



LABORATORIO DI ALGORITMI E STRUTTURE DATI

USO DI GIT

SLIDES PROF. POZZATO

USO DI GIT

- Sistema di versioning del software
 - Maturo
 - Ricco di funzionalità
- Breve tutoriale
 - Libro open source Pro Git (<https://git-scm.com/book/it/v2>)
- Accesso
 - <https://gitlab2.educ.di.unito.it>
- Credenziali del laboratorio


Login

Credenziali per
l'accesso alle
macchine del
laboratorio

Sign in · GitLab

gitlab2.educ.dl.unito.it/users/sign_in

GitLab for Labinfo

 UNIVERSITÀ
DEGLI STUDI
DI TORINO

L'utilizzo di questo servizio è vincolato alle attività proprie del Corso di Laurea in Informatica

Lo spazio occupato da ciascun utente è monitorato ed eventuali abusi saranno sanzionati.

Seleziona dal menu a lato **LDAP** ed usa le **credenziali di laboratorio** per accedere a questo servizio

LDAP Standard

LDAP Username

drago

Password

.....

☐ Remember me

Sign in

INSTALLAZIONE

- Mac: una versione di git ed una di ssh dovrebbero essere già installate;
- Linux: è necessario utilizzare il package manager della propria distribuzione per installare git (ssh in genere è già installato e configurato).

Ubuntu/Debian-like: **`sudo apt-get install git-all`**

- Windows: scaricare e installare git-for-windows

<https://git-scm.com/downloads>

- Si consiglia di lavorare nella shell bash creata dall'installer («Git Bash»).
- Per maggiori informazioni si veda la sezione 1.5 del libro Pro Git

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

CONFIGURAZIONE GIT

- Già installato sulle macchine del laboratorio
- Necessario fornire a Git i dati necessari per tracciare i commit
- Da terminale, ipotizzando che l'utente sia Mario Rossi con email mario.rossi@unito.it

```
git config --global user.name "Mario Rossi"
```

```
git config --global user.email mario.rossi@di.unito.it
```

- Attivazione del supporto alla visualizzazione a colori dei risultati dei comandi git

```
git config --global ui.color true
```

CONFIGURAZIONE GITLAB

- Creare copie locali dei repository presenti su GitLab in modo che ciascun componente del gruppo possa contribuire al progetto
- Necessità di configurare la propria macchina in modo che possa accedere in modo sicuro a GitLab
- Due possibilità:
 - ssh — laboratorio Dijkstra (e vostre macchine)
 - https — laboratori windows (Von Neumann e Turing)

CONFIGURAZIONE GITLAB (SSH)

- Generazione coppia di chiavi crittografiche
- Inserimento **chiave pubblica** tra quelle accettate dal vostro utente su GitLab

GENERAZIONE CHIAVI

<https://gitlab2.educ.di.unito.it/help/ssh/index.md>

- Controllare che il vostro sistema non abbia già una chiave SSH

```
cat ~/.ssh/id_rsa.pub
```

- Se visualizza una lunga stringa che inizia con ssh-rsa o ssh-dsa la chiave esiste e non è necessario crearla. Esempio:

```
ssh-rsa AAAAB3Nz[...]rQE8+oaEkw==  
idilio.drago@unito.it
```

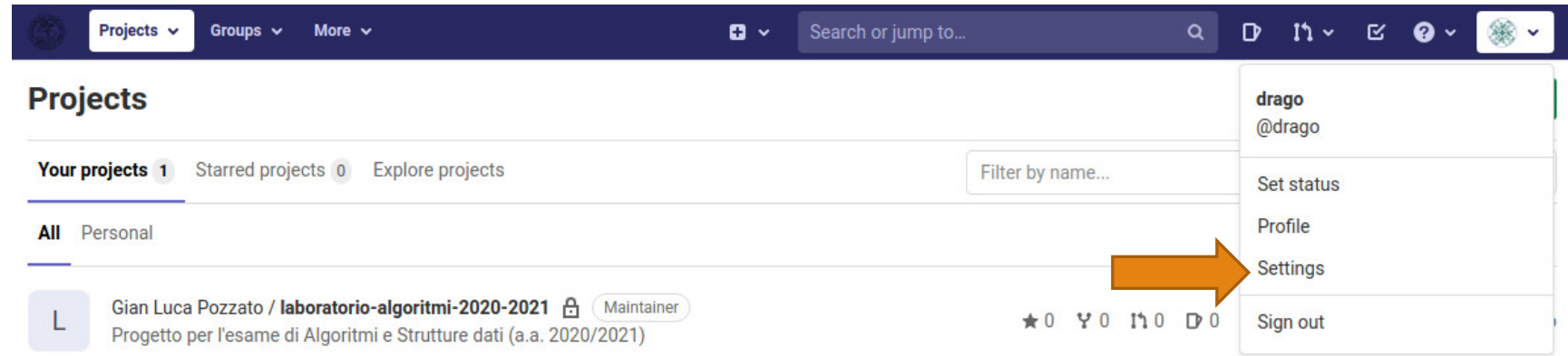
- Altrimenti, da terminale

```
ssh-keygen -t rsa -C "indirizzo email"
```


REGISTRAZIONE SU GITLAB

```
cat ~/.ssh/id_rsa.pub
```

- Accedere alla pagina personale su GitLab e cliccare su «Settings», quindi cliccare su «SSH Keys»



REGISTRAZIONE SU GITLAB

```
cat ~/.ssh/id_rsa.pub
```

- Procedere con l'aggiunta della chiave generata
- Copia e incolla della chiave generata

User Settings > SSH Keys

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Do not paste your private SSH key, as that can compromise your identity.

Typically starts with "ssh-ed25519 ..." or "ssh-rsa ..."

Title **Expires at**

e.g. My MacBook key mm/dd/yyyy

Give your individual key a title.

Add key

Your SSH keys (1)

idrago@joaninha	c9:6b:ec:77:e6:da:ae:2c:2b:86:b1:a5:99:9f:50:04	Last used: Never	Expires: 2025-09-01	Created 2 weeks ago
-----------------	---	------------------	---------------------	---------------------

CREAZIONE DEL PROGETTO

- Fare login su

<https://gitlab2.educ.di.unito.it>
(credenziali del laboratorio)

- Utilizzare l'interfaccia web per creare una copia personale del repository su cui lavorerete
- Un membro per ogni gruppo acceda all'url
`https://gitlab2.educ.di.unito.it/drago/laboratorio-algoritmi-2021-2022`
- Clicchi sull'icona «Fork»

CREAZIONE DEL PROGETTO

- Clicchi sull'icona «Fork»

The screenshot shows the OpenLab interface for a project named 'laboratorio-algoritmi-2020-2021'. The left sidebar contains navigation links: Project overview, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages & Registries, Analytics, Wiki, Snippets, and Members. The main content area displays the project name, ID (4685), and a 'Leave project' link. It also shows statistics: 6 Commits, 1 Branch, 0 Tags, 225 KB Files, and 225 KB Storage. A description states: 'Progetto per l'esame di Algoritmi e Strutture dati (a.a. 2020/2021)'. An 'Auto DevOps' section explains that it will automatically build, test, and deploy the application based on a predefined CI/CD configuration, with a link to the documentation and an 'Enable in settings' button. Below this, there are buttons for 'History', 'Find file', 'Web IDE', a download icon, and a 'Clone' button. A commit history section shows a commit titled 'Small change in 1st exercise' by Idilio Drago, authored 4 days ago, with a commit hash of 7f695890. At the bottom, there are buttons to add README, LICENSE, CHANGELOG, CONTRIBUTING, Kubernetes cluster, and set up CI/CD. A table lists the project files:

Name	Last commit	Last update
FAQ.md	Project created	2 weeks ago
Git.md	Project created	2 weeks ago

CREAZIONE DEL PROGETTO

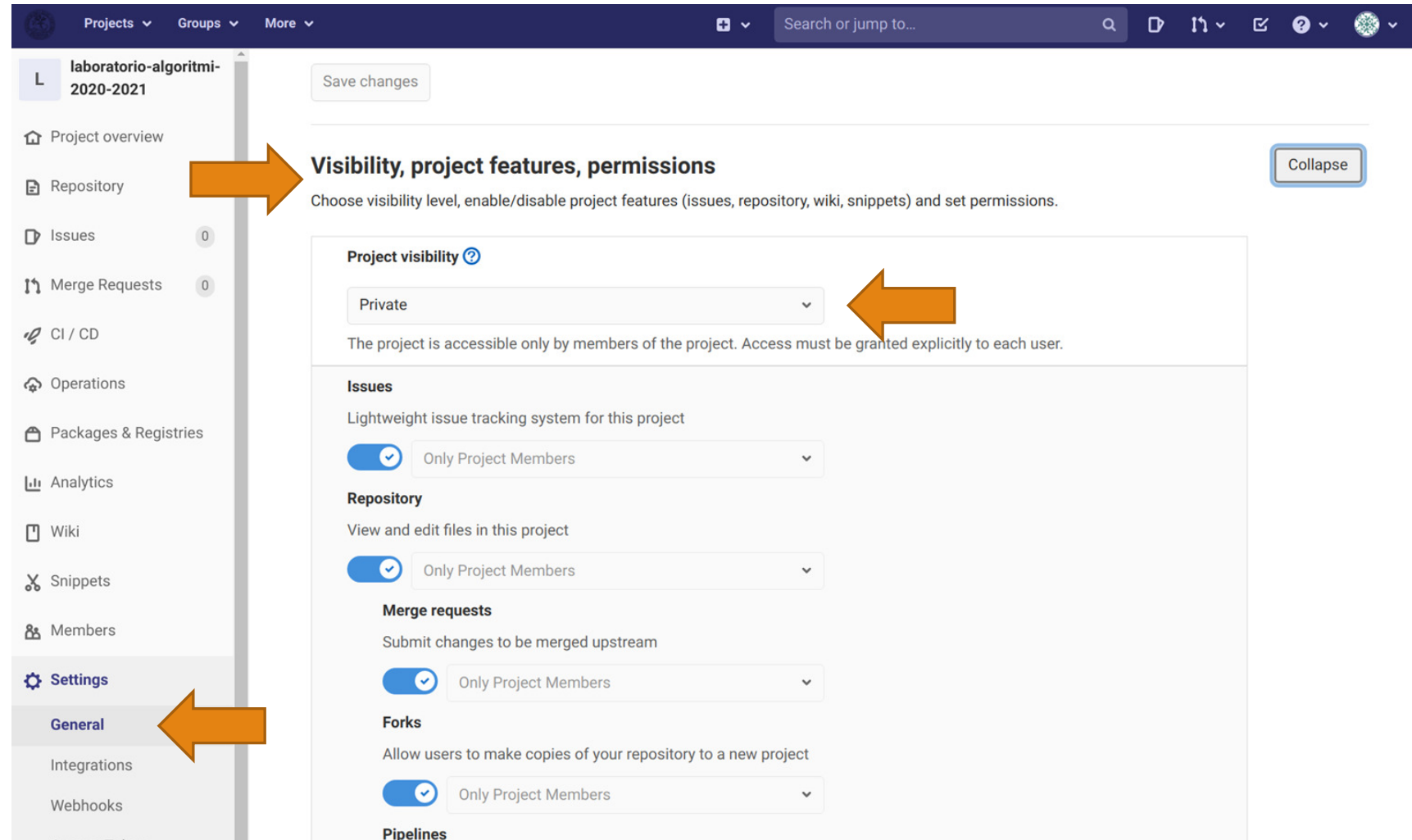
- Aggiungere al gruppo di sviluppo del progetto (Settings - Members)
 - I membri del gruppo come maintainer
 - I docenti di laboratorio (drago, magro, micalizi, pozzato) come developer



The screenshot shows the GitLab web interface for a project named 'laboratorio-algoritmi-2020-2021'. The left sidebar contains navigation links: Project overview, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages & Registries, Analytics, Wiki, Snippets, Members (highlighted with a blue bar), and Settings. The main content area is titled 'Members 3' and lists three members: drago @drago (with a green 'It's you' badge), Gian Luca Pozzato @pozzato, and magro @magro. All three members have 'Given access 2 weeks ago'. A 'Change permissions' modal is open over the member list, showing a dropdown menu with options: Guest, Reporter, Developer, and Maintainer (which is selected with a checkmark). An orange arrow points from the right towards the 'Maintainer' option in the modal. In the background, the member's current role is shown as 'Maintainer' with a 'Leave' button. At the bottom of the modal, there is an 'Expiration date' field and a trash icon.

PERMESSI DEL PROGETTO

- Impostare i permessi del repository creato a «private»
- Non permettere l'accesso a nessun altro utente che non sia esplicitamente un membro del gruppo di sviluppatori del repository
- Plagio (anche involontario) «progetto annullato»



CLONE

Creazione della copia locale del repository

```
git clone REPO_REF
```

REPO_REF è il nome del repository ottenuto dal fork

The screenshot shows the GitLab web interface for a project named 'laboratorio-algoritmi-2020-2021'. The left sidebar contains a navigation menu with options like 'Project overview', 'Details', 'Activity', 'Releases', 'Repository', 'Issues', 'Merge Requests', 'CI / CD', 'Operations', 'Packages & Registries', 'Analytics', 'Wiki', 'Snippets', 'Members', and 'Settings'. The main content area displays the project details, including the 'Clone' button. A dropdown menu is open, showing options to clone the repository using SSH or HTTPS. An orange arrow points to the 'Clone with HTTPS' option.

laboratorio-algoritmi-2020-2021

Project ID: 4685 [Leave project](#)

6 Commits 1 Branch 0 Tags 225 KB Files 225 KB Storage

Progetto per l'esame di Algoritmi e Strutture dati (a.a. 2020/2021)

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the [Auto DevOps documentation](#)

Enable in settings

master laboratorio-algoritmi-2020-2021 / +

History Find file Web IDE Clone

Small change in 1st exercise
Idilio Drago authored 5 days ago

README Add LICENSE Add CHANGELOG Add CONTRIBUTING

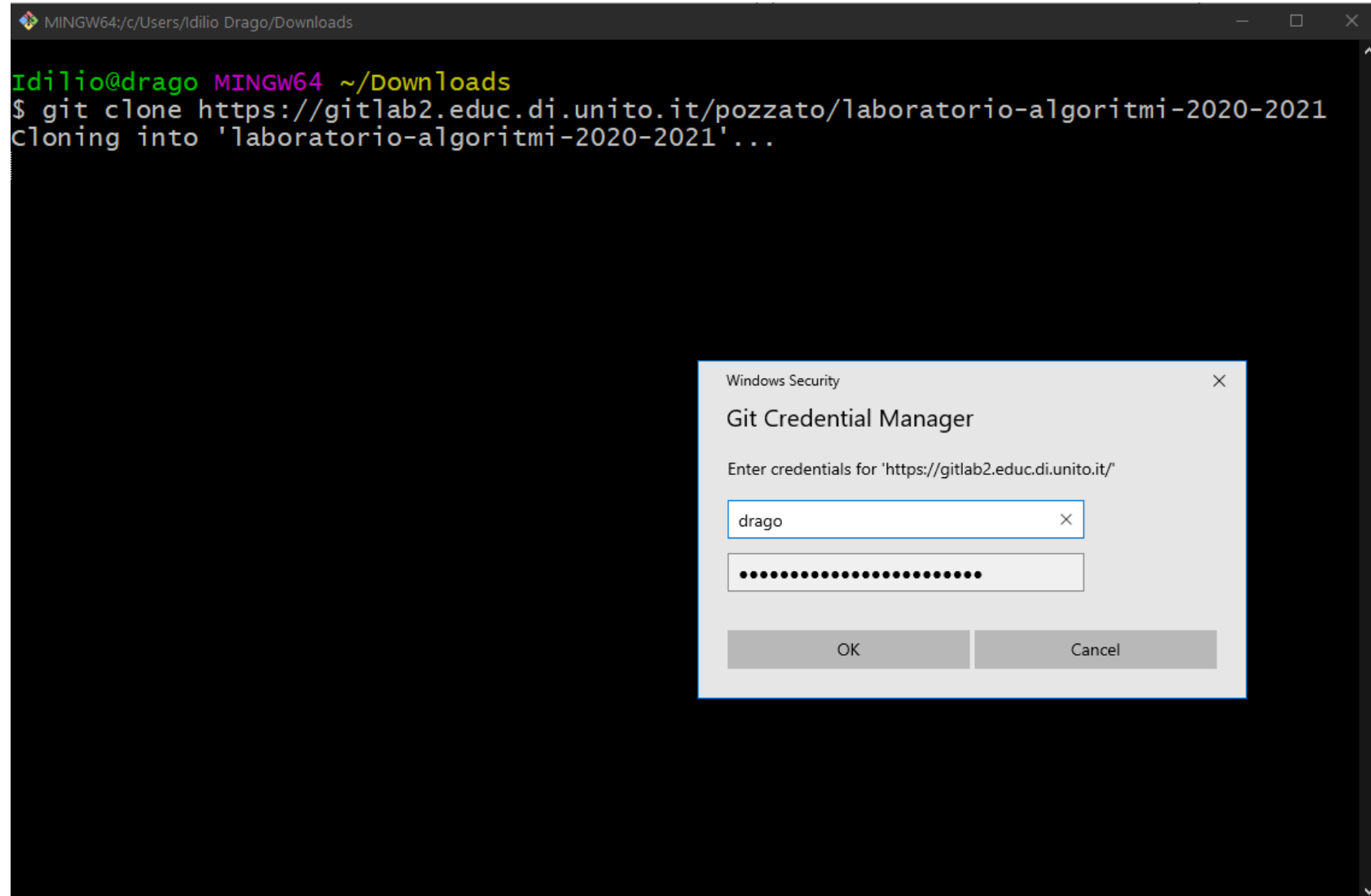
Name	Last commit	Last update
FAQ.md	Project created	2 weeks ago
Git.md	Project created	2 weeks ago
README.md	Small change in 1st exercise	5 days ago
UnitTesting.md	Project created	2 weeks ago

CLONE - Windows

Creazione della copia locale
del repository

`git clone REPO_REF`

REPO_REF è il nome del
repository ottenuto dal fork



COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

```
git status
```

mostra lo stato del progetto. File identici all'ultima versione salvata non vengono mostrati. File modificati, cancellati, o nuovi vengono visualizzati (con colori diversi se avete abilitato questa opzione)

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

`git log`

mostra un log dell'evoluzione del progetto. Nel caso il progetto sia stato clonato le informazioni includeranno la storia del progetto originale

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

```
git pull
```

recupera lo stato del progetto «remoto» associato alla copia corrente e lo fonde con la copia attuale

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

```
git add FILE_1 FILE_2 ... FILE_N
```

aggiunge uno o più file alla «staging area». Il contenuto di questa area determinerà quali modifiche andranno a fare parte del prossimo commit. È necessario aggiungere esplicitamente tutti i file che sono stati modificati o creati

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

```
git rm FILE_1 FILE_2 ... FILE_N
```

rimuove uno o più file sia dalla directory locale che sul repository (al prossimo commit). È necessario aggiungere esplicitamente tutti i file che sono stati cancellati.

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

```
git commit -am "messaggio"
```

salva lo stato del progetto («messaggio finirà nei log del progetto)

COMANDI DI BASE

Git consente di salvare il lavoro svolto e mantiene la storia delle varie versioni create insieme all'indicazione di chi ha effettuato ogni aggiornamento

Comandi utili:

`git push`

salva le modifiche fatte al repository locale integrandole nel repository remoto

Uso tipico

```
git pull
```

(mi accerto che la copia locale sia sincronizzata con quella remota)
(modifiche alla copia locale del progetto)

```
git add <lista dei file modificati>
```

```
git commit -m "messaggio modifiche apportate"
```

```
git push
```

(aggiornamento della copia remota)