Git

Tuesday, 28 March 2023 09:08

Allora, il prof non ha spiegato benissimo git, peccato che noi dobbiamo usare git per poter consegnare il progetto, allora ci penso io.
Si lo so, la consegna è per domani, avrei potuto creare questo documento questo weekend? Si
Ho procrastinato? Si, molto

1) Cos'è git
   Praticamente, immaginate il vostro drive online dove potete inserire ogni tipologia di file dentro
   Questo è git, solo con qualche funzione in più:
   - Poter lavorare contemporaneamente con le persone
   - Poter tornare indietro nel tempo
   - Estremamente più ordinato

   Questo è git, non più non meno, se volete essere fancy potete dire che è un "version control application".
   Una repository non è altro che una cartella del vostro drive.

2) Come usare git
   Allora, il prof ci ha detto di usare il git che è integrato a quella, bellissima applicazione che è tanto veloce quanto il mio bubble sort scritto male.
   Però, personalmente non sono riuscito a farlo funzionare; mi continuava a dare errore nell'autorizzazione.
   Se volete usare il git integrato in quell'applicazione, questo documento non è per voi.
   Io vi spiegherò come usare git attraverso:
   - Github Desktop (Semplice)
   - CLI (Per veri programmatori)

   Bisogna però prima comprendere quali sono i comandi di git.

3) Comandi
   Vi spiegherò i comandi base di git
   a. Clone
      Con questo noi scarichiamo una repository online

   b. Add

Quando noi vogliamo aggiornare dei file che sono online, noi dobbiamo dire a git quali file aggiornare
E per dire a git "Ehy, io ho modificato questo file, dopo lo voglio aggiornare" noi usiamo il comando add
Es.
Abbiamo creato/aggiornato/rimosso il file pippo.txt
Noi allora facciamo un "git add pippo.txt", così facendo noi diciamo a git che abbiamo creato/aggiornato/rimosso quel determinato file.

c. Commit
Ora che abbiamo detto a git che abbiamo apportato delle modifiche ad 1 o più file, noi vogliamo salvare queste modifiche.
Con il comando commit noi diciamo a git di salvare tutte le modifiche che abbiamo apportato ai file in locale.
Quando noi facciamo un commit, bisogna sempre aggiungerci una descrizione che riassuma le modifiche che abbiamo fatto.
Es:
- Abbiamo creato il file "ciao.txt", la descrizione potrebbe essere "Creato file ciao.txt"
- Abbiamo aggiunto una funzione alla classe "main.java", la descrizione potrebbe essere "Aggiunta funzione"
- Abbiamo fixato un bug nel nostro programma, allora la descrizione potrebbe essere "Fixato bug"

Ps. Quando dicevo "Poter tornare indietro nel tempo", io intendevo che git ci permette di tornare indietro a qualunque commit passato che abbiamo fatto.

d. Push
Prima ho detto che abbiamo salvato i file in locale, noi però vogliamo che queste modifiche siano anche online.
Affinchè questo succeda noi dobbiamo fare un push dei nostri commit locali nella nostra repository.
Una volta fatto il push la nostra repository online verrà aggiornata con le modifiche a noi fatte

e. Pull
Facendo un pull noi aggiorniamo i nostri file locali con i file che sono nella nostra repository.

4) Pratica
Bene, detto questo ora abbiamo tutte le conoscenze per potere gestire

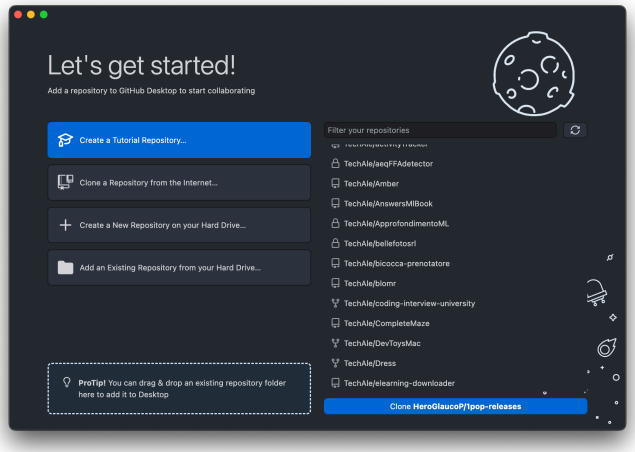
una repository semplice.
Come detto prima, noi possiamo usare 2 strumenti:

- Github Desktop
- CLI

Se volete la easy route, andate al punto 5
Personalmente suggerisco di imparare la CLI però, nel caso voi abbiate poco tempo e volete semplicemente consegnare questo compito il più velocemente possibile, si và per github desktop.

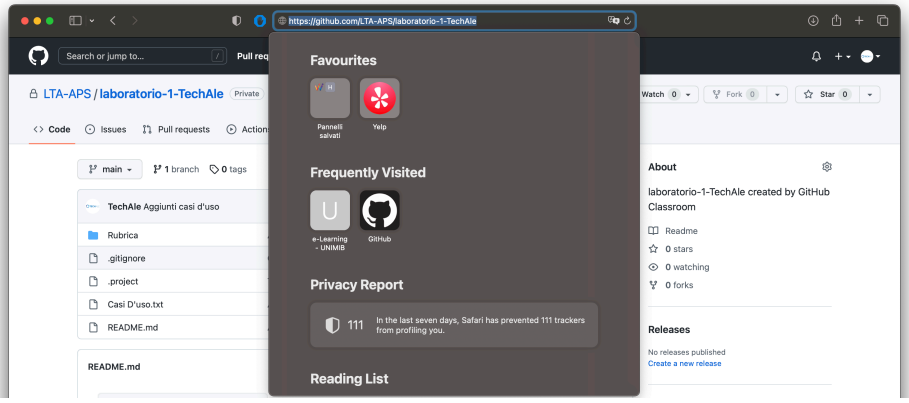
E si, esiste anche l'opzione di fare un semplice "drag and drop" da github, però questo ve lo consiglio; piuttosto usare github desktop, perché? Così, per quando dovremo fare il progetto potrete fare i pull senza dover scaricare ogni volta la repository.

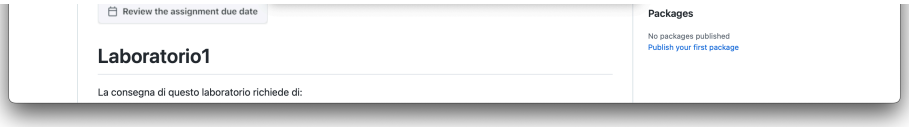
5) Github Desktop
Questa è una applicazione che ci permette di usare git molto facilmente.
Prima di tutto lo dovete scaricare: https://desktop.github.com
Una volta scaricato ed aperto, ci troveremo davanti questa interfaccia



Da qui:

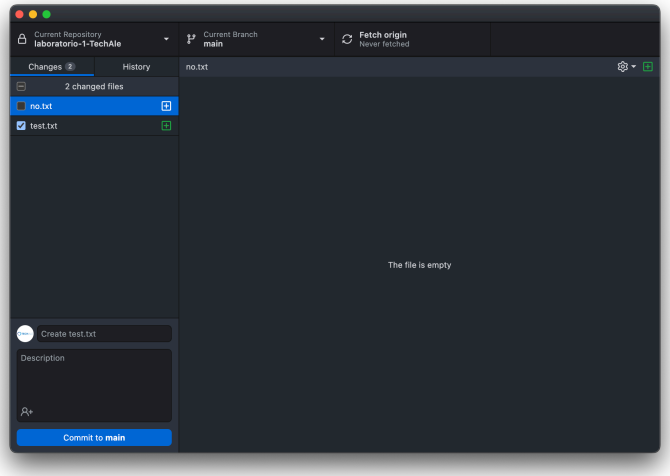
- clicchiamo su "Clone a repository from the internet"
- Andiamo su "URL" e poi dobbiamo incollare l'url della nostra repository





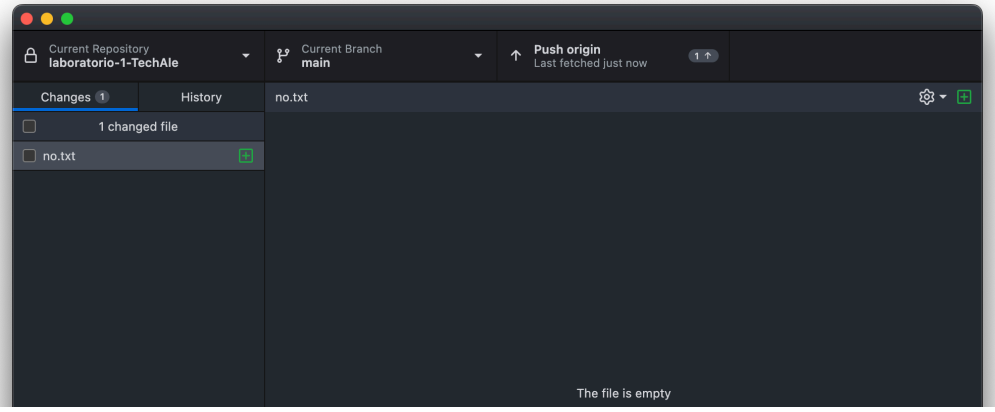
Ed infine "clone", facendo così abbiamo fatto il clone della nostra repository.
Vi chiederà di fare il login, fatelo assicurandovi di usare la vostra email universitaria

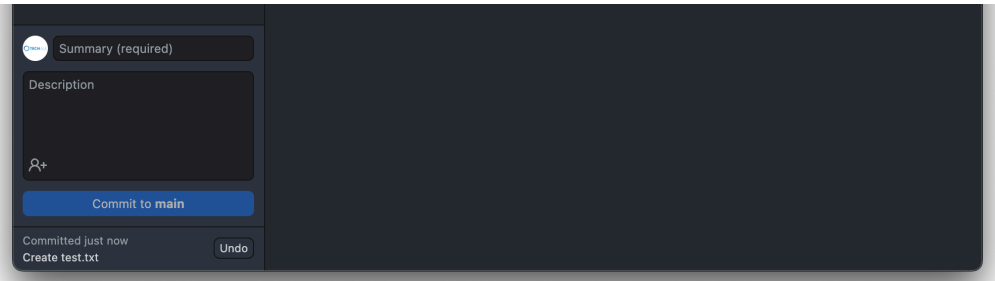
- Ora facciamo il lavoro dentro alla cartella che abbiamo creato (Facendo tasto destro su "current repository" possiamo aprire la cartella nel file explorer)
  Ed una volta finito il compito dobbiamo fare il git add ed il git commit.
- Il git add lo si fa cliccando sui vari checkbox



In questo caso sto dicendo a github desktop di fare il git add solo di text.txt

- Una volta selezionato/Deselezionato, in basso a sinistra diamo un titolo, e se volete anche una descrizione seppur sia opzionale.
- Una volta fatto il commit, possiamo fare un push cliccando in alto a destra
  "push origin"





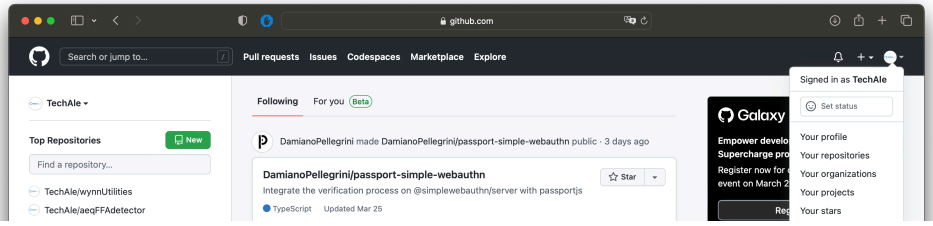
- Ora nella vostra repository i file si sono aggiornati, ed avete consegnato il compito!

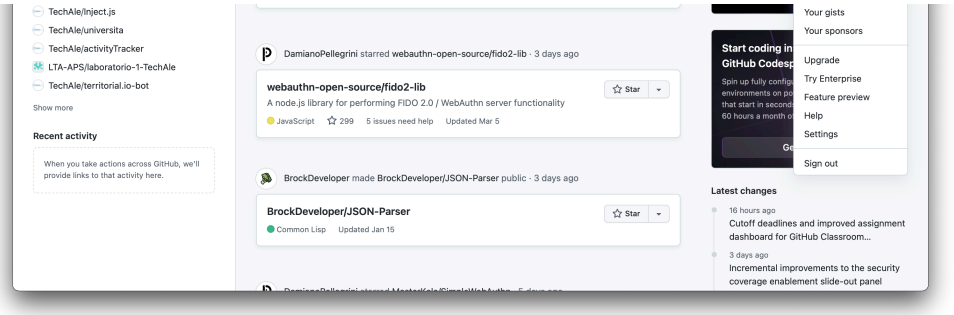
6) CLI

Quindi volete essere fancy, eh?
Da CLI non ci guadagniamo nulla, siccome ciò che si può fare da github desktop si può fare anche da qui, e ciò che si può fare qui si può fare anche li, quindi semplicemente è un flex.
Si, ci sono funzioni e siamo più veloci e bla bla, però in pratica chi crede che git da cli è migliore rispetto a github desktop stà solamente cercando di trattenere le lacrime siccome è consapevole di aver perso 5+ ore della sua vita per qualcosa che sicuramente non userà.
Detto questo, eccovi divertitevi a studiare a memoria tutti questi comandi!

- Noi vogliamo scaricare la repository da online in locale
  Per farlo dobbiamo fare git clone https://github.com/LTA-APS/laboratorio-1-TechAle
  (Sostituite il mio link con il vostro)
  E qui ora sorge un problema! Dobbiamo fare il login!
  Esistono vari modi per autenticarsi:
  - Fare il login inserendo un codice [Solo per chi ha windows]
  - Fare il setup delle proprie chiavi
    E qui esistono 2 modi per fare il setup:
    - Seguire dei video tutorial che tuttora non comprendo cosa fanno e perché lo fanno
    - Generare una propria chiave ssh

    Ovviamente vi spiegherà come si genera una chiave ssh.
    Da github premere in alto sinistra e poi "settings"





Da settings andare in "Developer settings"
Poi aprire "Personal access token" ed andare in "Fine-grainted tokens"
Nella creazione del token ricordatevi di premere "All repositories"
E poi "Generate token"
Una volta generato, una luunga stringa vi apparirà davanti, copiatela.
Questa vi servirà per il login:

- Quando vi verrà chiesto il vostro username mettete il nome del vostri github (Nell'immagine di prima, il vostro nome è la parola in grassetto dopo "Signed in as"
- Come password incollate quella chiave

Avete fatto il login! Forse. Pregate che funzioni, siccome a volte dà problemi
E' da 4 anni che uso git e tuttora non so perché a volte dà problemi, e poco mi interessa siccome io non utilizzo mai la CLI praticamente.

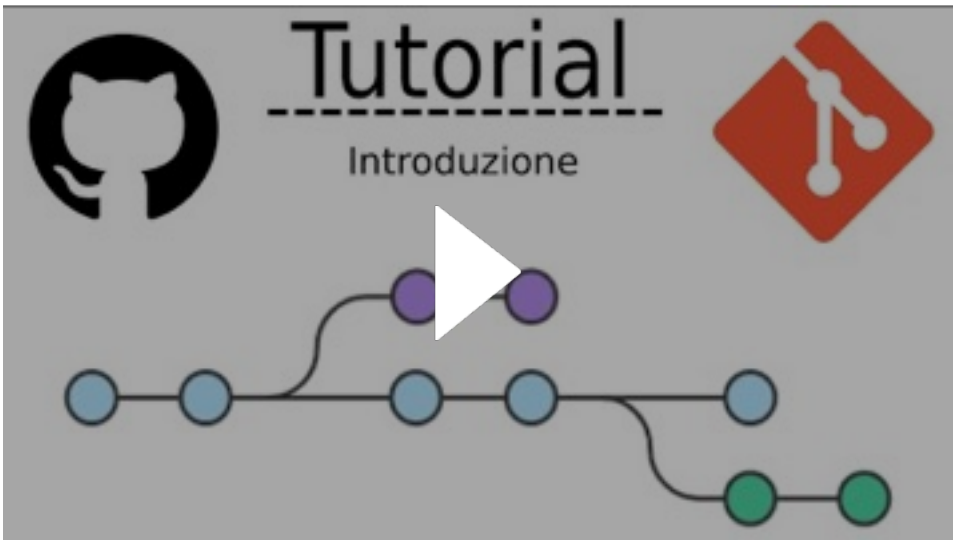
- Una volta fatto il clone, create i file che dovete create e, una volta finito il vostro lavoro
  Dovete fare "git add FILE"
  E per file intendo sia i singoli file, oppure anche le cartelle.

- Commit
  Una volta aggiunti tutti i file fate "git commit -m "INSERIRE UNA DESCRIZIONE CHE E' OBBLIGATORIA"

- Push
  Allora, qui è un po' un casino.
  Prima dobbiamo reperire il branch in cui noi siamo, e per farlo digitare "git branch -v" e, il nome affianco all'asterisco è il nostro branch.
  Poi dobbiamo reperire il link remoto della nostra repository.

Per reperirlo fare "git remote -v", e voi dovete prendere il nome affianco al link della vostra repository
Nel caso non vi compare nessun testo, allora dovete fare "git remote add origin LINK_REPOSITORY"
Ora possiamo finalmente fare "git push REMOTE_NAME BRANCH_NAME"

Detto questo, vi potrebbero apparire comunque degli errori se usate la CLI siccome, l'utilizzo di essa diciamo che richiede il presupposto che l'utente sappia tutti i comandi git e come agire in caso di eccezioni.
In questo caso, non mi metto a fare un tutorial completo di git qui, se volete potete guardare o dei video tutorial online, oppure (Momento pubblicità) potete guardare il video tutorial che ho fatto su git 1/2 anni fa: [IT] Git Tutorial



Warning: vi consiglio di mettere la velocità 0.75.