eclipse e che era offerta da qualche add-on di Visual Studio.

La barra di scorrimento si può anche trasformare in una mappa in miniatura del codice, che mostra per intero la forma del file su cui stiamo lavorando.

*CodeLens* è il nome di una finestra di informazioni che si può far comparire vicino a un identificatore per esaminare l'uso del codice dall'interno dell'editor. Si può vedere dove è usata una funzione, per esempio, oppure esaminare la



storia delle modifiche di una funzione, identificare lo sviluppatore che l'ha creata e anche contattarlo senza lasciare l'editor.

La documentazione chiama CodeLens uno Hud (head up display), ispirandosi a quei visori in realtà aumentata installati sugli aerei. Un articolo molto esaustivo su questo complesso strumento si trova su uno dei blog di Microsoft, ecco una url breve: [bit.ly/vs2013codelens](http://bit.ly/vs2013codelens). Purtroppo, CodeLens, che è sicuramente una delle novità che definiscono Visual Studio 2013, è dedicata agli utenti della versione più costosa dell'applicazione, la Ultimate.

*PeekDefinition* è una caratteristica simile: invece di premere F12 per navigare verso la definizione di una funzione, si può premere Alt-F12 per visualizzarla insieme al codice corrente.

Come abbiamo anticipato la novità più significativa non è un miglioramento ma un cambiamento di prospettiva. Si chiama Visual Studio Online.

## Nome in codice: Monaco

La versione online di Visual Studio inizia la gestazione nel 2011, quando Erich Gamma entra in Microsoft, forte di un curriculum spettacolare. A Gamma, infatti, va il credito del libro di maggiore influenza sullo sviluppo object oriented, realizzato insieme a Helm, Johnson e Vlissides, più noti come la banda dei quattro. Design Patterns: Elements of Reusable Object-Oriented Software è il libro che ha reso famoso il movimento dei pattern.

A Gamma va anche il credito di Eclipse, un monumento alla proteiforme duttilità di un ambiente a oggetti scritto nel più rigido dei linguaggi object oriented: Java. Eclipse è sicuramente la dimostrazione pratica del fatto che i pattern

funzionano e che decine di migliaia di classi possono incastrarsi dinamicamente per dare luogo a un ambiente di lavoro mirabilmente configurabile ed estensibile.

Gamma ha anche realizzato il framework più noto per la creazione di test automatici del codice, JUnit, che ha fatto nascere decine di emuli, fra cui NUnit su piattaforma Microsoft.

È molto significativa, quindi, la scelta di Microsoft di affidare a Gamma la responsabilità della creazione del primo avamposto sulla nuvola della sua futura strategia per gli sviluppatori.

Eclipse, fra le altre cose, è stato il primo ambiente di lavoro a consentire realmente di progettare il software strada facendo, sostenendo la crescita dello sviluppo agile.

Gli strumenti di refactoring avanzati, infatti, hanno reso poco costosi interventi pesanti di ristrutturazione del codice e, insieme agli strumenti per l'analisi dell'architettura, hanno reso marginale l'esigenza di strumenti di analisi e modellazione, che oggi sono vissuti più come strumenti per la comunicazione di architetture, che per la definizione a priori di architetture.

Insomma, Gamma ha dimostrato una passione per lo sviluppo agile, di cui ci sono segni evidenti anche in Visual Studio Online.

## Editing nel browser

Vediamo allora di inquadrare subito Visual Studio online e cosa significa per noi. Parliamo innanzitutto di quello che era noto in fase di sviluppo come Monaco, un eccellente ambiente per l'editing di testi, che Microsoft offre gratuitamente all'interno della sua offerta di servizi online. Per esempio, è possibile richiamarlo come editor per

MODIFICA IN VISUAL STUDIO ONLINE

ATTIVATO

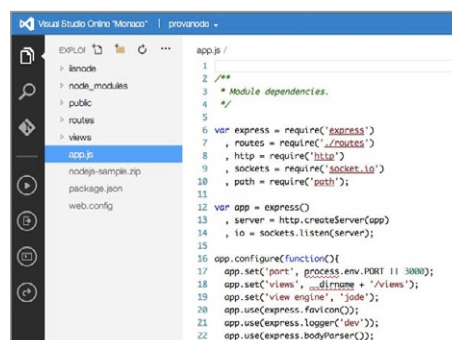
DISATTIVATO

PREVIEW

**Un progetto web su Azure deve essere abilitato per l'uso dell'editor di Monaco**

- ⌚ Reimpostare le proprie credenziali del profilo di pubblicazione
- 🔌 Disconnetti da Dropbox
- ➔ Modifica in Visual Studio Online **ANTEPRIMA**

**Dopo l'abilitazione delle modifiche avremo un link nella sidebar destra per modificare i file nel browser**



**L'interfaccia dell'ambiente Monaco, come si nota nell'immagine qui sopra, è scarna e funzionale**

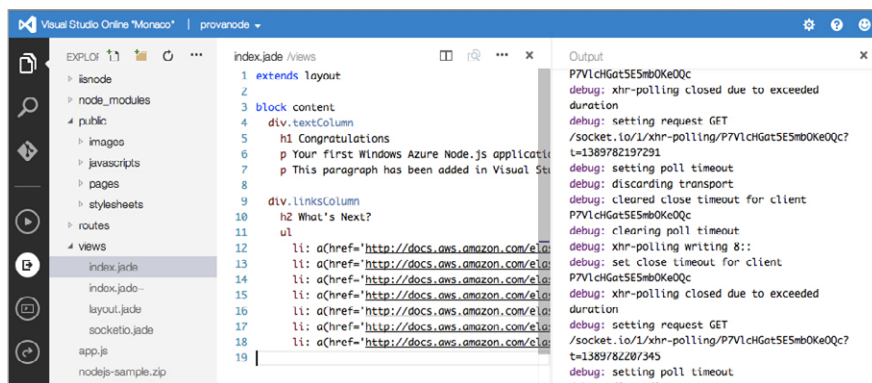
i siti web ospitati su Windows Azure. Allo stesso modo, l'editor di file di SkyDrive riconosce i file di programma e offre le ricche funzioni di editing di Visual Studio.

Con questo, Microsoft, si mette in pari con l'offerta di editor ospitati nel browser che abbiamo visto nel numero scorso.

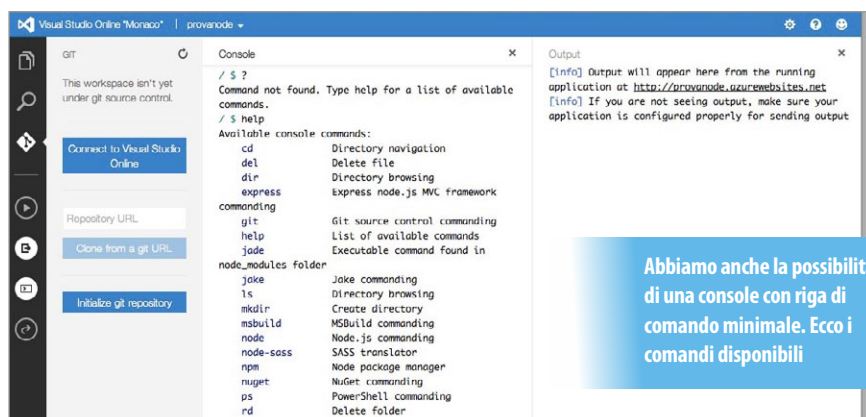
Inoltre, Windows Azure diventa una piattaforma molto appetibile per gli sviluppatori che desiderano sperimentare e avventurarsi. Azure, infatti, offre hosting gratuito, una scelta di linguaggi sufficientemente ampia, un cruscotto di gestione semplificato quanto è possibile, un modello di fatturazione con ampie fasce gratuite e un ottimo controllo sulla previsione di spesa. Aggiungendo funzioni avanzate di editing online, abbiamo una soluzione completa per lo sviluppo e l'hosting di siti web, anche attivi, basati su Asp, Node.js, Php o Python e, con poca spesa aggiuntiva, anche con un database alle spalle.

Nel modello gratuito di uso non c'è l'obbligo di una licenza open source per i progetti, quindi si tratta di una proposta ideale per sperimentare, anche in gruppo e collaudare in diretta le proprie soluzioni.

Sarebbe interessante, ma, per fortuna, non finisce qui.



**Si può avviare il codice dell'applicazione web da dentro l'editor. A destra il log di esecuzione.**



Abbiamo anche la possibilità di una console con riga di comando minimale. Ecco i comandi disponibili

## Sorgenti nella nuvola, anche con git

Visual Studio online (visualstudio.com) offre spazio per l'archiviazione dei sorgenti e la creazione di progetti. La registrazione è gratuita, l'hosting non è a pagamento fino a cinque utenti e lo spazio occupato può essere gestito remotamente attraverso l'infrastruttura di Team Foundation Server, ma anche usando Git, l'ambiente di condivisione di sorgenti open source più avanzato.

La disponibilità di Git, significa, che possiamo ospitare nella nuvola di Microsoft anche progetti Java gestiti con Eclipse o applicazioni gestite con XCode. Il protocollo Git libera, infatti, dalla necessità di uno specifico ambiente Ide sul client.

## Gestione del gruppo nella nuvola

Visual Studio Online offre strumenti integrati per la gestione del gruppo di lavoro e del ciclo di vita dell'applicazione.

La creazione di un progetto, per esempio, richiede la scelta di un modello di organizzazione del processo. Ci sono tre processi disponibili: *Scrum*, *Agile* e *Cmmi*, che corrispondono a diversi modi di organizzare lo sviluppo. Scegliendo uno di questi template si organizza il processo scegliendo gli strumenti e il metodo in modo da non avere a che fare con burocrazia superflua. Per il nostro progetto di test abbiamo scelto un modello *Agile*, il più essenziale. Fra le opzioni disponibili c'è la possibilità di usare Kanban e Scrum, due metodologie piuttosto popolari. La riunione mattutina del gruppo può avvenire davanti alla bacheca del backlog, che può essere riordinato trascinando gli

elementi col mouse. Il sistema integrato permette di tenere sotto controllo la velocità di avanzamento del gruppo di lavoro e di fare proiezioni sulla velocità del progetto. Naturalmente, non è difficile integrare un server Sharepoint per gestire i documenti comuni e la comunicazione interna al gruppo. Infine, non manca un bug database, una struttura molto utile per pianificare le attività e fare analisi di tendenza sulla qualità del software, che spesso viene dimenticata per non doversi prendere la briga di installare un sistema di gestione, fosse anche gratuito e open source come Bugzilla.

## Build e test nella nuvola

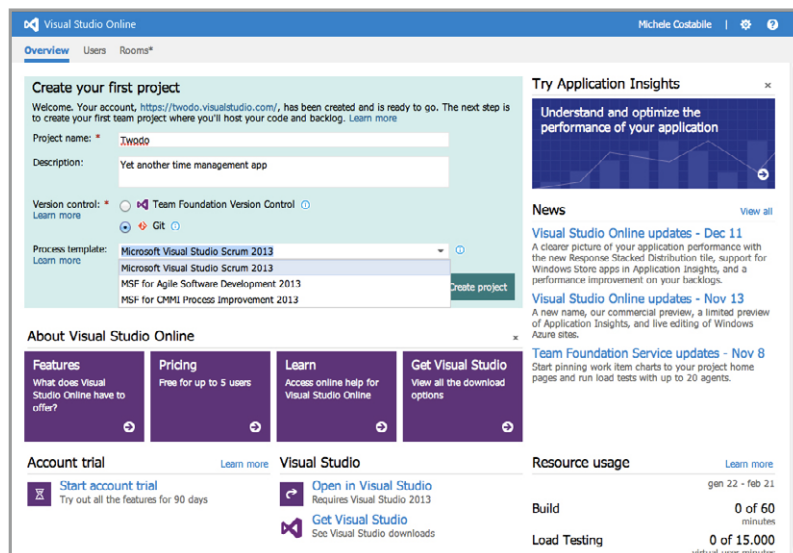
L'ultimo tassello dello sviluppo software, quello più essenziale, che produce l'eseguibile a partire dai sorgenti, può essere delegato alla nuvola attraverso la creazione di robot che

eseguono build periodici, per esempio build scatenati da un commit dei sorgenti.

I passi sono molto semplici: bisogna creare una definizione di compilazione, andando alla pagina di compilazione in Team Explorer, creando una *definizione di compilazione*, scegliendo un nome per il target e un trigger che scatena il build, per esempio il check-in di nuovo codice e impostando il percorso delle cartelle che contengono i sorgenti da compilare.

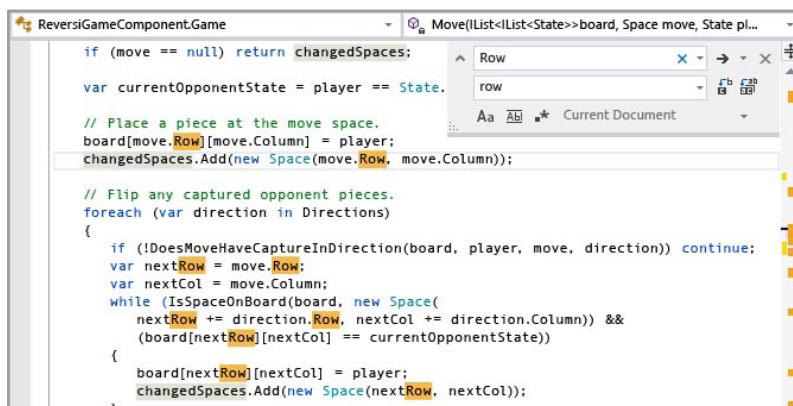
Al termine di ogni consegna del codice ci sarà una compilazione.

È molto utile avere compilazioni automatiche, soprattutto se il gruppo di lavoro è distribuito geograficamente, su più sedi. Se una compilazione, infatti, fallisce per qualche ragione, si tratta di un evento che richiede immediata attenzione da parte del gruppo e di chi ha consegnato le ultime modifiche. Ricordiamo che Git permette di creare un repository personale sulla macchina di ogni sviluppatore, un repository di gruppo per ogni laboratorio e un repository generale per l'intero team. In un'organizzazione simile, è perfettamente accettabile avere degli errori nel repository personale, è un problema averne sull'archivio condiviso, presumibilmente usato per i test delle funzionalità create da ogni gruppo. Quello ospitato su visualstudio.com dovrebbe essere il codice nella sua forma definitiva, integrato col lavoro di tutti, quindi una consegna che genera errore è un problema piuttosto grave.

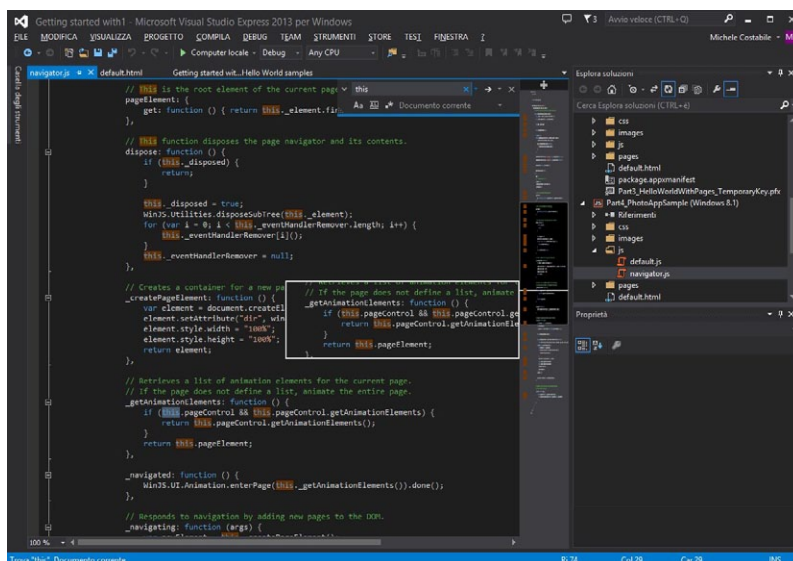


**Creare un progetto su Visual Studio Online è semplicissimo e gratuito. In fase di creazione specifichiamo se usare Git o Team Foundation Server e scegliamo il modello agile da adottare.**





L'editor di Visual Studio ha una scrollbar potenziata, prima disponibile solo attraverso l'installazione di componenti add-on. Qui mostriamo la lista delle occorrenze di una stringa.



L'interfaccia di Visual Studio con il tema di colori scuro. Si noti la scrollbar che abbiamo trasformato in una miniatura del codice. La personalizzabilità dell'ambiente è elevata.

Per tenere d'occhio la qualità, si può configurare l'ambiente in modo da eseguire una serie di test ad ogni consegna. Ricordiamo, di passaggio, che Erich Gamma è anche l'autore del più popolare framework per l'esecuzione di test automatici: JUnit, che ha dato origine a tutti gli altri framework. L'esito dei test di ogni build è una misura importante del progresso di un prodotto. Come abbiamo visto, la versione maggiore di Visual Studio permette di tracciare le modifiche nella storia dei check-in di una pagina e associare una riga allo sviluppatore che ne è responsabile. Questo, insieme a strumenti di comunicazione, come una chat room di progetto, rende immediato il dialogo fra i membri di un team. Per inciso, le diverse comunità di sviluppatori che gravitano intorno a Microsoft, come CodePlex, o comunità

che si formano intorno a progetti centrati su Windows, possono avere a disposizione strumenti potentissimi per la comunicazione, la condivisione e il controllo delle versioni del codice, l'esecuzione di test, l'assegnazione dei bug e la consegna delle release.

## La rivoluzione della nuvola

Ci sembra quindi di poter affermare che il ruolo della nuvola è la novità principale della versione 2013 di Visual Studio. Le possibilità offerte da questo cambio di prospettiva faranno certamente riflettere sulla qualità della pianificazione tutte le persone che hanno la responsabilità di un gruppo di sviluppo. Microsoft con visualstudio.com e Visual Studio offre un sistema completo, dotato di un corredo di strumenti

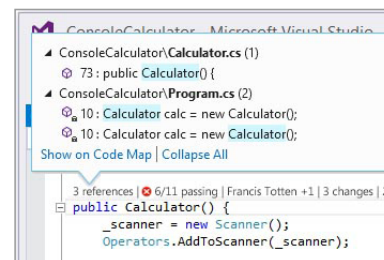
esteso, che incorpora il necessario per seguire metodologie apprezzate in tutti gli ambienti.

Paradossalmente, chi è stato impermeabile alle tendenze e non ha mai considerato l'idea di costruire un repository di sorgenti condiviso, con un backup automatizzato e affidabile, insieme a un sistema di build e test automatici per l'integrazione continua, può passare da zero al 100% di risposte positive nella famosa lista di dodici domande di Joel Spolsky ([www.joelonsoftware.com/articles/fog0000000043.html](http://www.joelonsoftware.com/articles/fog0000000043.html)).

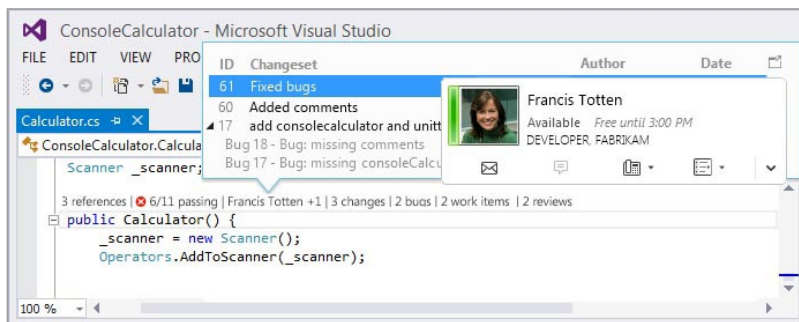
## Mettiamoci al lavoro: una soluzione web

Un primo modo di vedere la novità di Microsoft è usare l'eccellente editor di Visual Studio nel browser, per un progetto web. Abbiamo quindi provato a usare gli strumenti di editing di Visual Studio su un sito web gratuito ospitato su Windows Azure e basato sulla piattaforma server Node.js.

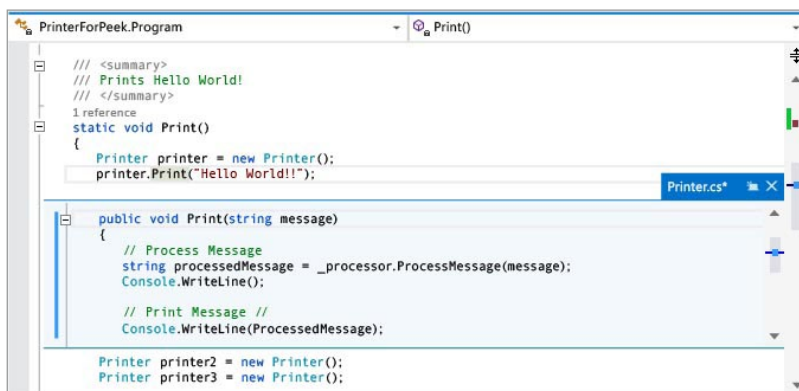
L'inizio è molto semplice: basta aprire il cruscotto di gestione del sito nel portale di Windows Azure e quindi abilitare l'opzione "modifica in Visual Studio Online". Dopo avere abilitato le modifiche, nelle opzioni rapide della sidebar di destra comparirà anche "modifica in Visual Studio Online". Un clic e appare l'editor, con una vista sui file che compongono l'applicazione (provanode.azurewebsites.net). L'editor è quello di Visual Studio, con la familiare evidenziazione degli identificatori sotto il cursore e la consueta Intellisense attiva, anche sulle strutture dati di Node.js e dei suoi package aggiuntivi. Ci sono due temi visuali, uno basato su VisualStudio, con fondo chiaro e parole chiave in blu, l'altro tema ha un fondo nero e un tema di colori piuttosto gradevole per dare riposo agli occhi o lavorare dove c'è poca luce. Chi ama la brevità, viene da unix o, semplicemente, ha dei



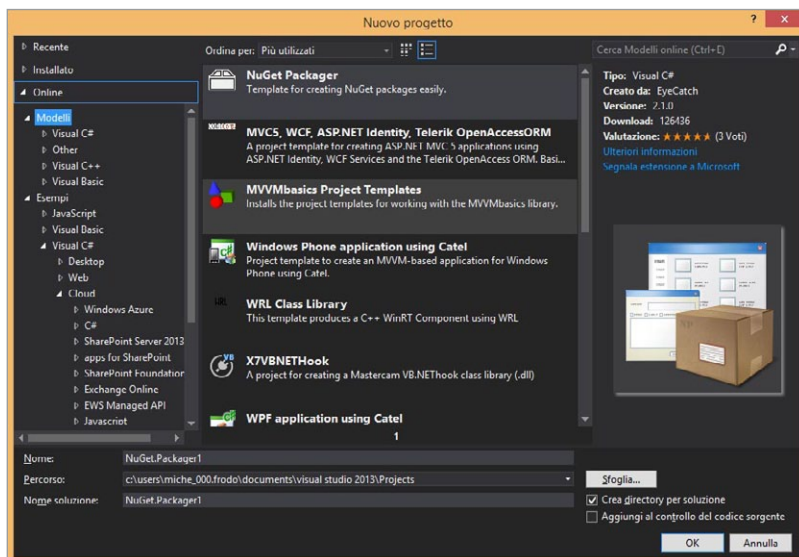
CodeLens è il nome di una finestra transiente che raccoglie tutte le informazioni possibili su un elemento di codice, per esempio la dichiarazione.



**CodeLens** è una funzionalità che si può usare anche per scorrere la storia di una riga di codice e contattare in chat lo sviluppatore che ci ha lavorato.



**Peek Declaration** è una funzione che mostra la dichiarazione di una funzione senza essere obbligati ad abbandonare la porzione di codice in cui la si sta esaminando.



La libreria di esempi che possiamo scegliere dal menu di creazione progetti è vastissima.

riflessi condizionati nelle dita ritroverà buona parte degli shortcut da tastiera di Emacs, come `ctrl-a` per andare a inizio riga e `ctrl-k` per cancellare oltre il cursore, in aggiunta ai consueti tasti di movimento e cancellazione. Visual Studio Online non ci dà solo l'editor. Nella barra laterale di sinistra troviamo icone per aprire la console e la finestra di output. La finestra di output mostra tutte le stampe di

tracciamento inviate alla console dal codice JavaScript dell'applicazione, mentre la console consente di operare da una riga di comando. I comandi della console sono lo stretto necessario per gestire i file, lanciare Git e aprire un archivio zip. I comandi contenuti nei pacchetti aggiuntivi sono attivi e – sorpresa – troviamo anche il completamento automatico delle opzioni di npm, il sistema di gestione dei

pacchetti opzionali di Node.js. Con la struttura dei prezzi attuale di Windows Azure, possiamo avere gratis un sistema completo di hosting e un Ide, di altissima qualità, che non impone neanche il vincolo dell'apertura dei sorgenti. Insomma, una soluzione molto interessante per la sperimentazione di idee a latere di un progetto, o la collaborazione in rete con sviluppatori distanti.

## Mettiamoci al lavoro da riga di comando

Non occorre avere Visual Studio nemmeno per creare una collezione di sorgenti online, grazie al supporto per Git. Possiamo, infatti, usare `visualstudio.com` come un repository Git con ambiente di sviluppo integrato. Questa soluzione è utile per chi non desidera installare Visual Studio, oppure gestisce un progetto in un altro ambiente, come Java, con un Ide specifico. Possiamo lavorare da riga di comando occorre abilitare l'autenticazione semplice su Visual Studio Online, come descritto nella documentazione ([bit.ly/vs2013gitcl](http://bit.ly/vs2013gitcl)). Ci colleghiamo Visual Studio Online, usando l'account Microsoft su `visualstudio.com`, creiamo un'applicazione chiamata Twodo, con l'idea di sperimentare prima o poi certe idee sulla gestione del tempo. Dopo avere abilitato la gestione dell'autenticazione basic authentication, possiamo dare il comando

```
git clone https://
twodo.visualstudio.com/
DefaultCollection/_git/Twodo
```

L'output è

```
Cloning into 'Twodo'...
error: git-credential-osxkeychain
died of signal 11
Username for 'https://twodo.
visualstudio.com':
```

a questo punto inseriamo login e password che abbiamo creato in Visual Studio Online, credenziali create ad hoc, diverse da quelle dell'account Microsoft, e il processo è completo: abbiamo i nostri sorgenti nella cartella Twodo e siamo salutati da una versione ASCII del logo di Visual Studio. Brivido anni '80. Dopo avere creato un file `Hello.cs` minimale, come questo:

```
// A Hello World! program in C#
using System;
```

```

Macintosh:qq mico$ history | grep clone
509 git clone https://twodo.visualstudio.com/DefaultCollection/_git/Twodo
510 git clone https://twodo.visualstudio.com/DefaultCollection/_git/Twodo
511 git clone https://twodo.visualstudio.com/DefaultCollection/_git/Twodo
558 git user clone
559 history | grep clone
Macintosh:qq mico$ 1511
git clone https://twodo.visualstudio.com/DefaultCollection/_git/Twodo
Cloning into 'twodo'...
error: git-credential-osxkeychain died of signal 11
Username for 'https://twodo.visualstudio.com':
Password for 'https://twodo.visualstudio.com':
error: git-credential-osxkeychain died of signal 11
remote:
remote:      FTFx
remote:      FSSSSSSSSS
remote:      FSSSSSSSSS
remote:      TSSF      FSSSSSSSSSSS
remote:      SSSSSSF      FSSSSSSST SSSSS
remote:      SSSSSSSSfSSSSSSSt SSSSS
remote:      SS      tSSSSSSSSSg SSSSS
remote:      SS      FSSSSSSST SSSSS
remote:      SS      FSSSSSFSSSSSF SSSSS
remote:      SSSSSST      FSSSSSSSt SSSSS
remote:      SSSSt      FSSSSSSSSSSS
remote:      FSSSSSSSSS
remote:      FSSSSS
remote:      FSSS      CMO
remote:
remote: Microsoft (R) Visual Studio (R) Team Foundation Server
remote:

```

**Nulla ci vieta di usare Git da riga di comando per lavorare col codice depositato su visualstudio.com.**

```

namespace HelloWorld
{
    class Hello
    {
        static void Main()
        {
            System.Console.
WriteLine("Hello World!");

            // Keep the console
window open in debug mode.
            System.Console.
WriteLine("Press any key to
exit.");
            System.Console.
ReadKey();
        }
    }
}

```

Ci basta la sequenza di comandi

```

git add Hello.cs
git commit
git push remote origin

```

Per notificare Git che deve gestire il file Hello.cs, aggiungere il file all'archivio Git e infine sincronizzare la copia locale dell'archivio con il server remoto. Adesso il nostro file è online e visibile a tutti gli utenti abilitati. Solamente, non siamo stati in grado di compilare il progetto online, perché non ci è riuscito di creare una build definition in Visual Studio Online. Anche la documentazione illustra solo come crearne da dentro Visual Studio. Si tratta di una limitazione minima: esistono diverse versioni gratuite di Visual Studio nella linea Express, tutte scaricabili liberamente. Secondariamente, non è logico usare Visual Studio Online per applicazioni su piattaforma Microsoft, conviene lavorare sul desktop, sia per poter sviluppare e provare localmente le cose prima di spedirle, sia per avere il tempo di risposta e la potenza di

un'applicazione locale. Ha senso vedere visualstudio.com come un repository Git per gestire progetti in linguaggi non Microsoft, per esempio applicazioni Java, sviluppate con Eclipse o app OS X create con XCode, che concorrono a formare un'applicazione enterprise insieme a soluzioni Microsoft. La disponibilità di Git assicura che non ci sono ostacoli per chi voglia collaborare con la piattaforma che preferisce.

## Lo sviluppo nativo

Naturalmente, l'ambiente privilegiato per dialogare con Visual Studio Online è Visual Studio in versione desktop, che per esempio è richiesto per creare agenti che fanno il build nella nuvola.

Anche con la più semplice e gratuita versione Express possiamo gestire un archivio software Git locale e collegarci a un repository remoto, eventualmente su visualstudio.com avendo a disposizione tutti gli strumenti necessari per tenere traccia delle versioni, studiare le differenze e integrare le modifiche.

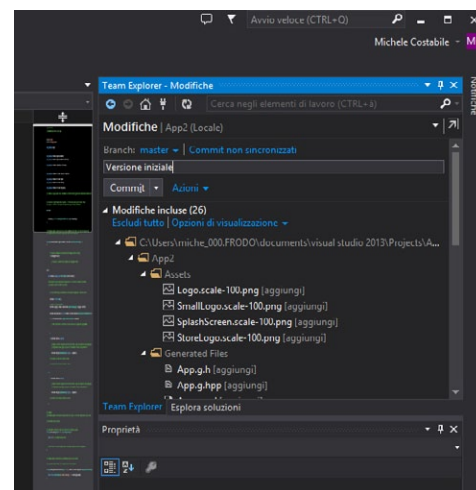
L'ultimo nato di casa Microsoft ha un disegno snello, decorazioni ridotte al minimo e un set di colori molto minimalista. Fa l'effetto di uno strumento di liuteria o di una automobile di qualità: tutto funziona senza sorprese e si usa senza sforzo. Questo, in estrema sintesi è quello che ne possiamo dire. Esistono diverse versioni a pagamento di Visual Studio, con un insieme di funzionalità crescente in proporzione al prezzo. Non c'è spazio per discutere in dettaglio delle diverse funzioni, ma troviamo una utile tabella di raffronto all'indirizzo [web.visualstudio.com/products/compare-visual-studio-products-vs](http://web.visualstudio.com/products/compare-visual-studio-products-vs). Un catalogo dei prezzi dell'applicazione e di eventuali abbonamenti ai servizi della developer network Msdn si può trovare qui [visualstudio.com/products/how-to-buy-vs](http://visualstudio.com/products/how-to-buy-vs).

## Conclusioni

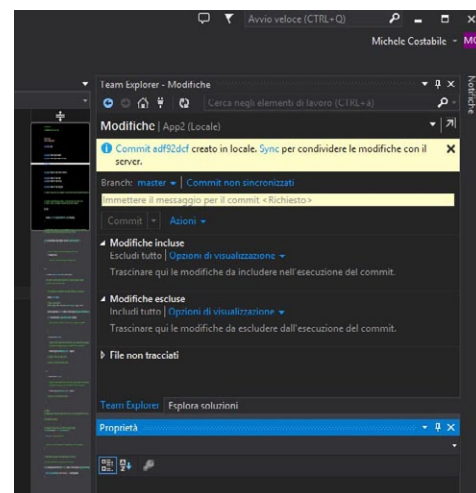
Le conclusioni, in realtà, le abbiamo già anticipate diversi paragrafi sopra, comunque tiriamo le somme. L'ultima versione di Visual Studio è, come di consueto, lo stato dell'arte per quanto riguarda la programmazione su Windows e dà un ventaglio di possibilità applicative e di strumenti, sempre più ampio.

Questa volta la parte online del prodotto ruba la scena all'applicazione desktop

e richiede riflessioni e valutazioni per quanto riguarda le opportunità di organizzazione e di gestione della qualità. Sono interessanti le possibilità di gestione di siti web gratuiti su Azure, compresi siti web attivi, ai quali si può dare un backend con database a un prezzo contenuto. L'apertura verso strumenti per il lavoro in team open source e di elevata qualità, come Git, offre possibilità integrazione per le attività che richiedono ambienti diversi da Visual Studio, per esempio una app per Android che fa parte di un corredo di strumenti enterprise. Si aprono, infine possibilità inattese per gruppi di sviluppatori estemporanei, aggregati in rete per un progetto non necessariamente open source.



**Anche nella versione Express gli strumenti per il controllo delle versioni sono potenti. Ecco un commit nell'archivio Git locale**



**Siamo pronti per fare un push delle modifiche nel repository remoto. Le finestre di aiuto a fondo giallo sono utili per orientarsi all'inizio.**