

# Prefazione

Il libro che avete fra le mani non è un libro comune; rimettetelo sullo scaffale, se non avete intenzione di iniziare un viaggio affascinante, che durerà una vita, vi metterà alla prova e, qualche volta, sarà molto impegnativo. Il volume di Ferdinando decisamente non è del tipo “impara Git in due giorni senza fatica”.

In effetti, esistono due modi di affrontare Git.

Forse siete il tipico sviluppatore (e, probabilmente, un po' annoiato), le cui attività quotidiane sono orientate più dalle scadenze da rispettare e dalle attività da completare che dalla passione e dal gusto della sfida. Se così fosse, non preoccupatevi; limitatevi a ignorare questo libro e piuttosto chiedete a un collega di insegnarvi il minimo indispensabile sui comandi di Git (come clonare, effettuare checkout, commit e fusioni) per sopravvivere in futuro e fate affidamento su Google o su Stack Overflow in caso di bisogno.

Altrimenti c'è la possibilità che siate un programmatore appassionato, ardente e pragmatico, ben disposto a mettere in dubbio umilmente tutte le proprie convinzioni.

In questo secondo caso, vi farà molto piacere scoprire che Git è solo esternamente uno strumento semplice, per quanto molto efficiente e potente, per il controllo delle versioni del codice sorgente; è, in realtà, un punto di svolta nella storia dell'informatica, e questo libro è la guida perfetta per scoprire questo suo lato.

Non ho molti esempi di punti di svolta analoghi nella storia dell'IT. Al di là del World Wide Web, che è l'esempio più ovvio, posso pensare a Unix, con la sua filosofia “Fai una cosa e falla bene” e i suoi comandi componibili all'infinito, a TDD di Ken Beck o a Emacs, con le sue stupefacenti e infinite possibilità di estensione, e a Docker, che ha reso disponibile a tutti l'infrastruttura come codice e ha fatto esplodere il movimento DevOps. Incontrate una di queste tecnologie o di queste idee e di sicuro nella vostra vita di sviluppatori nulla sarà più come prima. Credo che lo stesso valga per Git, che state per imparare a conoscere con l'aiuto di questo volume.

A differenza della maggior parte degli altri sistemi di versioning, il progetto di Git è stato sviluppato sull'idea fondamentale che l'attività di sviluppo sia intrinsecamente un esercizio di comunicazione e di collaborazione. Ovviamente potete decidere di usare Git semplicemente come se fosse un Subversion o un TFS potenziato, ma vi sfuggirebbe la sua bellezza: Git fa leva su alcuni concetti molto semplici, ma potenti, che consentono agli sviluppatori di costruire reti scalabili e collaborative.

Git appartiene ai pilastri fondamentali del movimento open source: Richard Stallman, Eric S. Raymond e altri giganti ne hanno fornito le fondamenta ideologiche; la Free

Software Foundation e licenze come la GPL gli hanno dato la necessaria autorevolezza legale; GNU e Linux sono le concretissime e funzionanti implementazioni di queste idee e Git, con i suoi concetti sociali inscritti nel DNA, è il quarto pilastro, lo strumento che ha davvero consentito alla comunità di costruire reti come GitHub, che hanno decretato il successo dell'open source su scala globale.

Se dunque siete programmatori appassionati e volete entrare nella dimensione sociale di Git e nell'universo che gli ruota attorno, imparare solo i nudi comandi può essere riduttivo. Dovete cogliere davvero il significato profondo di che cosa sia un *fork*, di come gestire una *richiesta di pull*, e di che cosa sia un *rebase*, al di là dei puri aspetti tecnici.

Per arrivare a questo risultato, avrete bisogno della guida esperta di qualcuno che sia appassionato quanto voi.

Credo che sia esattamente questo il valore del libro che state leggendo: non è un sostituto delle *man page* di Git né una scorciatoia veloce per imparare Git senza fatica. Al contrario, essendo un ampio tutorial che vi costringe a sporcarvi le mani, vi porterà in un viaggio profondo, non necessariamente facile ma sempre entusiasmante.

Questo volume vi porterà nel cuore del funzionamento interno di Git, così che, non appena ne riemergerete, potrete ragionare su tutte le immaginabili peculiarità di comportamento anche mentre lavorate con comandi superficiali.

È un libro che attribuisce più valore al codice che alle parole. Perciò aprite una shell, mettetevi comodi e godetevi il viaggio!

*Arialdo Martini*

Innovation Manager, TCPOS SA

Per me è un piacere scrivere la prefazione a questo libro. Guiderà neofiti e utenti abituali verso un uso esperto dello strumento che è diventato lo standard di fatto per il controllo delle versioni del codice sorgente: Git.

Sette anni fa, Git emergeva dalle grandi difficoltà che Linus Torvalds stava affrontando nella gestione di un team su scala globale impegnato sul kernel di Linux. Sulla scena dei sistemi distribuiti di controllo delle versioni si sono presentati molti concorrenti, ma alla fine quello che si è diffuso è stato Git, grazie alla sua semplicità. Molte aziende lo hanno adottato: GitHub, per esempio, ha rivoluzionato il modo in cui le persone contribuiscono a progetti open source.

Se siete sviluppatori di software per professione, o contribuite a progetti open source, o entrambe le cose, semplicemente non potete ignorare Git. Non potete sperare che la prima GUI creata su di esso sia sufficiente per svolgere il vostro lavoro. Questo libro, costruito con grande cura da Ferdinando, vi aiuterà a comprendere dalle basi fino alle funzioni più avanzate che lo strumentario di Git ha da offrire. Il vostro flusso di lavoro sicuramente trarrà vantaggio dall'aver ogni cosa sotto controllo; potrete muovervi facilmente lungo la cronologia dei vostri progetti come in una macchina del tempo.

Il volume illustra i meccanismi interni di Git e i flussi di lavoro migliori da adottare in ogni progetto software.

Dopo averlo letto non perderete mai più neanche un'ora preziosa di lavoro. Sono pronto a scommetterci.

*Giovanni Toraldo*

Lead Software Developer, Cloudesire.com