



SAPIENZA
UNIVERSITÀ DI ROMA

”Sapienza” Università di Roma
Ingegneria dell’Informazione, Informatica e Statistica
Dipartimento di Informatica

Programmazione WEB

Autore
Vincenzo Bova

A.A. 2025/2026

Indice

1	Introduzione a Git	2
1.1	Sistemi di versionamento	2
1.2	Git	3
1.2.1	Commit	3

1

Introduzione a Git

1.1 Sistemi di versionamento

Durante lo sviluppo di un progetto c'è spesso la necessità di effettuare revisioni, correzioni o modifiche ai file che lo compongono.

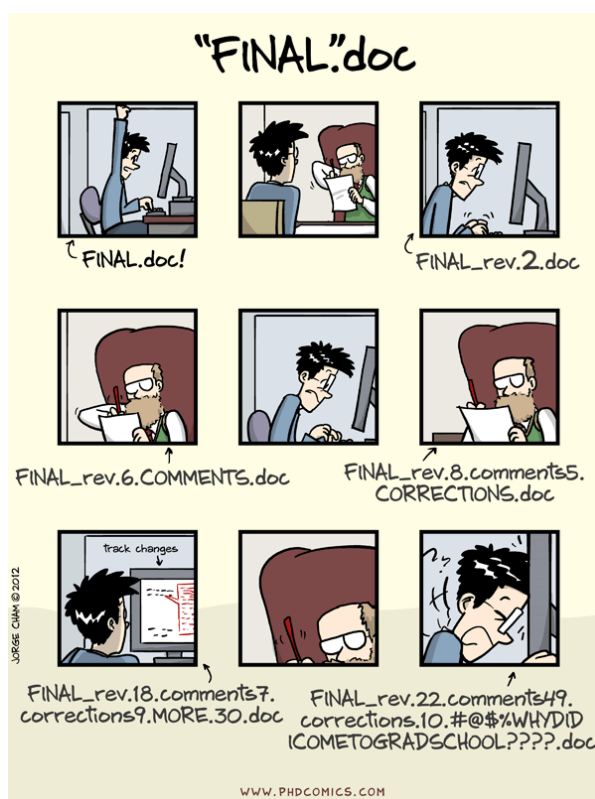


Figura 1: Sviluppo di un progetto mediante la creazione di nuovi file

Gestire ciò creando ogni volta nuovi file, tuttavia, comporta evidenti problemi:

- **Duplicazione del contenuto:** che rende il sistema inefficiente e aumenta la difficoltà nel mantenere integrità;
- **Assenza di Naming Convention:** che rende impossibile risalire ad uno storico delle modifiche;
- **Autori incerti;**
- ...

Per ovviare a ciò sono stati creati i **sistemi di versionamento** (git, csv, mercurial, svn...), i quali offrono vari benefici:

- **Gestione delle versioni:** il sistema si occupa automaticamente di etichettare le varie versioni in modo consistente;
- **Tracciamento delle modifiche:** è possibile accedere ad uno storico delle modifiche effettuate;
- **Presenza di metadati:** ogni modifica ha un autore, una data...;
- **Creazione di linee di sviluppo parallele:** è possibile creare una versione parallela del codice per non modificare la versione principale, e poi riunirle integrando i cambiamenti;
- **Sincronizzazione tra computer:** il sistema consente di mantenere il progetto allineato tra più computer.

1.2 Git

Git è un sistema di versionamento distribuito e veloce, creato nel 2005 e capace di gestire progetti di grandi dimensioni. Si basa su un design semplice e utilizza DAG (*Directed Acyclic Graph*) e Merkle trees come strutture dati.

Definizione 1.2.1 : Repository

È un insieme di commit, branch e tag.
Per semplicità assumiamo che un progetto equivale ad un repository.

Definizione 1.2.2 : Working copy

È un insieme di file tracciati nella copia locale.
Un nuovo file non sarà ancora tracciato e bisognerà aggiungerlo.
Quando aggiorniamo i file (*update*), stiamo aggiornando la working copy.

1.2.1 Commit

Un commit è un'istantanea del repository in un determinato momento.
Viene identificato dallo **SHA1** del commit stesso e contiene diversi campi:

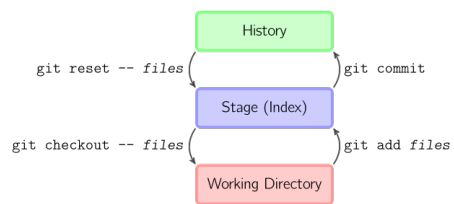
- data+autore, data+committer
- commento **obbligatorio**
- 0,1 o più genitori
- tree: hash di tutti i file nel commit

In particolare il commit può contenere un sottoinsieme delle modifiche (anche ad un singolo file), le quali devono essere aggiunte alla staging area dei cambiamenti.

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

(a) Serie di commit ad un progetto



(b) Serie di commit ad un progetto