Introduzione Version Control

Gilberto Conti

Fondazione ITS ICT

2025

Cofinanziato dall'Unione europea

REGIONE PIEMONTE

Agenda

Risorse per lo studio

- Documentazione ufficiale di git `https://git-scm.com/book/en/v2`
- Queste slide saranno caricate su `Moodle` come pdf
- Visualizing git
  `https://git-school.github.io/visualizing-git/`
- Molte risorse e anche parti delle slide saranno in inglese

Esame

- Domande di teoria sulle slide
- Realizzazione di un progetto a gruppi o individualmente, il progetto potrà essere svolto in classe durante le ore di lezione

Dare una versione

- Sviluppo collaborativo di progetti
- Storico delle modifiche
- Formalizzare dei punti chiave (alpha, beta, release, etc)

La galleria degli orrori

- Documento.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc
- Documento-definitivo-4.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc
- Documento-definitivo-4.doc
- Documento-definitivo-questavoltaveramentedaiviprego.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc
- Documento-definitivo-4.doc
- Documento-definitivo-questavoltaveramentedaiviprego.doc
- Documento-ricominciamo.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc
- Documento-definitivo-4.doc
- Documento-definitivo-questavoltaveramentedaiviprego.doc
- Documento-ricominciamo.doc
- Documento-ricominciamo-versionediclaudio.doc

La galleria degli orrori

- Documento.doc
- Documento-buono.doc
- Documento-buono-2.doc
- Documento-buono-3.doc
- Documento-buono-10.doc
- Documento-buonissimo.doc
- Documento-definitivo.doc
- Documento-definitivo-1.doc
- Documento-definitivo-4.doc
- Documento-definitivo-questavoltaveramentedaiviprego.doc
- Documento-ricominciamo.doc
- Documento-ricominciamo-versionediclaudio.doc
- Documento-orabasta.doc

Il Salvatore

Le motivazioni

Linus Torvalds

I'm an egotistical bastard, and I name all my projects after myself.
First **Linux**, now **Git**.

Le motivazioni

Linus Torvalds

I'm an egotistical bastard, and I name all my projects after myself.
First **Linux**, now **Git**.

- **Random three-letter combination that is pronounceable**, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.

Le motivazioni

Linus Torvalds

I'm an egotistical bastard, and I name all my projects after myself.
First **Linux**, now **Git**.

- **Random three-letter combination that is pronounceable**, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- **Stupid**. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.

Le motivazioni

Linus Torvalds

I'm an egotistical bastard, and I name all my projects after myself.
First **Linux**, now **Git**.

- **Random three-letter combination that is pronounceable**, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- **Stupid**. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.
- **Global information tracker**: you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.

Le motivazioni

Linus Torvalds

I'm an egotistical bastard, and I name all my projects after myself.
First **Linux**, now **Git**.

- **Random three-letter combination that is pronounceable**, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- **Stupid**. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.
- **Global information tracker**: you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- **Goddamn idiotic truckload of sh*t**: when it breaks.

Non solo Git

- **Subversion**, chiamato amichevolmente SVN. Una soluzione ancora ampiamente utilizzata da importanti progetti, ma ormai sempre piu' minoritaria. Utilizza una struttura centralizzata.
- **Mercurial**, simile ad SVN, usato da Google.
- **Concurrent Version System**, antesignano dei sistemi di vcs in rete.
- **Bit keeper**, antenato di Git.
- **Bazaar**, un tempo utilizzato da Ubuntu.
- Soluzioni commerciali: Amazon CodeCommit, Azure DevOps Server, etc

Alcune definizioni

Repository

L'archivio del codice, può essere in locale o remoto
(Nota: in italiano può essere sia maschile che femminile)

Push

Inviare delle modifiche al repository.

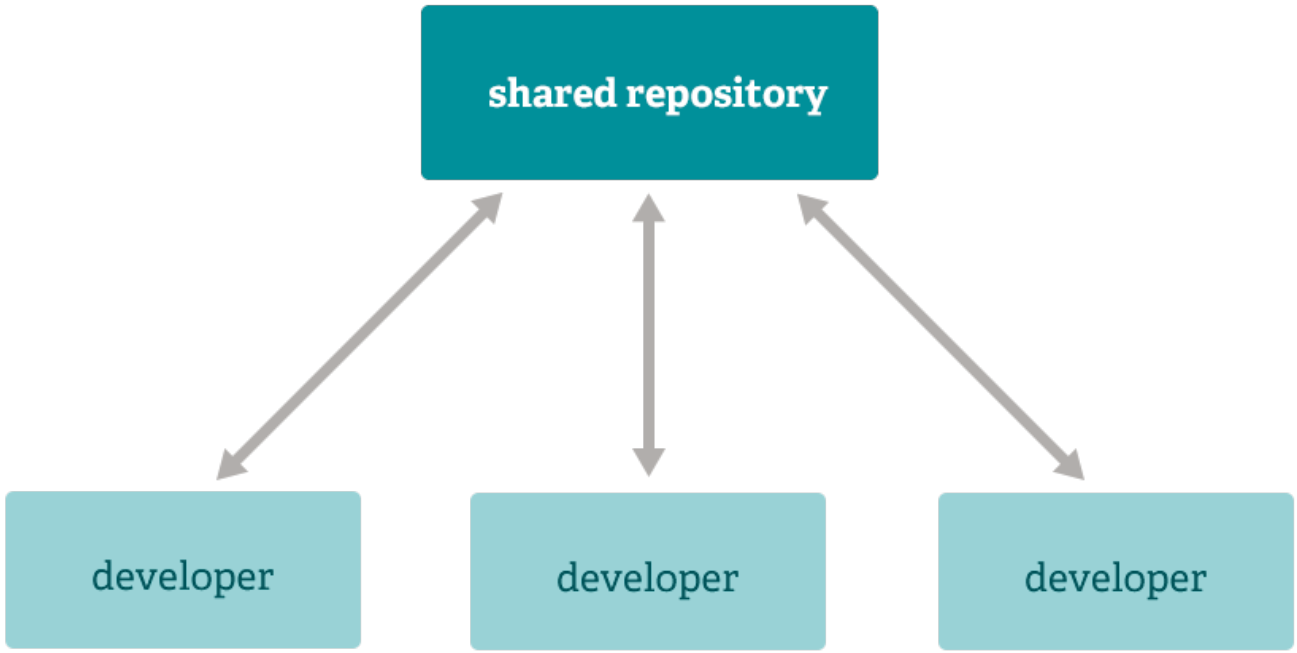
Fetch/Pull

Ricevere delle modifiche dal repository.

Caratteristiche

Centralizzato

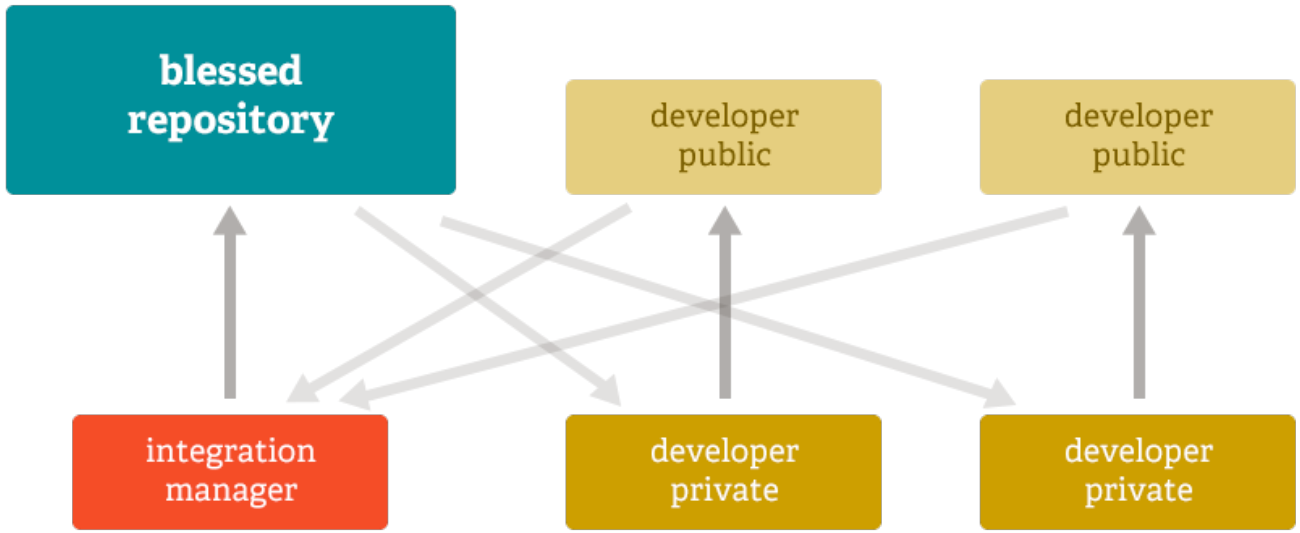
Una struttura centralizzata permette di **sincronizzare l'accesso** al repository. Concretamente impedisce che qualcuno modifichi il repository se nel mentre sono state fatte delle modifiche da altri.



Caratteristiche

Distribuito

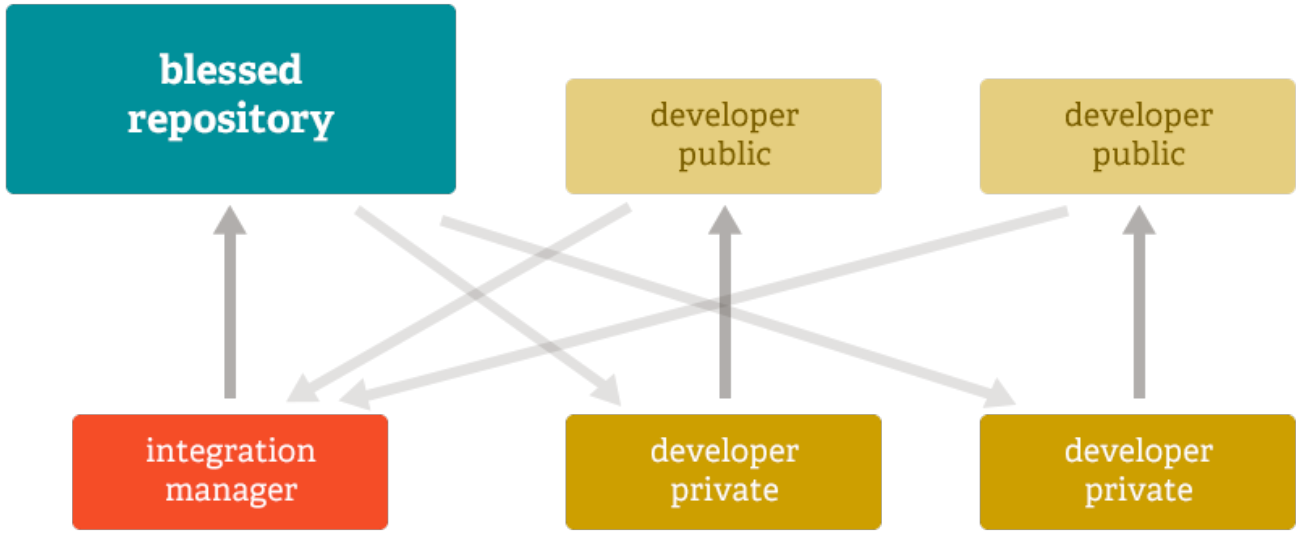
Diversi sviluppatori hanno la possibilità di creare una **copia completa** del repository ed operare tutti i cambiamenti che desiderano sulla loro copia. Quando lo ritengono necessario possono chiedere al proprietario del repository principale di **importare i loro cambiamenti**.



Caratteristiche

Branching Model

La caratteristiche che distingue **git** da altri sistemi di versionamento è il suo `branching model`.
Git permette ed incoraggia la creazione di diverse versioni del codice che possono essere **indipendenti tra loro**. Per navigare, creare ed eliminare queste versioni bastano **pochi secondi**.



Caratteristiche

Open Source

Software distribuito sotto i termini di una licenza che ne concede lo studio, l'utilizzo, la modifica e la redistribuzione.

Cose da fare oggi

- Installare git
- Configurare git per il proprio utente
- (Facoltativo) Verificare che l'editor di testo utilizzato supporti git.

Windows

- Potete scaricarlo all'indirizzo:
  `https://git-scm.com/download/win`
- Visitate https://gitforwindows.org/

Mac

Probabilmente è già installato se usate Linux. Potete verificare l'installazione aprendo un terminale:

```
$ git --version
```

Altrimenti potete scaricarlo all'indirizzo:
`https://git-scm.com/download/mac`

Linux

Probabilmente è già installato se usate Linux. Potete verificare l'installazione aprendo un terminale:

```
$ git --version
```

Installazione su **Fedora/Centos**

```
$ dnf install git-all
```

Installazione su **Debian/Ubuntu/etc**

```
$ apt install git-all
```

Configurazione Editor

- **Visual Studio Code**: supportato nativamente
- **IntelliJ**: supportato nativamente
- **Vim/Neovim**: supportato nativamente, consiglio di installare `tpope/vim-fugitive` e `airblade/vim-gitgutter`
- **Emacs**: supportato nativamente con diversi plugin per estendere

Configurazioni

Git può essere completamente configurato grazie al comando che mette a disposizione: **git config**.
Non fa niente di più che modificare delle variabili in alcuni file di testo:

- `/etc/gitconfig`: contiene le variabili globali che vengono utilizzate da tutti gli utenti
- `~/.gitconfig` o `~/.config/git/config`: contiene valori specifici per il tuo utente
- `.git/config`: riferito al singolo progetto

Configurazioni obbligatorie

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

Creare un repository

Vogliamo incominciare a versionare una cartella in cui teniamo il nostro progetto, ci sono delle leggere differenze a seconda dei sistemi operativi che usate.
Linux:

```
$ cd /home/user/my_project
```

macOS:

```
$ cd /Users/user/my_project
```

Windows:

```
$ cd C:/Users/user/my_project
```

Creare un repository

Inizializzare il repository

```
$ git init
```

Questo comando crea un directory chiamata `.git` che contiene tutti i file necessari per far funzionare git.

Creare un repository

Inizializzare il repository

```
$ git init
```

Questo comando crea un directory chiamata `.git` che contiene tutti i file necessari per far funzionare git.

Creare un repository locale

Creare un repository git equivale a creare un sottodirectory chiamata `.git` all'interno della quale vengono mantenuti tutti i file necessari per il funzionamento di git.

Creare un repository

Inizializzare il repository

```
$ git init
```

Questo comando crea un directory chiamata `.git` che contiene tutti i file necessari per far funzionare git.

Creare un repository locale

Creare un repository git equivale a creare un sottodirectory chiamata `.git` all'interno della quale vengono mantenuti tutti i file necessari per il funzionamento di git.

Nel corso di queste lezioni proveremo anche a dare uno sguardo a questa directory per sfatare i miti sui suoi contenuti.

Importare un repository

Importare un repository remoto

Importare un repository da una risorsa di rete equivale a copiare la cartella `.git` all'interno di una directory locale.

Importare un repository

Importare un repository remoto

Importare un repository da una risorsa di rete equivale a copiare la cartella `.git` all'interno di una directory locale.

```
$ git clone https://github.com/libgit2/libgit2 mylibgit
$ git clone <indirizzo> <nome cartella>
```

Importare un repository

Importare un repository remoto

Importare un repository da una risorsa di rete equivale a copiare la cartella `.git` all'interno di una directory locale.

```
$ git clone https://github.com/libgit2/libgit2 mylibgit
$ git clone <indirizzo> <nome cartella>
```

Piattaforme Online

- **GitHub.com**: completamente proprietaria, ora di proprietà della Microsoft. Sicuramente la piattaforma più utilizzata al mondo.
- **GitLab.com**: nato come copia di GitHub, con una versione a pagamento, ma è possibile installarla sui propri server. Il codice è open source.
- **BitBucket.com**: completamente proprietaria, usata in ambiti di informatica industriale.
- Launchpad, SourceForge, Beanstalk, Apache Allura, Git Kraken, Gitea, Gogs, etc

GitLab

Perché utilizzeremo **GitLab**:

- Permette di creare **repository privati**
- Le funzionalità fondamentali sono gratuite
- Il codice è **open source**, quindi il rischio di chiusura a pagamento nel giro di pochi anni.
- Essendo open source potete installarlo ed utilizzarlo localmente senza dover dipendere dall'azienda che lo sviluppa e mantiene.

GitLab

Cosa farete adesso:

- Creare un account su **gitlab.com**
- Generare una chiave ssh
- Caricare una chiave ssh
- Configurare git per utilizzare la vostra chiave

Risorse:

- `https://gitlab.com/users/sign_up`
- `https://docs.gitlab.com/ee/user/ssh.html`

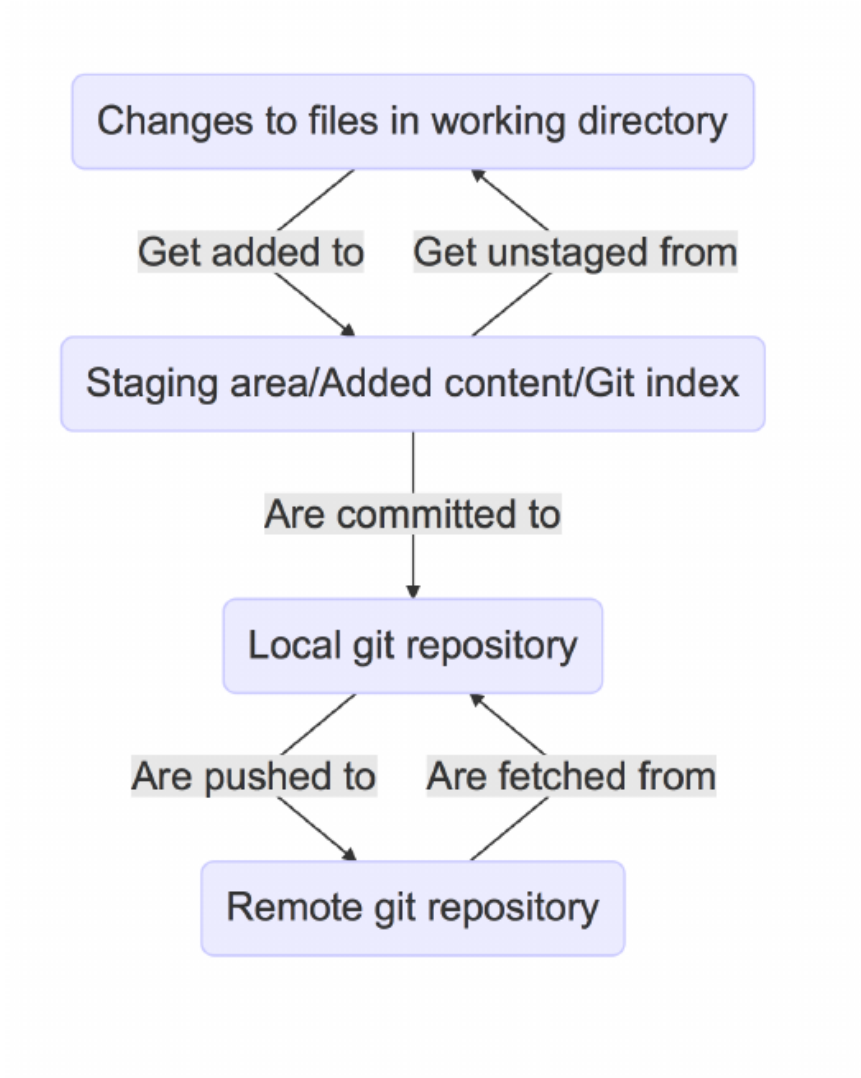
Monitorare lo stato

```
$ git status
```

Suggerimenti

Monitorare lo stato dovrebbe diventare un automatismo durante l'utilizzo di git e lo sviluppo. Molti editor e shell integrano lo stato in elementi grafici.

Le fasi del commit



The Four Phases of Git Content

Usare l'area di staging

```
$ git add <file>

$ git restore --staged <file>
```

Registrare un commit

```
$ git commit
```

Rimuovere/Modificare i file

```
$ git rm <file>

$ git mv <file> <destination>
```

1. Introduzione
2. Repository
3. Gestire i file
4. Branch
   - Guardare la storia
   - Navigare la storia
   - Creare/navigare i branch
   - Merge

Guardare la storia

```
git log
git log --graph --pretty=oneline --decorate
git show <commit id>
```

Guardare la storia

```
git checkout
```

Guardare la storia

```
git checkout -b <nome del nuovo branch>
git checkout <nome del branch esistente>
```

Guardare la storia

```
git merge <nome del branch>
```