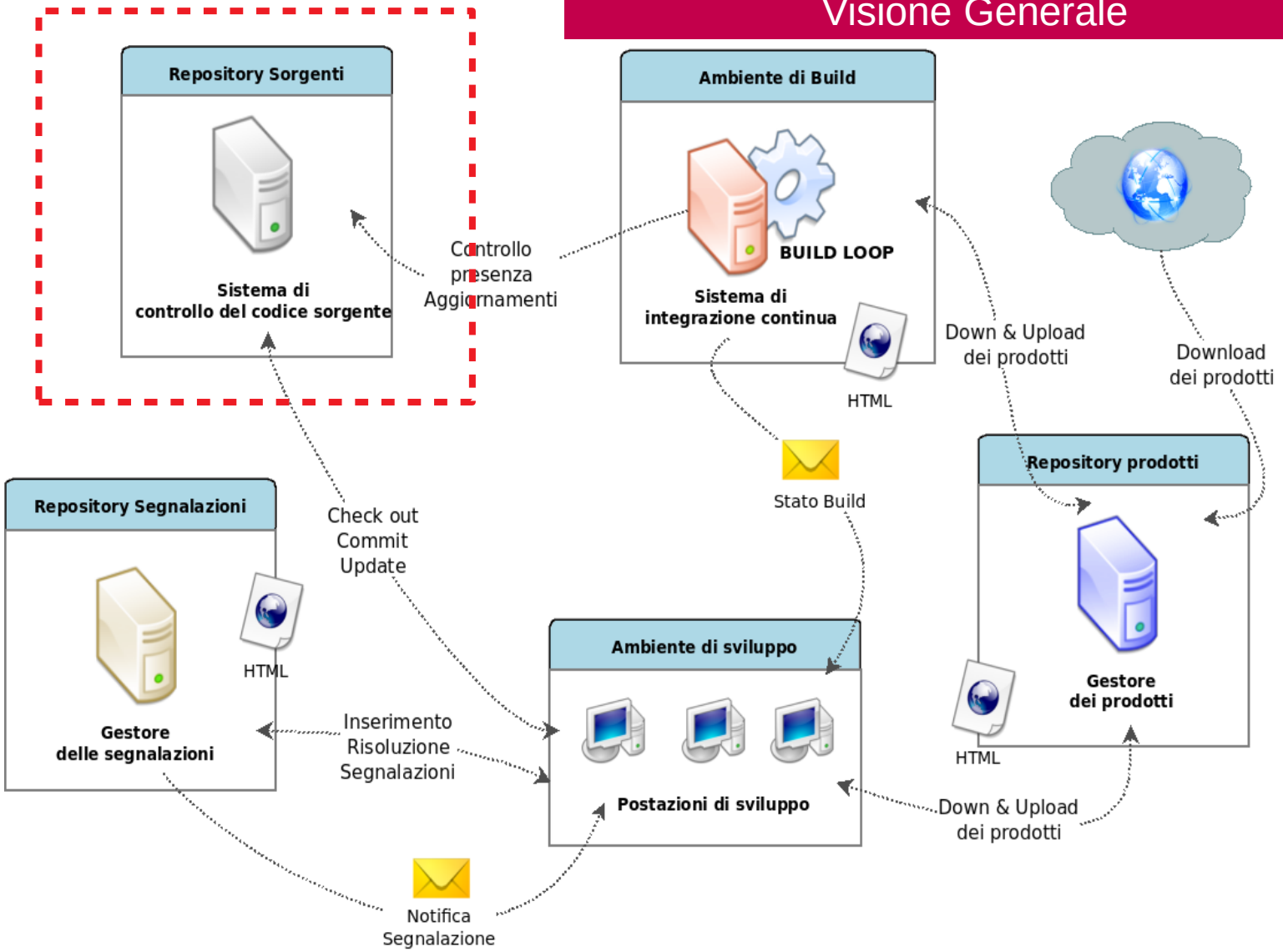
METODI E TECNOLOGIE PER LO SVILUPPO SOFTWARE

Nicola Bertazzo
nicola.bertazzo [at] unipd.it

Università degli Studi di Padova
Dipartimento di Matematica
Corso di Laurea in Informatica, A.A. 2022 – 2023

Visione Generale



Definizione

Git è un **software di controllo versione distribuito** utilizzabile da interfaccia a riga di comando, creato da **Linus Torvalds** nel **2005**.

La sua progettazione si ispirò a strumenti (allora proprietari) analoghi come BitKeeper e Monotone.

Git (che nello slang americano significa idiota) nacque per essere un semplice strumento per facilitare lo **sviluppo del kernel Linux** ed è diventato uno degli strumenti di controllo versione più diffusi.

Caratteristiche (1/3)

Branching and Merging:

- E' incentivato lo sviluppo su Branch diversi (locali o condivisi)

Piccolo e veloce:

- la maggior parte delle operazioni viene fatta in locale
- E' sviluppato in C. La velocità e la performance sono stati i requisiti primari

Distribuito (per "clone" del repository):

- Backups multipli
- Possibilità di adottare diversi Work flow

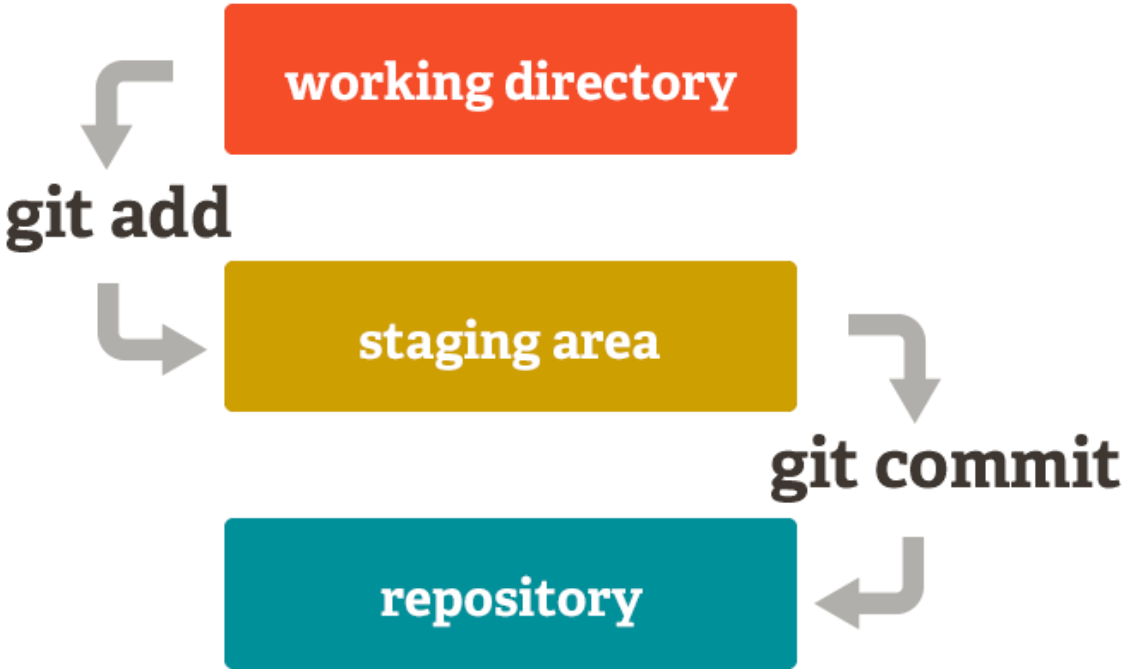
Integrità:

- Ogni commit è identificato da un ID (checksum **SHA-1** di **40-caratteri** basato sul contenuto di file o della struttura della directory) che ne garantisce l'integrità
- Non è possibile cambiare un commit senza modificare l'ID del commit stesso e di i commit successivi

Caratteristiche (2/3)

Staging Area:

- È stata aggiunta un'area di staging dove vengono validati i file modificati che potranno essere versionati con un commit



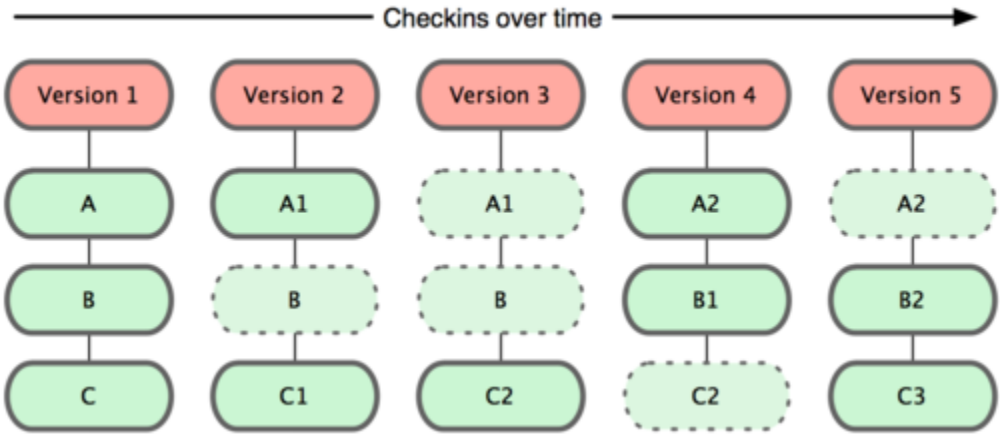
Caratteristiche (3/3)

Free and Open source:

- E’ rilasciato con licenza GNU General Public License version 2.0
- Il codice sorgente è pubblico https://github.com/git/git

Istantanee, non Diff

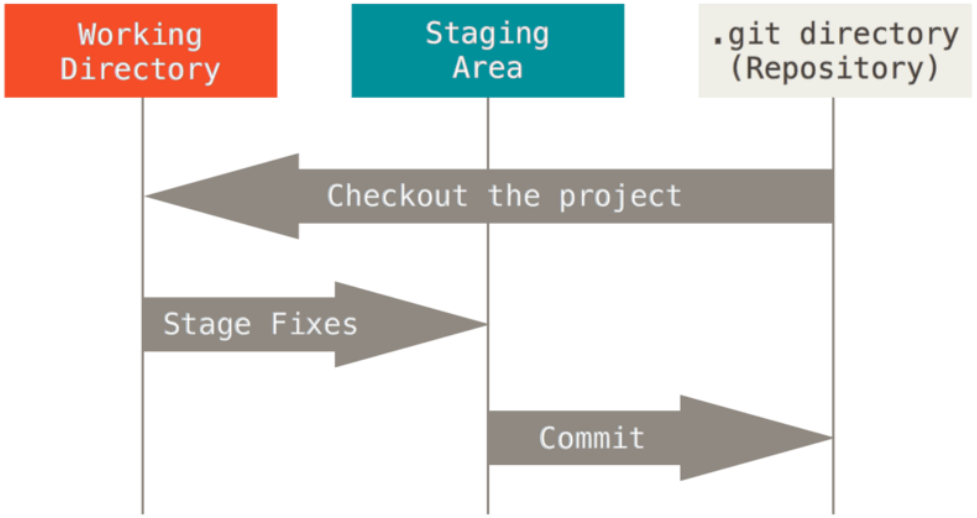
"… Git thinks of its data more like a **set of snapshots** of a miniature filesystem. Every time you **commit**, or **save the state of your project** in Git, it basically **takes a picture of what all your files look like at that moment** and stores a reference to that snapshot. To be efficient, **if files have not changed**, Git doesn't store the file again, just **a link to the previous identical file it has already stored**. Git thinks about its data more like a stream of snapshots."



Aree locali

In GIT i file della copia locale possono essere:

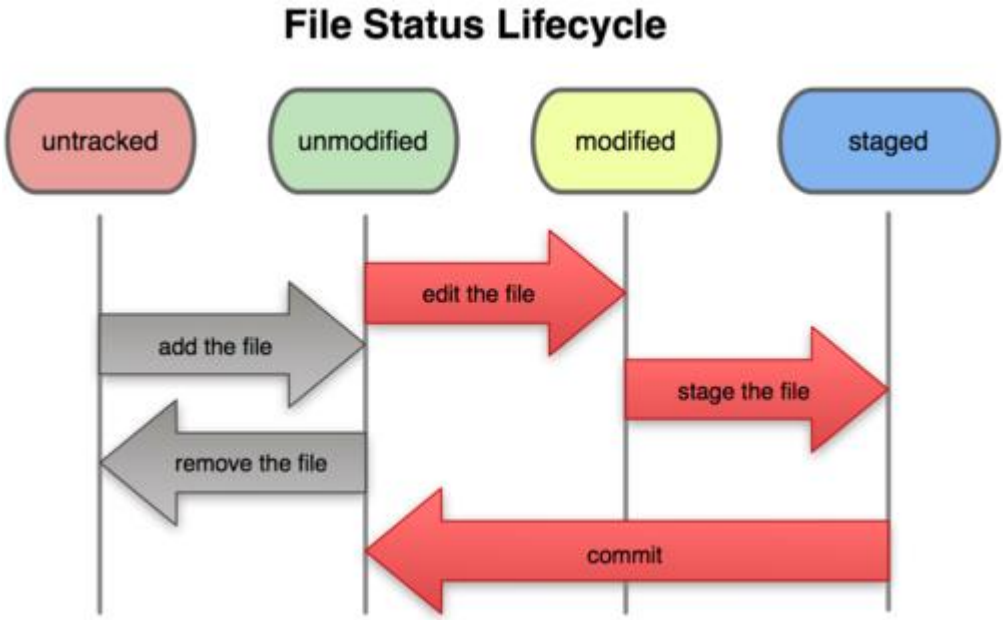
- Nella **Working directory**, *Checked out*, modificati ma non ancora validati (Modified)
- Nella **Staging Area**, validati ma non ancora committati. Il commit salva uno snapshot di tutti i file presenti nella staging area (Staged)
- Nel **Repository locale** (Committed)



Stato di un file in GIT

Un file in GIT può essere in uno dei seguenti 3 stati:

- **Modifed:** modificato nella working directory
- **Staged:** salvata una snapshot nella Staging Area
- **Commited:** preso dalla Staging Area e salvato nel repository locale



Installazione di GIT

Il manuale di installazione del client GIT è persente a questo indirizzo:
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

E' disponibile per tutti i sistemi operativi

I client grafici più utilizzati sono:
https://www.sourcetreeapp.com
https://www.gitkraken.com
Oppure i client integrati su IDE

Configurazione di GIT

E' richiesta una configurazione iniziale dove vengono impostati l'username e l'email da usare per ogni commit:

git config --global user.name "Bugs Bunny"

git config --global user.email bugs@gmail.com

E' possibile invocare il seguente comando per avere la lista di tutte le configurazioni

git config --list

Le configurazioni possono essere fatte a vari livelli:

- **system**: per l'intero sistema per tutti gli utenti
- **global**: per il singolo utente
- **local** (di default): per singolo repository

https://git-scm.com/docs/git-config

Creazione del repository

Due scenari:

Creazione del repository locale nella cartella corrente:

git init

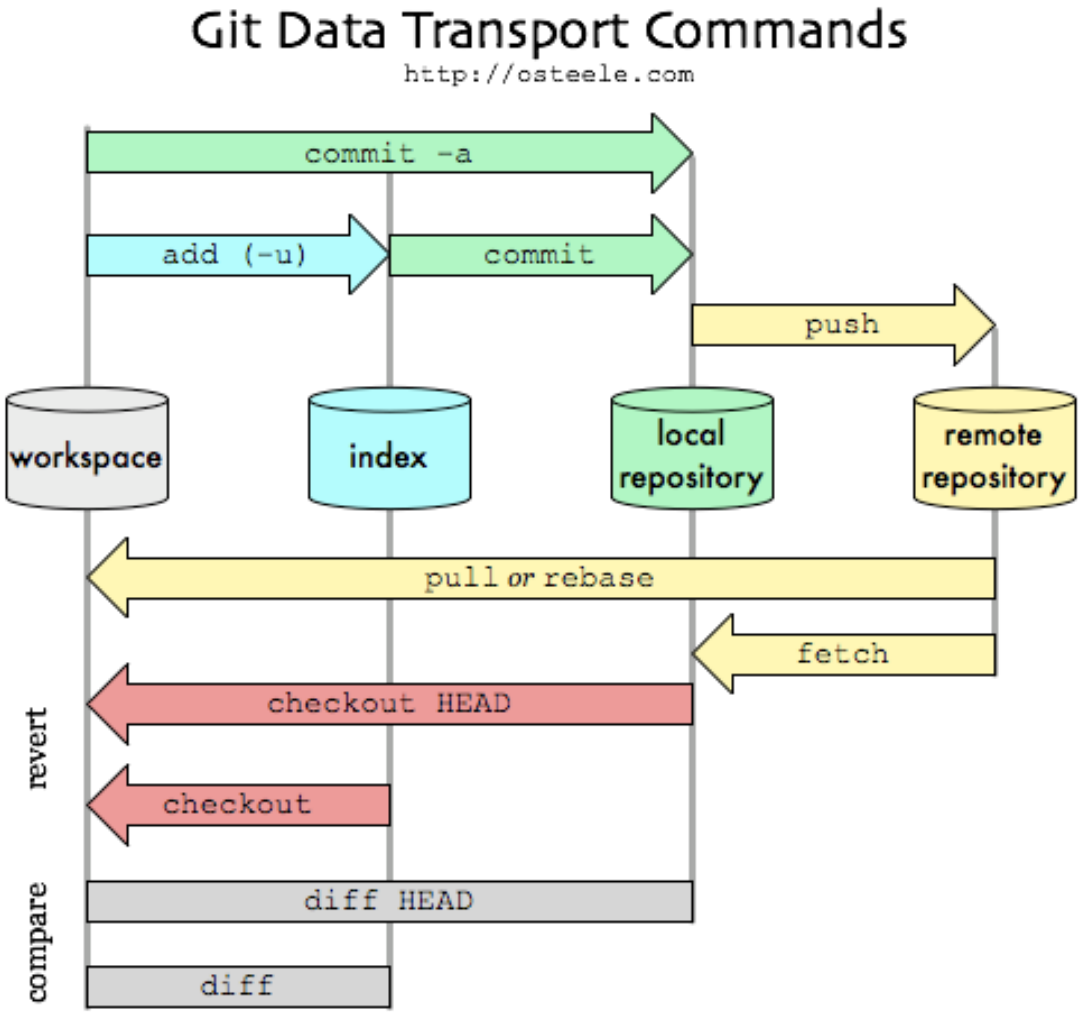
- Crea una cartella .git nella cartella corrente
- Da questo momento è possibile iniziare il versionamento dei file localmente

Clonazione di un repository remoto nella cartella corrente:

git clone url localDirectoryName

- Clona il contenuto del repository remoto nella cartella corrente
- Crea una cartella .git che rappresenta il repository locale

Riassunto Comandi Base



Add and Commit

Aggiungere i file nella staging area e creazione di una Snapshot

git add Hello.java Goodbye.java

Commit e salvataggio di una nuova versione nel repository locale

git commit –m "Fixing bug #22"

Rimuovere un file dalla staging area senza perdere le modifiche

git reset HEAD filename

Rimuovere un file dalla staging area perdendo le modifiche

git checkout -- filename

Stato e ripristino modifiche

Vedere lo **stato** dei file workspace o nella staging area:

git status

o

git status –s (short version)

Per vedere cos'è stato **modificato** ma non ancora validato nella staging area:

git diff

Per vedere cos'è stato **modificato nella staging area**:

git diff –cached

Per vedere la **lista dei cambiamenti** (commit) nel repository locale:

git log

Per vedere le ultime 2 modifiche:

git log -2

Branch e merging

Per creare un nuovo branch

git branch name

Per avere la lista dei branch locali:

git branch

Per passare in un branch

git checkout branchname

Per effettuare attività di merge di un branch nel master

git checkout master

git merge branchname

Repository remoto

Lista dei repository remoti

git remote

o

git remote -v

Fetch: recupero i nuovi branch e le modifiche dal remoto senza aggiornare la working copy (senza fare merge):

git fetch origin

Pull: recupero le ultime modifiche dal repository remoto e aggiorno la working copy (fetch and merge):

git pull origin master

Push: Per inviare le modifiche al repository remoto:

git push origin master

Comandi e tutorial

La lista dei comandi base può essere recuperata da:
https://education.github.com/git-cheat-sheet-education.pdf
https://git-scm.com/docs/giteveryday
https://git-scm.com/docs/user-manual.html

Sono presenti dei tutorial per provare i comandi di GIT:
https://git-scm.com/book/it/v2
https://git-scm.com/docs/gittutorial

Gioco per imparare ad utilizzare git:
https://github.com/git-learning-game/oh-my-git

Riferimenti

https://it.wikipedia.org/wiki/Git_(software)
https://github.com/torvalds/linux
https://www.slideshare.net/valix85/introduzione-a-git-ita-2017
https://courses.cs.washington.edu/courses/cse403/13au/lectures/git.ppt.pdf
https://git.wiki.kernel.org/index.php/LinusTalk200705Transcript
https://git-scm.com/about
https://git-scm.com/book/en/v2