

Esercizio GitHub - progetto collaborativo

Antonio Pugliese Matricola: 987249

Titolo

Imparare i meccanismi di base di Git tramite GitHub.

Scopo

Uno dei maggiori punti di forza del Software Libero, è la caratteristica dell'*open source*, ovvero il codice sorgente viene condiviso a tutti gli utenti che sono quindi in grado di studiarne il funzionamento interno o perfino modificarlo per adattarlo alle proprie necessità. Per permettere questo viene utilizzato Git, un software per il controllo di versione distribuito. Per simulare questo processo, in questo esercizio, due utenti dovranno collaborare: il primo si occuperà di creare un progetto (*repository*) su GitHub (la piattaforma di Git più utilizzata) e aggiungere (*commit and push*) un file. L'altro utente, invece, si occuperà di modificare e aggiornare il file aggiunto (*pull request*). Sarà interessante vedere come si comporta la piattaforma in queste situazioni di collaborazione.

Destinatari finali

Utenti con conoscenze base in Informatica.

Durata

30 minuti

Livelli Arcobaleno CDT

L3 - Education

Difficoltà (relativa all'utente)

Facile-Media

Strumenti

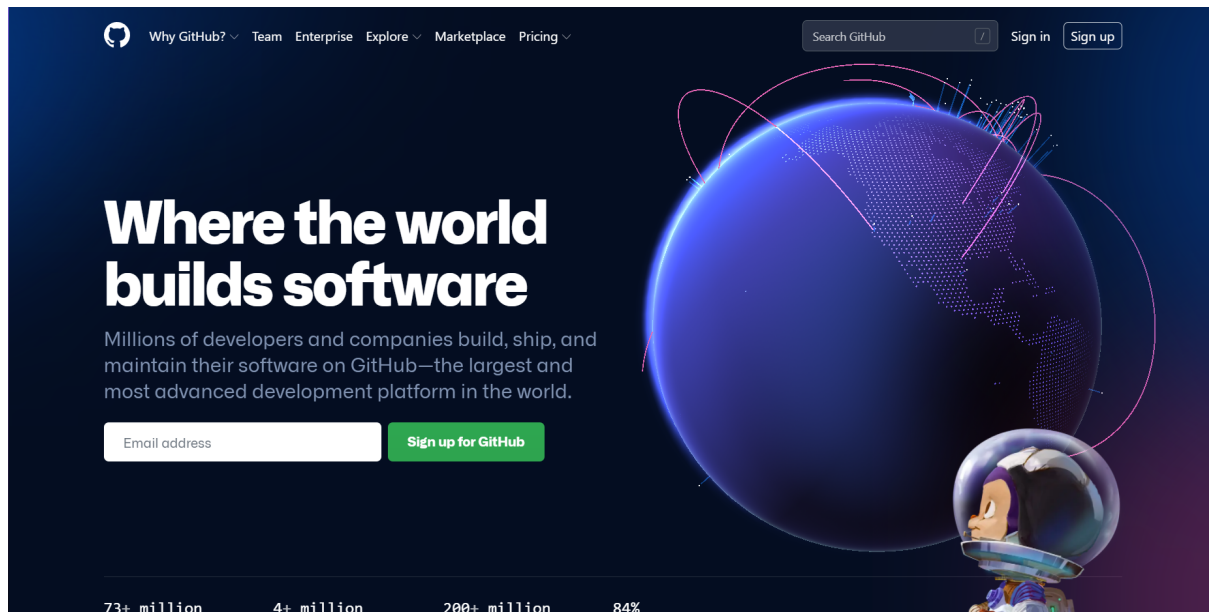
- Un PC sul quale sia installato un browser
- Una connessione Internet

Costi

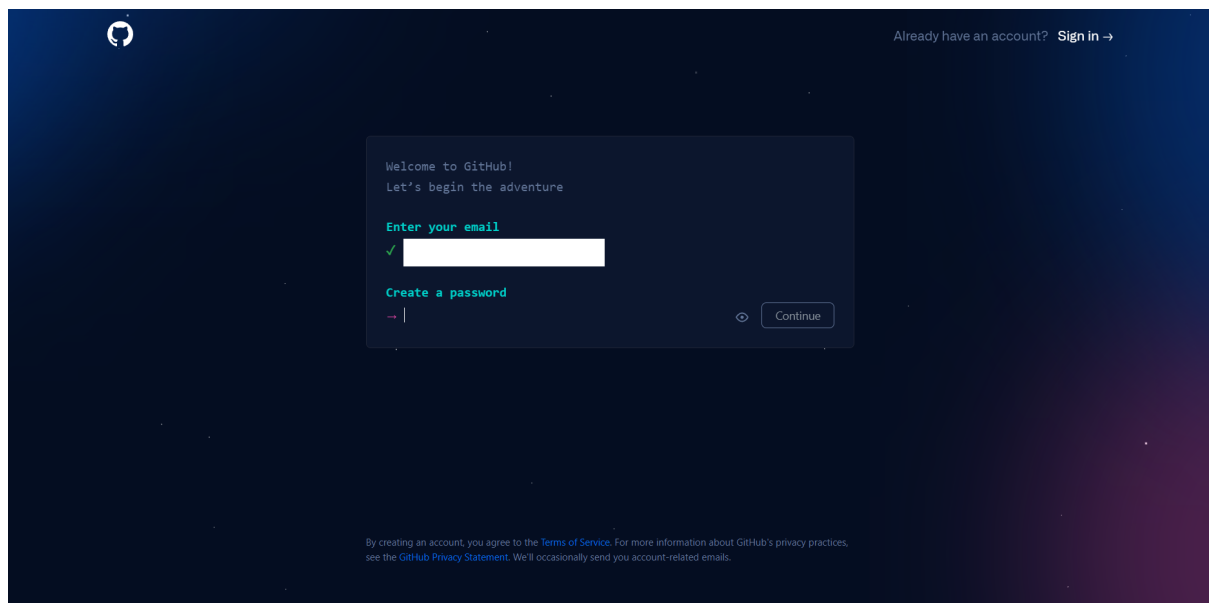
Gratuito

Istruzioni (Utente 1)

1. Tramite browser, collegarsi al sito: github.com. Per poter caricare il proprio codice, occorre creare un account. Se l'utente dispone già di un account GitHub, effettuare l'accesso cliccando su "Sign in" in alto a destra, altrimenti, seguire il punto 2.

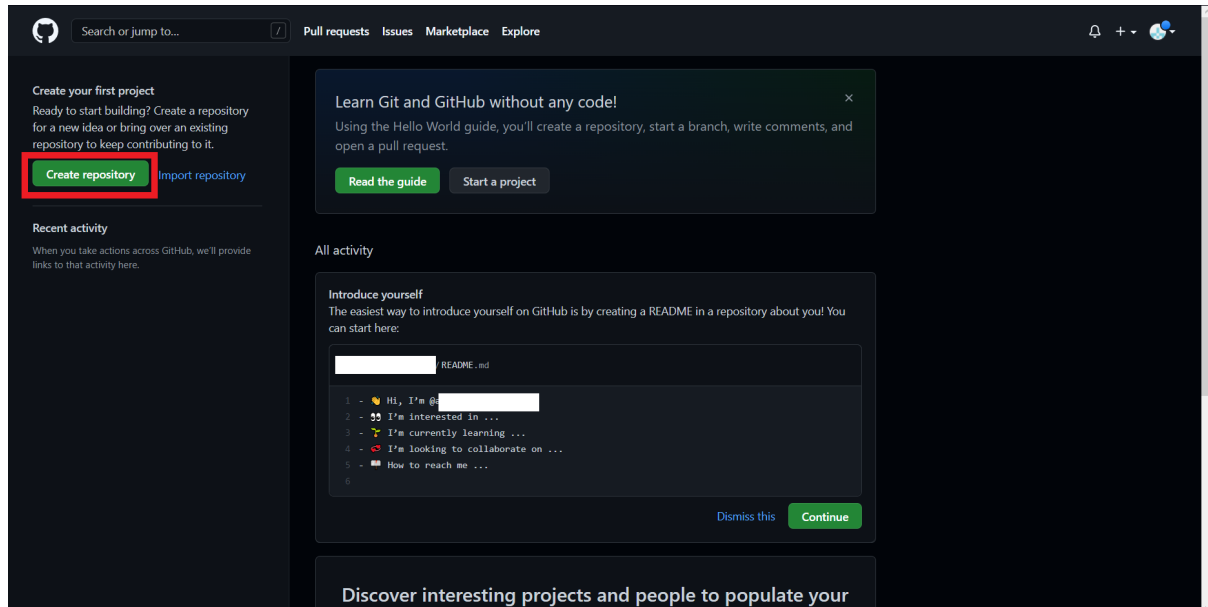


2. Per la creazione di un nuovo account, cliccare su "Sign up", GitHub offre un'interfaccia interattiva per guidare passo per passo l'utente. Seguire i passaggi fino alla corretta creazione dell'account.

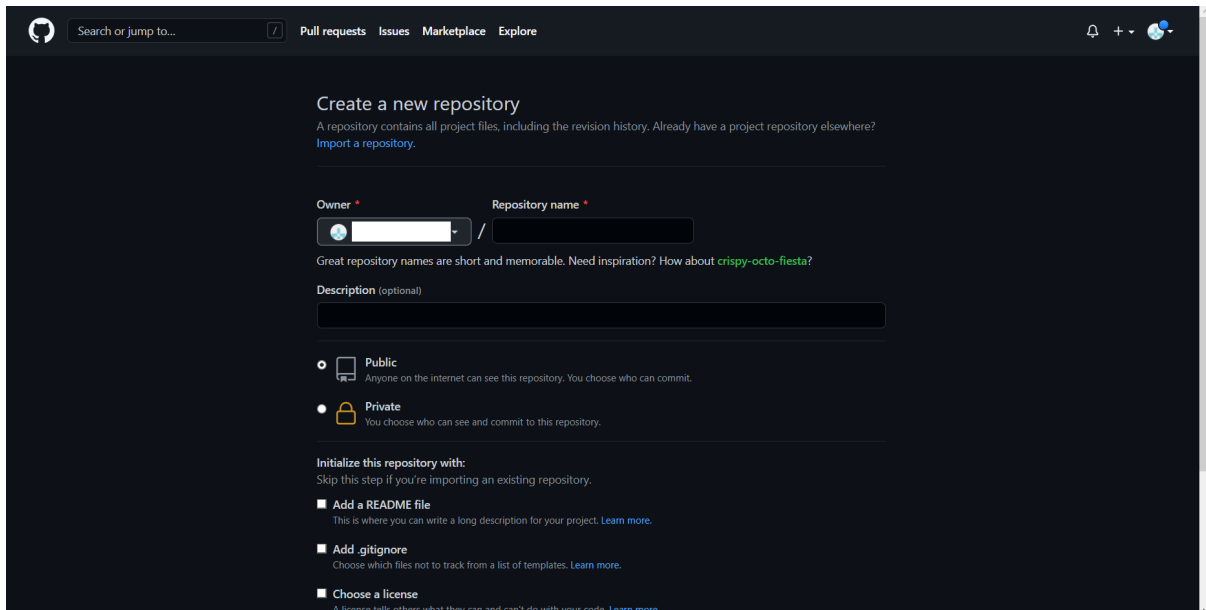


Creare un repository

1. Una volta effettuato l'accesso, viene visualizzata la *dashboard* dell'utente. Per creare un nuovo repository. Cliccare su "Create repository".



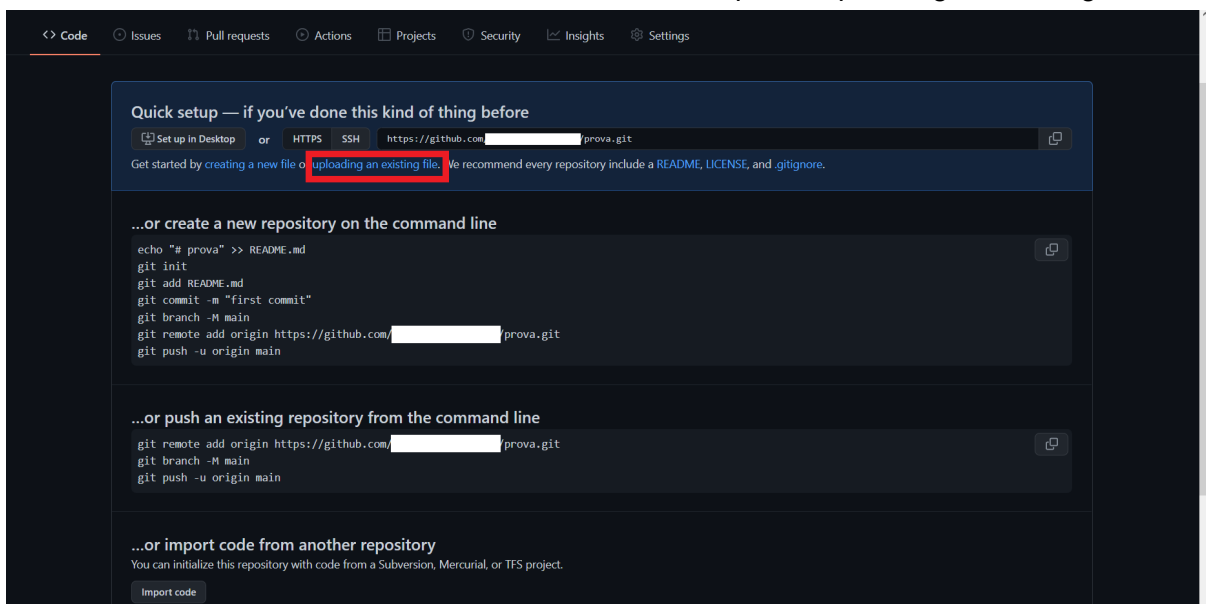
2. Ci sono tre voci da compilare:
 - Repository name: Nome del repository, dovrebbe essere il più significativo possibile per identificarlo univocamente. Per questo esercizio va bene semplicemente il nome "prova".
 - Public/Private: Se pubblico, il repository sarà accessibile a chiunque. Questo va scelto se l'utente desidera che chiunque possa accedere al suo codice. Se privato, solo il proprietario e gli utenti con permessi possono accedere al repository. Per questo esercizio, dove viene mostrata la collaborazione tra utenti, selezionare "public".
 - "Initialize this repository with...": Inizializzazione di file automaticamente alla creazione del repository. Per questo esercizio, va bene lasciare tutto deselezionato.



3. Dopo aver cliccato “Create repository” in fondo alla pagina, il repository è creato. Il repository è attualmente vuoto.

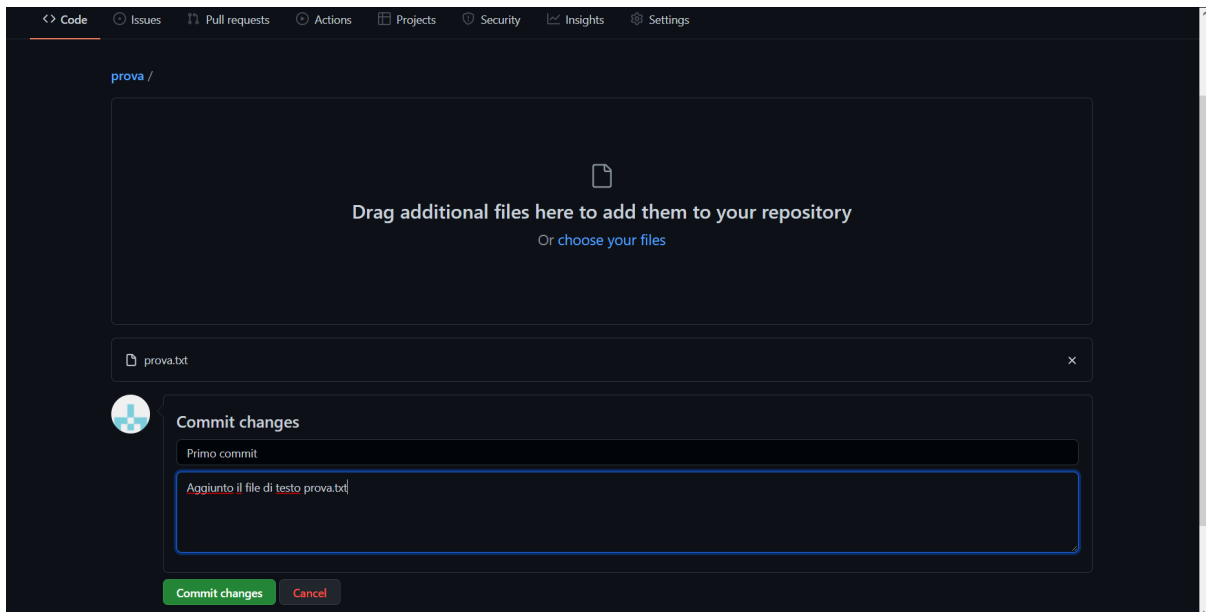
Commit and push

1. Creare un file di testo (ad esempio “prova.txt”). All’interno di questo file di testo inserire una testo a piacere (ad esempio “Hello World!”).
2. Come è possibile vedere nella figura sottostante, GitHub propone diversi metodi per caricare file in un repository. Il più immediato da comprendere e utilizzare è il primo, ovvero utilizzare l’interfaccia di GitHub. Cliccare quindi “uploading an existing file”.



3. Nella schermata successiva, aggiungere quindi il file di testo creato nel punto 1. È importante sottolineare che ogni commit va accompagnato da un commento e, se possibile, una descrizione esaustiva. Questo serve, soprattutto in grandi progetti sui quali lavorano un gran numero di persone, a specificare esattamente chi, cosa,

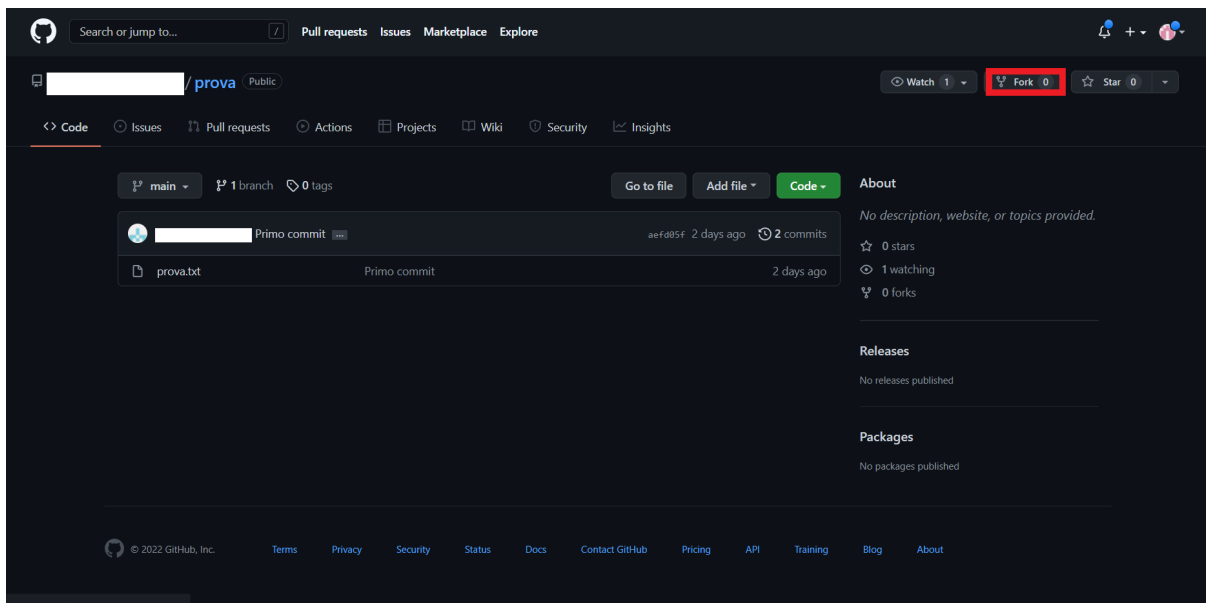
quando e perché una certa modifica è stata fatta. In un esercizio come questo, è sufficiente un commento come quello in figura.



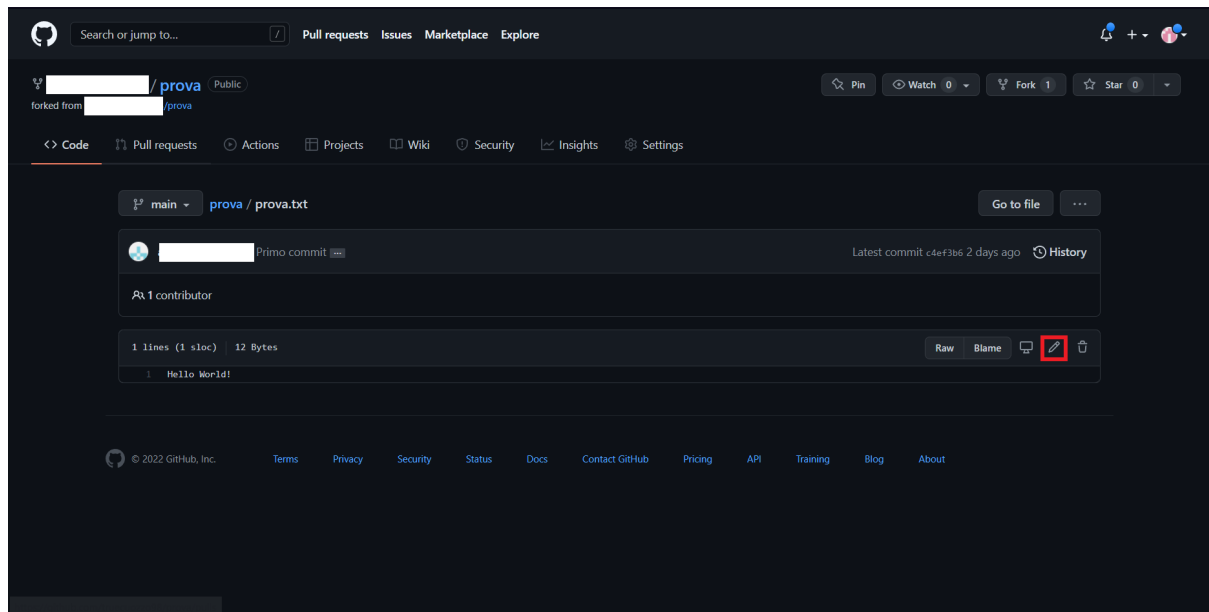
4. Al termine, cliccare “Commit changes”. Tramite interfaccia GitHub, questo equivale alla fase di *push* del *commit* nel *repository* (letteralmente, “spinta” delle “modifiche” nel “progetto”).

Istruzioni (Utente 2)

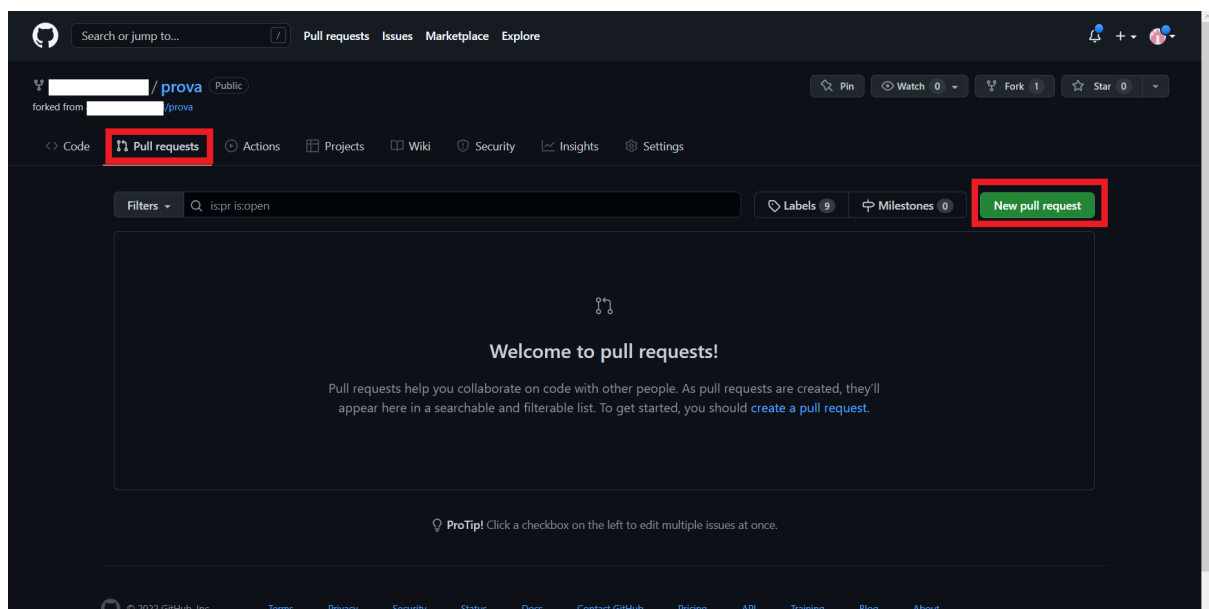
1. Collegarsi al link del repository creato dall'utente 1. Il link è nella forma `github.com/nickname utente 1/nome repository`
2. Cliccare in alto a destra su “Fork” e successivamente scegliere il proprio account. Una fork è una copia di un repository. In questo modo, è possibile effettuare liberamente modifiche senza danneggiare il progetto originale.



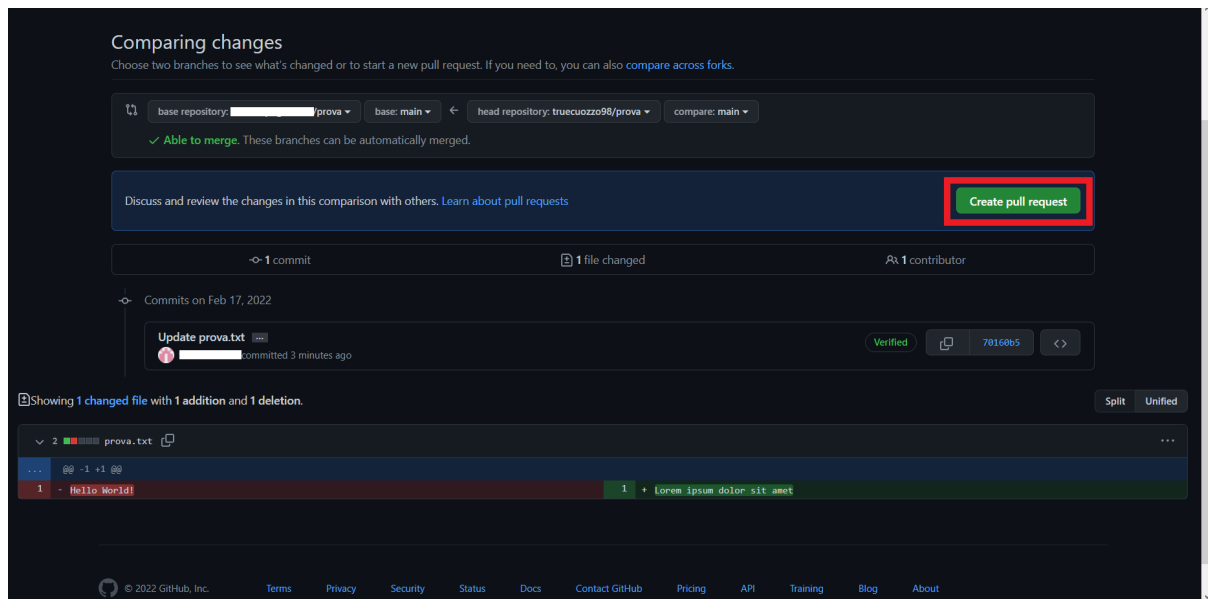
3. Come test, provare a modificare il file “prova.txt”, cliccandoci sopra e successivamente cliccare l'icona della matita.



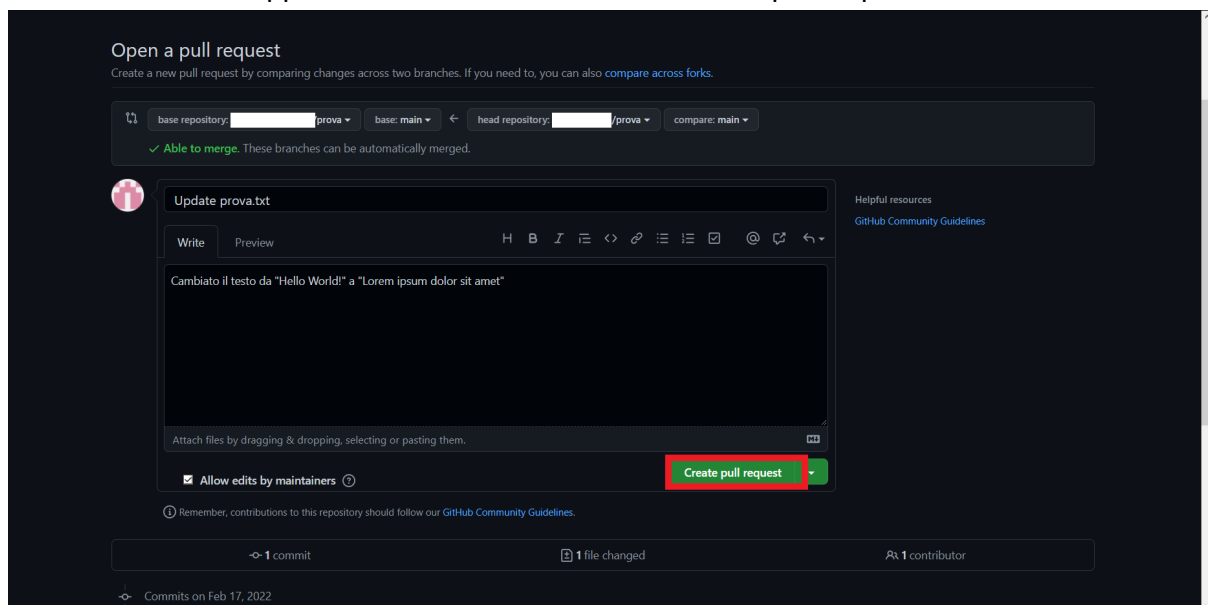
4. Inserire una qualsiasi frase a discrezione dell'utente. Una volta terminato, come spiegato nelle istruzioni per l'utente 1, occorre scegliere un messaggio significativo della modifica (*commit*), come ad esempio “update prova.txt” e scegliere di fare il *commit* direttamente nel *main branch* (“ramo principale”).
5. Cliccare sulla tab “Pull request” e successivamente “New pull request”. Questo servirà ad aggiornare il file creato dall'utente 1 con le modifiche effettuate dall'utente 2.



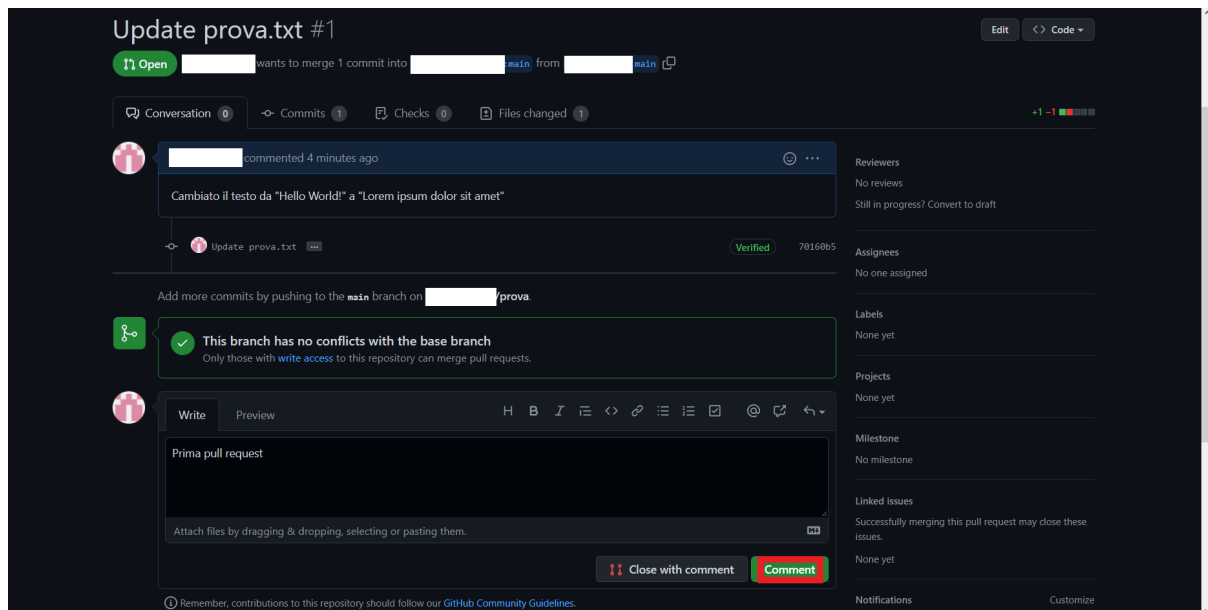
6. Nella schermata successiva, viene fatto un resoconto non solo di quali file sono stati modificati, ma anche in che punto del file e le effettive modifiche. Dopo aver revisionato i cambiamenti, cliccare su “Create pull request”.



7. Nella schermata successiva viene ripresentato il *commit* creato prima. Modificare se necessario oppure cliccare nuovamente su “Create pull request”.



8. GitHub controlla automaticamente che questa richiesta di *pull* non abbia conflitti con il *branch* base. In questo modo, chiunque abbia accesso al progetto, può collaborare effettuando queste richieste di *pull*, modificando i file che ritiene necessari senza danneggiare il progetto originale. Se necessario, scrivere un commento e cliccare su “Comment”.



Risultati attesi

Gli utenti hanno imparato e sanno utilizzare i meccanismi di base di Git, ovvero creare un *repository* e le funzioni *pull*, *commit* e *push*. Collaborando tra loro, i due utenti hanno appreso, seppur tramite una demo di gran lunga semplificata, le dinamiche che accadono in un grande progetto *open source*.

Obiettivi formativi

Opposto al software proprietario, l'idea alla base del Software Libero è quella di concedere formalmente diritti all'utente invece di toglierli. Le cosiddette libertà fondamentali del Software Libero sono: uso per qualunque scopo, studio del funzionamento anche attraverso il codice sorgente fornito sempre, redistribuzione *as is* (senza modifiche) a chiunque, redistribuzione dei lavori derivati (modificati) usando la stessa licenza. Gli utenti, familiarizzando con i meccanismi di base di Git, possono in futuro acquisire una conoscenza più profonda che permetta loro di partecipare attivamente alla condivisione, manutenzione e modifica di Software Libero.

Autore

Antonio Pugliese

Licenza

CC-BY-SA