

# Programmazione Mobile

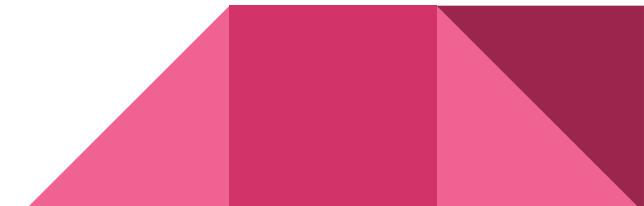
**Nicola Noviello**

[nicola.noviello@unimol.it](mailto:nicola.noviello@unimol.it)

Corso di Laurea in Informatica  
Dipartimento di Bioscienze e Territorio  
Università degli Studi del Molise  
Anno 2023/2024

# Lezione: Introduzione a Git (parte seconda)

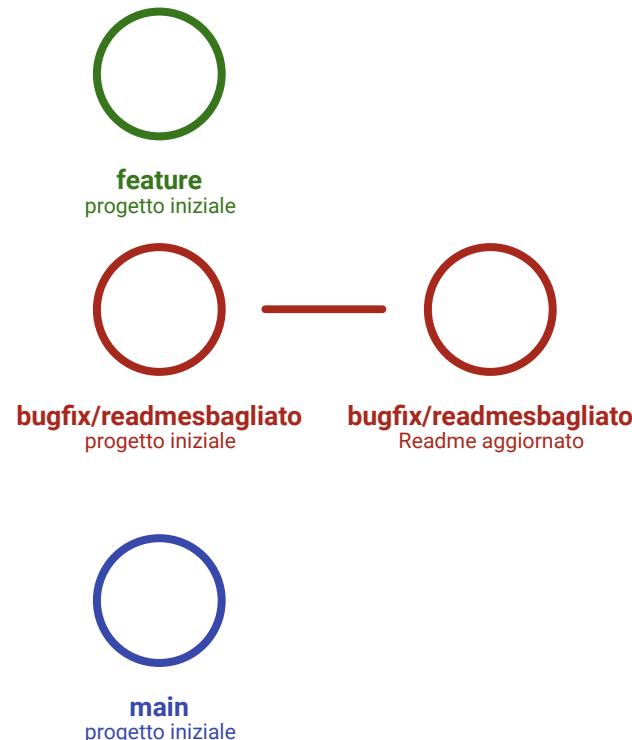
- Concetti base di Git
- Comandi principali
- Branches
- Pull (Merge) Request
- Risoluzione dei conflitti
- .gitignore
- Cenni introduttivi di tecniche avanzate





Dove eravamo

# Situazione attuale dei branch



# Situazione attuale dei branch



ma il nostro obiettivo è  
quello di avere sempre il  
branch di main allineato



# Pull ( o Merge) Request

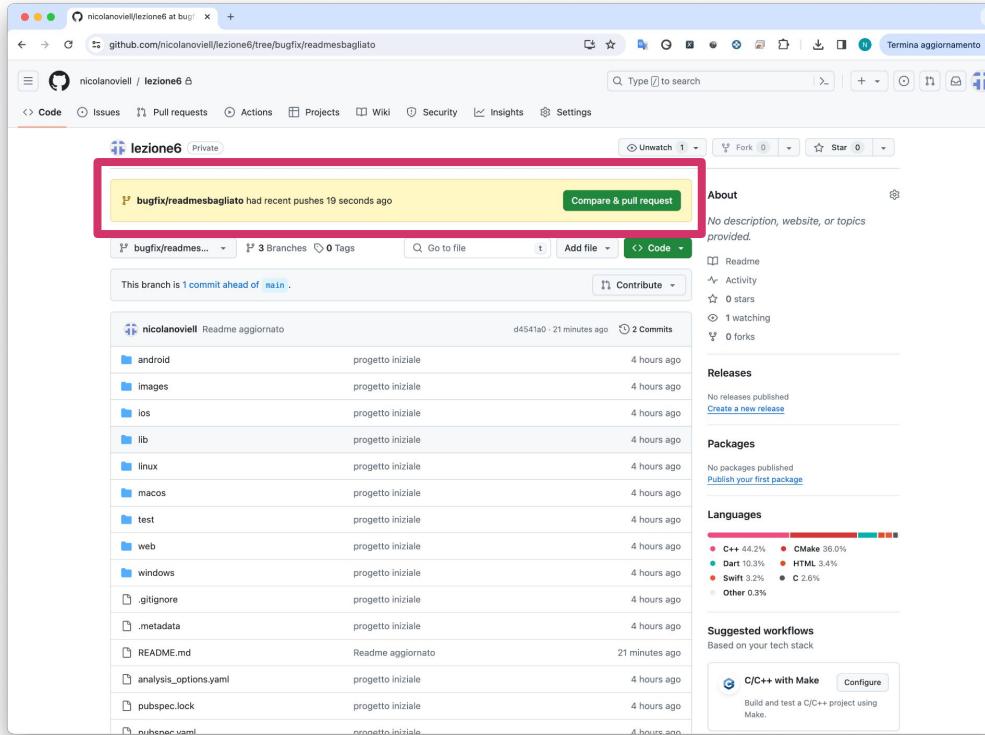
# Pull Request

**Buone pratiche:** gli altri sviluppatori devono fare review del codice prima di effettuare una merge tra due branch

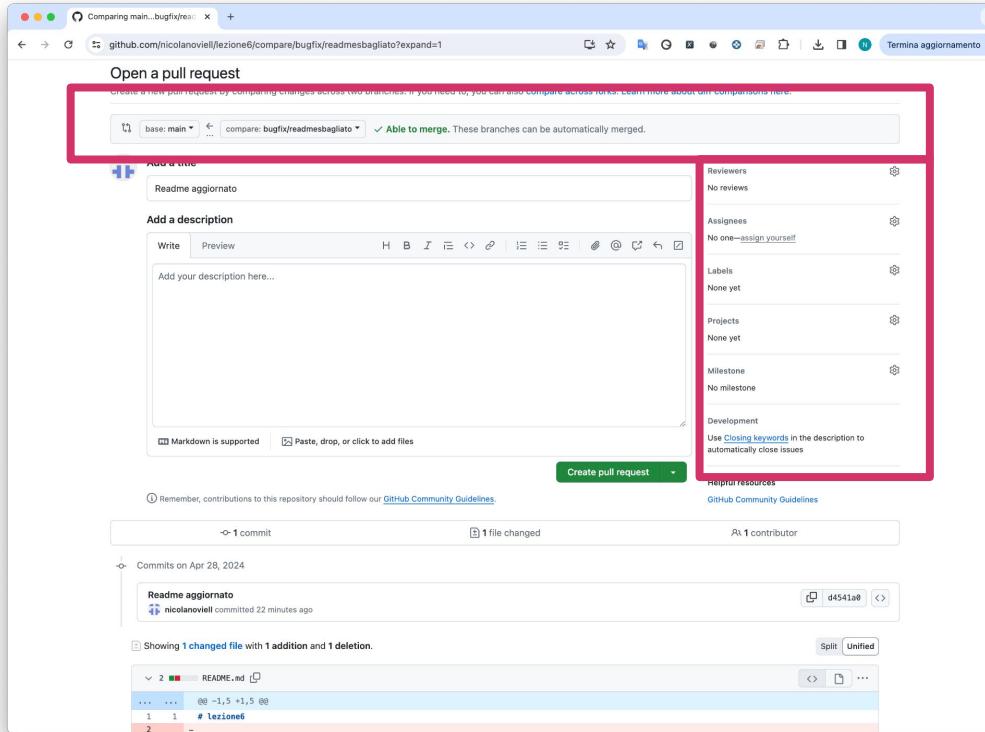
**Use case (quando):** funzionalità complesse, sviluppatori inesperti oppure con esperienza in altre aree

**Opportunità:** nella programmazione c'è crescita solo con la condivisione. Le review sono opportunità per imparare e migliorarsi

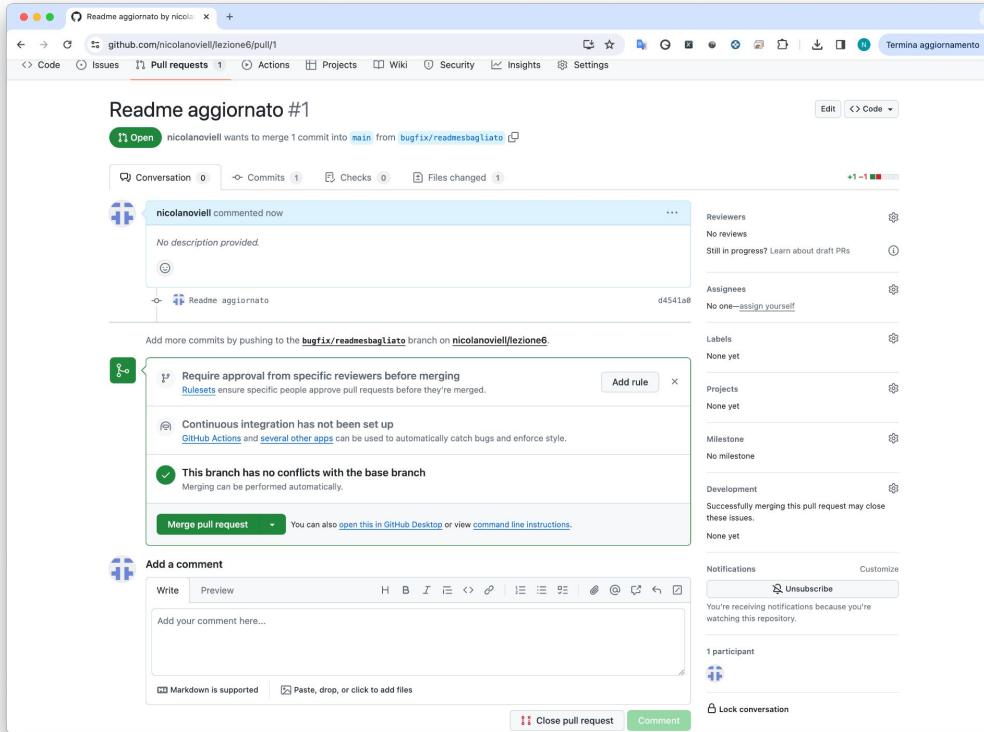
# Pull request



# Pull request



# Pull request



# Pull request

The screenshot shows a GitHub pull request page for a repository named "Readme aggiornato #1". The pull request has been merged by "nicolanoivell" into the "main" branch. A message indicates that the "bugfix/readmesbagliato" branch can be safely deleted. A red box highlights the "Delete branch" button.

Merged nicolanoivell merged 1 commit into main from bugfix/readmesbagliato now

Conversation 0 Commits 1 Checks 0 Files changed 1

nicolanoivell commented 1 minute ago  
No description provided.

nicolanoivell merged commit c28da76 into main now

Pull request successfully merged and closed  
You're all set—the bugfix/readmesbagliato branch can be safely deleted.

Delete branch

Add a comment

Write Preview

Add your comment here...

Markdown is supported Paste, drop, or click to add files

Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

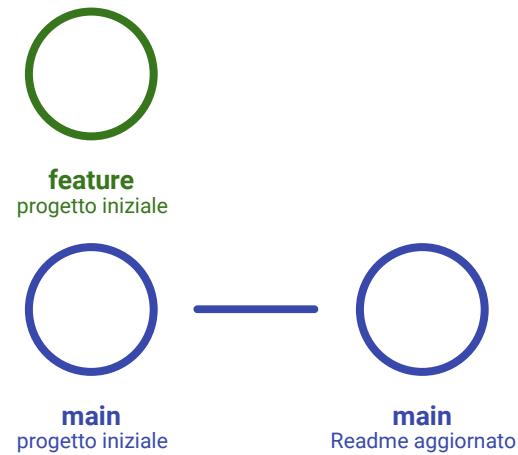
ProTip! Add .patch or .diff to the end of URLs for Git's plaintext views.

1 participant

Lock conversation

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

# Situazione attuale dei branch



# Cancellazione esplicita di un branch



# Cancellazione e pulizia costante dei branch

La gestione accurata dei branch in un repository Git è cruciale per mantenere l'ordine e la chiarezza nel processo di sviluppo. Eliminare i branch mergiati e vecchi offre numerosi vantaggi:

**Riduzione del disordine:** Eliminando i branch mergiati e vecchi, si evita l'accumulo di code obsolete e non più utilizzate nel repository. Ciò contribuisce a mantenere il repository pulito e organizzato;

**Risparmio di risorse:** La presenza di numerosi branch non utilizzati può occupare spazio e risorse preziose sul server. Rimuovendo questi branch, si ottimizza l'utilizzo delle risorse;

**Chiarezza nella gestione dei branch:** Mantenere solo i branch attivi e rilevanti semplifica la gestione e la navigazione nel repository. Gli sviluppatori possono concentrarsi sui branch pertinenti al lavoro attuale senza distrazioni da branch obsoleti;

**Prevenzione di conflitti:** Branch vecchi e non più utilizzati possono causare confusione e potenzialmente generare conflitti durante il processo di sviluppo. Eliminandoli, si riduce il rischio di errori e problemi di compatibilità.

# Cancellazione branch

The screenshot shows a web browser displaying the GitHub 'Branches' page for the repository 'nicolanoivelli / lezione6'. The URL is <https://github.com/nicolanoivelli/lezione6/branches>. The page has a light blue header with the GitHub logo and the repository name. Below the header, there are navigation links: Code (which is underlined in red), Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

The main content area is titled 'Branches' and contains three sections: 'Default', 'Your branches', and 'Active branches'. Each section has a table with columns: Branch, Updated, Check status, Behind | Ahead, and Pull request. A green 'New branch' button is located in the top right corner of the 'Branches' title bar.

- Default:** Shows one branch, 'main', which was updated 30 minutes ago and is marked as 'Default'. The 'Pull request' column contains a trash icon and three dots, which is highlighted with a red rectangle.
- Your branches:** Shows one branch, 'feature', which was updated 1 hour ago. The 'Pull request' column contains a trash icon and three dots.
- Active branches:** Shows one branch, 'feature', which was updated 1 hour ago. The 'Pull request' column contains a trash icon and three dots.

# Cancellazione branch

Ho dimenticato di mettere uno screen dove erano presenti ancora in locale i branch eliminati!

```
nico@Mac-mini-di-Nico lezione6 % git pull origin main
From github.com:nicolanoviell/lezione6
 * branch           main      -> FETCH_HEAD
Already up to date.
nico@Mac-mini-di-Nico lezione6 % git branch -d bugfix/readmesbagliato
Deleted branch bugfix/readmesbagliato (was c28da76).
nico@Mac-mini-di-Nico lezione6 % git branch -d feature
warning: not deleting branch 'feature' that is not yet merged to
      'refs/remotes/origin/feature', even though it is merged to HEAD.
error: The branch 'feature' is not fully merged.
If you are sure you want to delete it, run 'git branch -D feature'.
nico@Mac-mini-di-Nico lezione6 % git branch -D feature
Deleted branch feature (was c28da76).
nico@Mac-mini-di-Nico lezione6 % git branch
* main
nico@Mac-mini-di-Nico lezione6 %
```

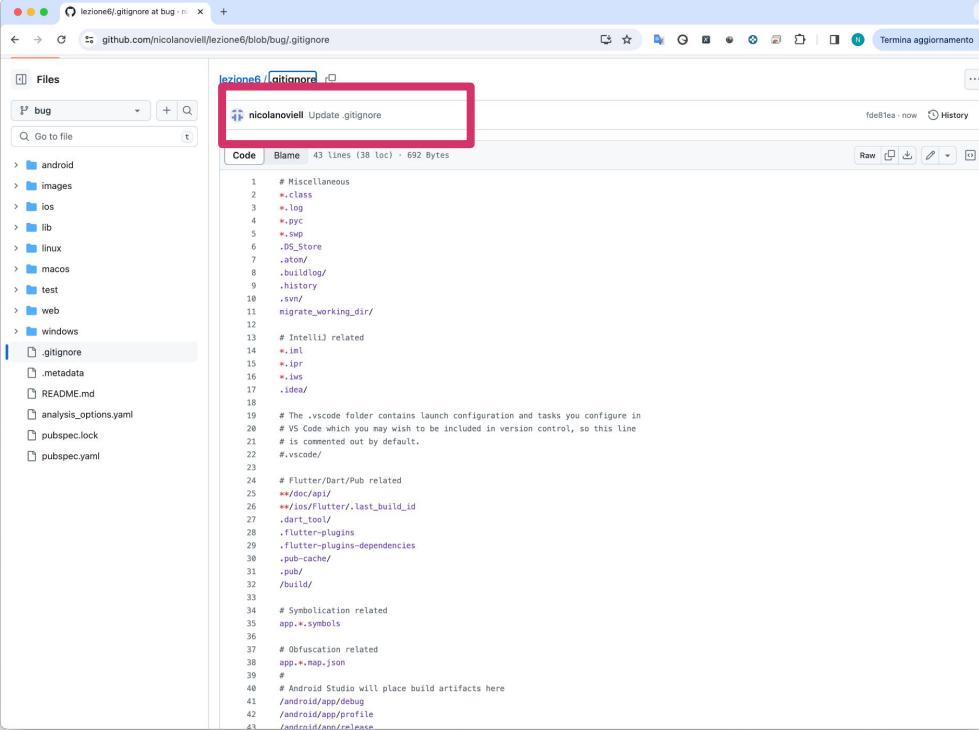


Lavorare con più  
utenti sullo stesso  
branch

# Creo una commit su un nuovo branch...

```
nico@Mac-mini-di-Nico lezione6 % git fetch origin --prune
nico@Mac-mini-di-Nico lezione6 % git branch -r
  origin/bug
  origin/main
nico@Mac-mini-di-Nico lezione6 % git checkout bug
Switched to branch 'bug'
Your branch is up to date with 'origin/bug'.
nico@Mac-mini-di-Nico lezione6 % vim README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit - "aggiornamento readme nuovo"
error: pathspec '-' did not match any file(s) known to git
error: pathspec 'aggiornamento readme nuovo' did not match any file(s) known to git
nico@Mac-mini-di-Nico lezione6 % git commit -m "aggiornamento readme nuovo"
[bug daa6ffb] aggiornamento readme nuovo
  1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 %
```

# ...ma lavoro sul branch anche dal client web



The screenshot shows a web browser window displaying a GitHub repository page. The URL in the address bar is `github.com/nicolanovielli/lezione6/blob/bug/.gitignore`. The page title is "lezione6/.gitignore at bug · n...". The left sidebar shows a tree view of the repository's directory structure, with the ".gitignore" file selected. The main content area displays the contents of the ".gitignore" file, which is a text-based ignore list for version control. A red box highlights the code editor area where the file content is being viewed.

```
1  # Miscellaneous
2  *.class
3  *.log
4  *.pyc
5  *.swp
6  .DS_Store
7  .atom/
8  .buildlog/
9  .history
10 .svn/
11 migrate_working_dir/
12
13 # IntelliJ related
14 *.iml
15 *.ipr
16 *.iws
17 .idea/
18
19 # The .vscode folder contains launch configuration and tasks you configure in
20 # VS Code which you may wish to be included in version control, so this line
21 # is commented out by default.
22 #.vscode/
23
24 # Flutter/Dart/Pub related
25 **/doc/api/
26 **/ios/flutter/.last_build_id
27 .dart_tool/
28 .flutter-plugins
29 .flutter-plugins-dependencies
30 .pub-cache/
31 .pub/
32 ./build/
33
34 # Symbolication related
35 app.*.symbols
36
37 # Obfuscation related
38 app.*.map.json
39 #
40 # Android Studio will place build artifacts here
41 ./android/app/debug
42 ./android/app/profile
43 ./android/app/release
```

# Faccio una push sul remote branch

```
nico@Mac-mini-di-Nico lezione6 % git checkout bug
Switched to branch 'bug'
Your branch is up to date with 'origin/bug'.
nico@Mac-mini-di-Nico lezione6 % vim README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit - "aggiornamento readme nuovo"
error: pathspec '-' did not match any file(s) known to git
error: pathspec 'aggiornamento readme nuovo' did not match any file(s) known to git
nico@Mac-mini-di-Nico lezione6 % git commit -m "aggiornamento readme nuovo"
[bug daa6ffb] aggiornamento readme nuovo
 1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git push
To github.com:nicolanoviell/lezione6.git
 ! [rejected]        bug -> bug (fetch first)
error: failed to push some refs to 'github.com:nicolanoviell/lezione6.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
nico@Mac-mini-di-Nico lezione6 %
```

La push mi genera un errore

# git pull

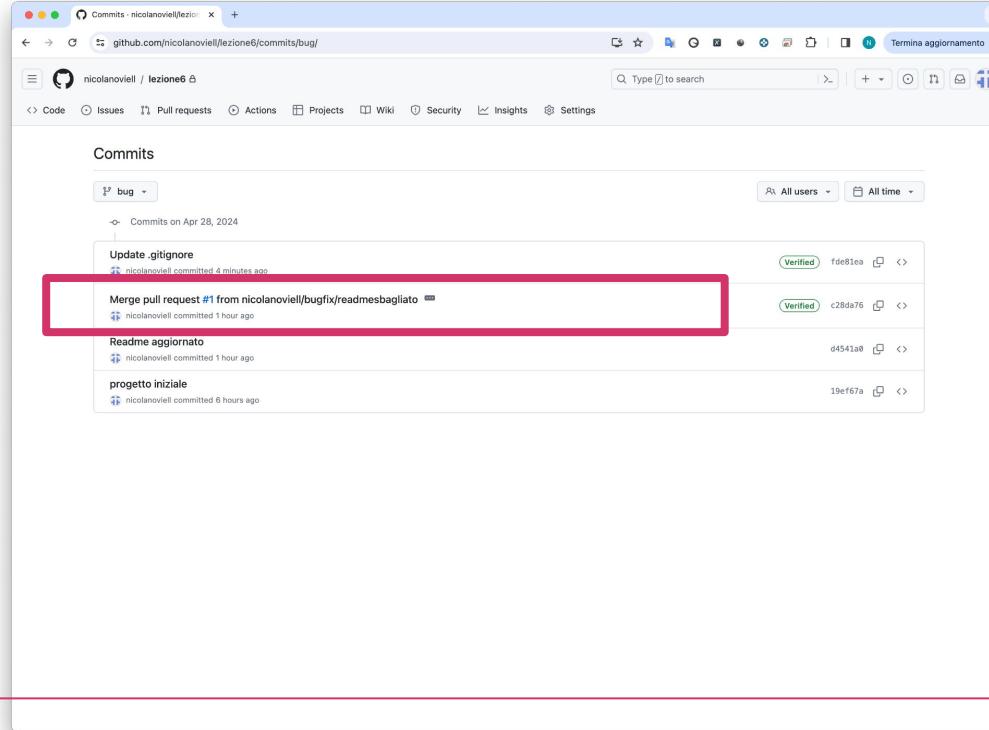
È necessario effettuare un **git pull** prima di effettuare una **git push** se sul branch remoto ci sono state modifiche

- Con **git pull**, Git esegue due azioni separate: **git fetch** per recuperare i cambiamenti dal repository remoto e **git merge** per integrare questi cambiamenti nel branch locale;
- Questo processo può generare un commit di merge aggiuntivo nel branch locale, creando una cronologia di commit più complessa;
- È utile quando si desidera mantenere la cronologia dei commit più esplicita, con i commit di merge che indicano chiaramente quando sono stati integrati i cambiamenti dal repository remoto.

# git pull

```
From github.com:nicolanoviell/lezione6
  c28da76..fde81ea  bug      -> origin/bug
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only      # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
nico@Mac-mini-di-Nico lezione6 % git config pull.rebase false
nico@Mac-mini-di-Nico lezione6 % git pull
Merge made by the 'ort' strategy.
 .gitignore | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 %
```

# remote branch history



Crea una merge nella history

# git pull –rebase

```
nico@Mac-mini-di-Nico lezione6 % vim README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "nuovo aggiornamento readme"
[bug b79f30c] nuovo aggiornamento readme
 1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git push
To github.com:nicolanoviell/lezione6.git
 ! [rejected]      bug -> bug (fetch first)
error: failed to push some refs to 'github.com:nicolanoviell/lezione6.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

La push mi genera un errore

# rebase

Il rebase è un'operazione potente ma può comportare alcuni rischi o situazioni in cui è sconsigliato utilizzarlo:

- **Branch condivisi:** Se si lavora su un branch condiviso con altri membri del team, il rebase può modificare la cronologia dei commit in modo significativo. Questo potrebbe creare confusione e rendere difficile il lavoro collaborativo, specialmente se altri membri del team hanno già basato il proprio lavoro sul branch originale;
- **Commit già condivisi:** Se i commit del branch locale sono già stati condivisi con altri sviluppatori o inclusi in una repository remota, modificare la cronologia dei commit attraverso il rebase può creare confusione e problemi di compatibilità. Altri membri del team potrebbero avere difficoltà nell'integrare i cambiamenti o potrebbero perdere la tracciabilità dei commit;
- **Branch pubblici o condivisi:** Se il branch su cui si desidera eseguire il rebase è pubblico o condiviso con altri utenti, modificare la cronologia dei commit può essere rischioso. Potrebbe essere preferibile utilizzare il merge per integrare i cambiamenti in modo da mantenere una cronologia dei commit più chiara e prevenire problemi di compatibilità;
- **Storia dei commit significativa:** Se la cronologia dei commit ha una grande importanza storica, ad esempio per fini di tracciamento o audit, il rebase potrebbe rendere difficile seguire il flusso del lavoro nel tempo. In questi casi, è preferibile mantenere la cronologia dei commit intatta utilizzando il merge anziché il rebase.

# git pull –rebase

- Con **git pull --rebase**, Git esegue anche due azioni: **git fetch** per recuperare i cambiamenti dal repository remoto e **git rebase** per applicare i cambiamenti del repository remoto sopra i commit locali, riscrivendo la cronologia dei commit locali;
- Questo processo riscrive la cronologia dei commit locali, evitando la creazione di commit di merge aggiuntivi e mantenendo una cronologia lineare dei commit;
- È utile quando si desidera mantenere una cronologia dei commit più pulita e lineare, senza la presenza di commit di merge aggiuntivi.

# git pull –rebase

```
nico@Mac-mini-di-Nico lezione6 % git pull --rebase
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 956 bytes | 956.00 KiB/s, done.
From github.com:nicolanoviell/lezione6
  fde81ea..8f72c2a  bug      -> origin/bug
Successfully rebased and updated refs/heads/bug.
nico@Mac-mini-di-Nico lezione6 % git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 586 bytes | 586.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To github.com:nicolanoviell/lezione6.git
  8f72c2a..fd9eb60  bug -> bug
```

# remote branch history

The screenshot shows a GitHub repository's commit history for the branch 'bug'. A red box highlights a group of three commits by the user 'nicolanovali' that have been squashed into a single commit message: 'nuovo aggiornamento readme', 'aggiornamento readme nuovo', and 'Update .gitignore 2'. This demonstrates that multiple commits can be combined into a single commit message, which is a common practice in Git.

Commit Message	Author	Date	SHA
nuovo aggiornamento readme	nicolanovali	committed 1 minute ago	fd9eb60
aggiornamento readme nuovo	nicolanovali	committed 1 minute ago	1e11918
Update .gitignore 2	nicolanovali	committed 1 minute ago	8f72c2a
Update .gitignore	nicolanovali	committed 7 minutes ago	fde81ea
Merge pull request #1 from nicolanovali/bugfix/readmesbagliato	nicolanovali	committed 1 hour ago	c28da76
Readme aggiornato	nicolanovali	committed 1 hour ago	d4541a0
progetto iniziale	nicolanovali	committed 6 hours ago	19ef67a

senza merge nella history

# Esercizio

Aggiungete una o un collega al vostro progetto e provate a lavorare sullo stesso branch ma su file differenti. Provate quanto visto a lezione.

# Gestione dei conflitti

# Modifico il file readme e creo una commit

```
# lezione6
Aggiornamento Readme con conflitto
A new Flutter project.

## Getting Started

This project is a starting point for a Flutter application.

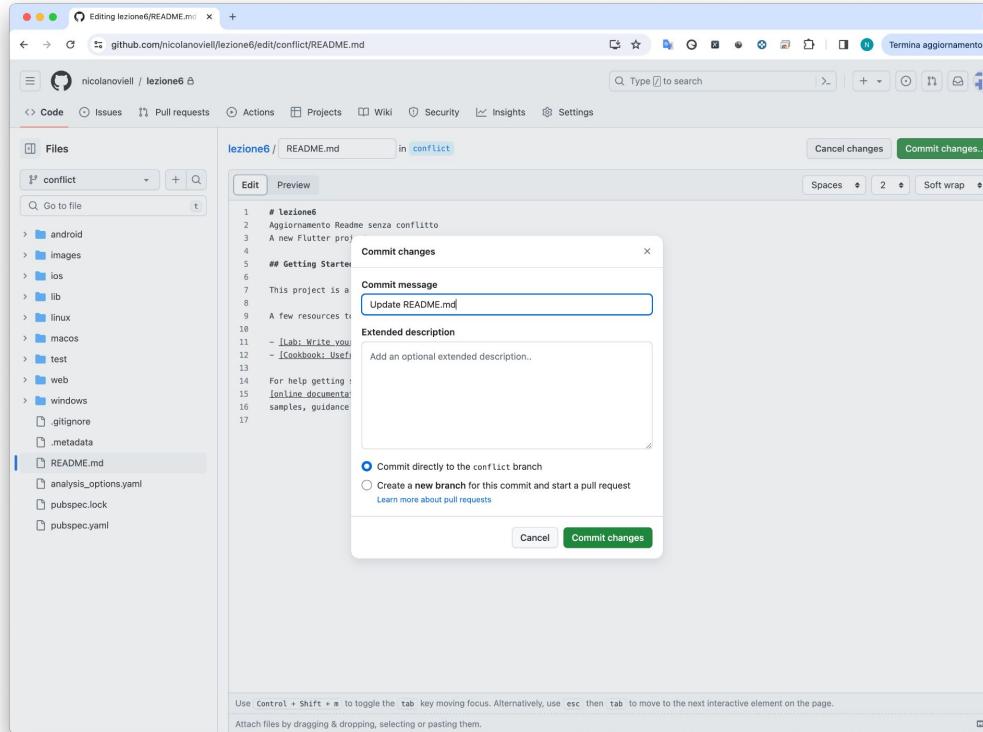
A few resources to get you started if this is your first Flutter project:

- [Lab: Write your first Flutter app](https://docs.flutter.dev/get-started/codelab)
- [Cookbook: Useful Flutter samples](https://docs.flutter.dev/cookbook)

For help getting started with Flutter development, view the
[online documentation](https://docs.flutter.dev/), which offers tutorials,
samples, guidance on mobile development, and a full API reference.

~  
~  
~  
~  
-- INSERT --
```

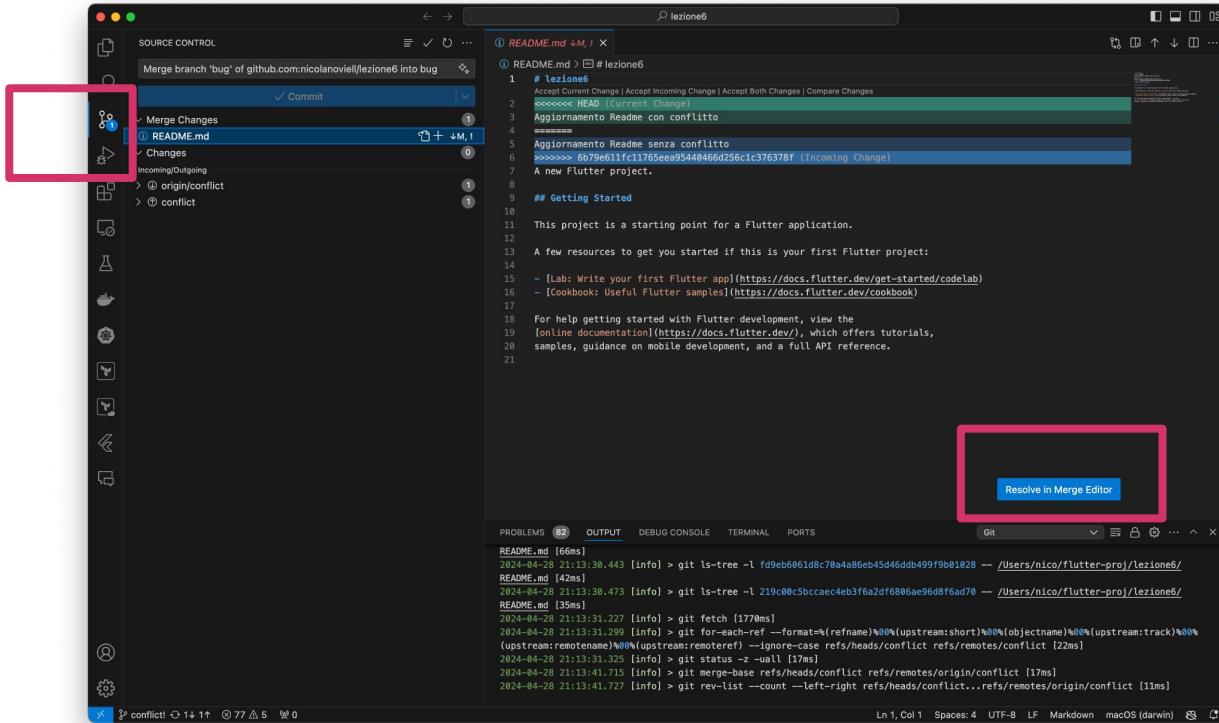
# Modifico lo stesso file da un altro client



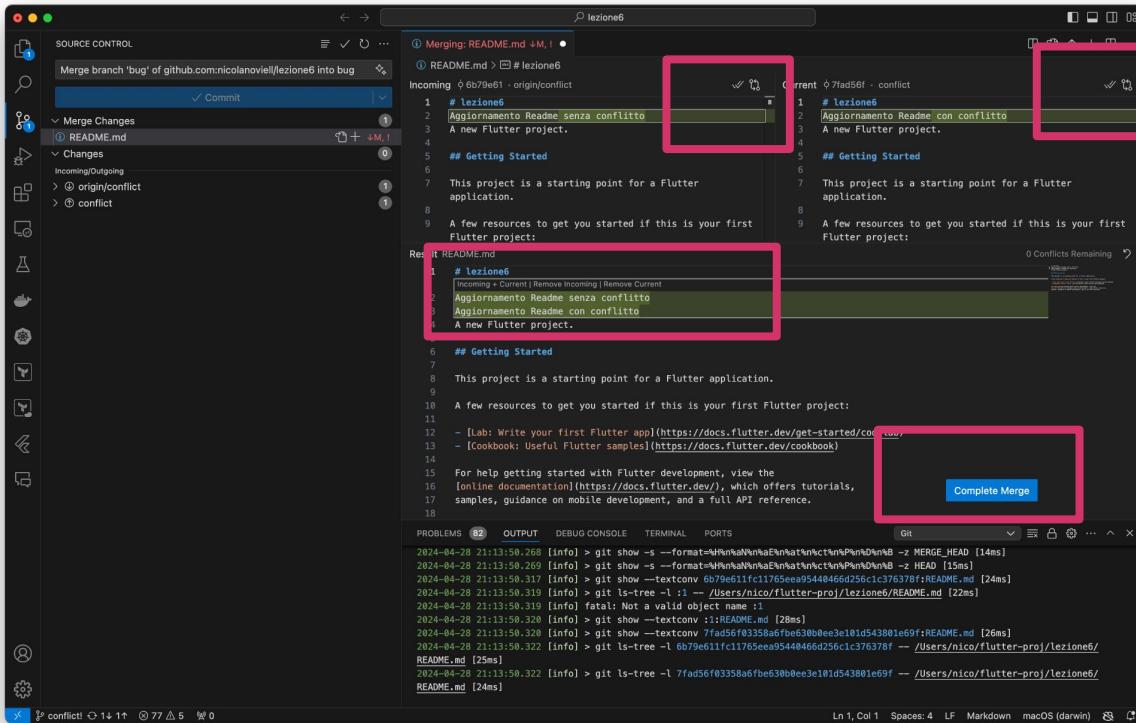
# provo a lanciare git pull ...

```
nico@Mac-mini-di-Nico lezione6 % git push
To github.com:nicolanoviell/lezione6.git
  ! [rejected]          conflict -> conflict (fetch first)
error: failed to push some refs to 'github.com:nicolanoviell/lezione6.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
nico@Mac-mini-di-Nico lezione6 % git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 954 bytes | 477.00 KiB/s, done.
From github.com:nicolanoviell/lezione6
  c28da76..6b79e61  conflict -> origin/conflict
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
nico@Mac-mini-di-Nico lezione6 %
```

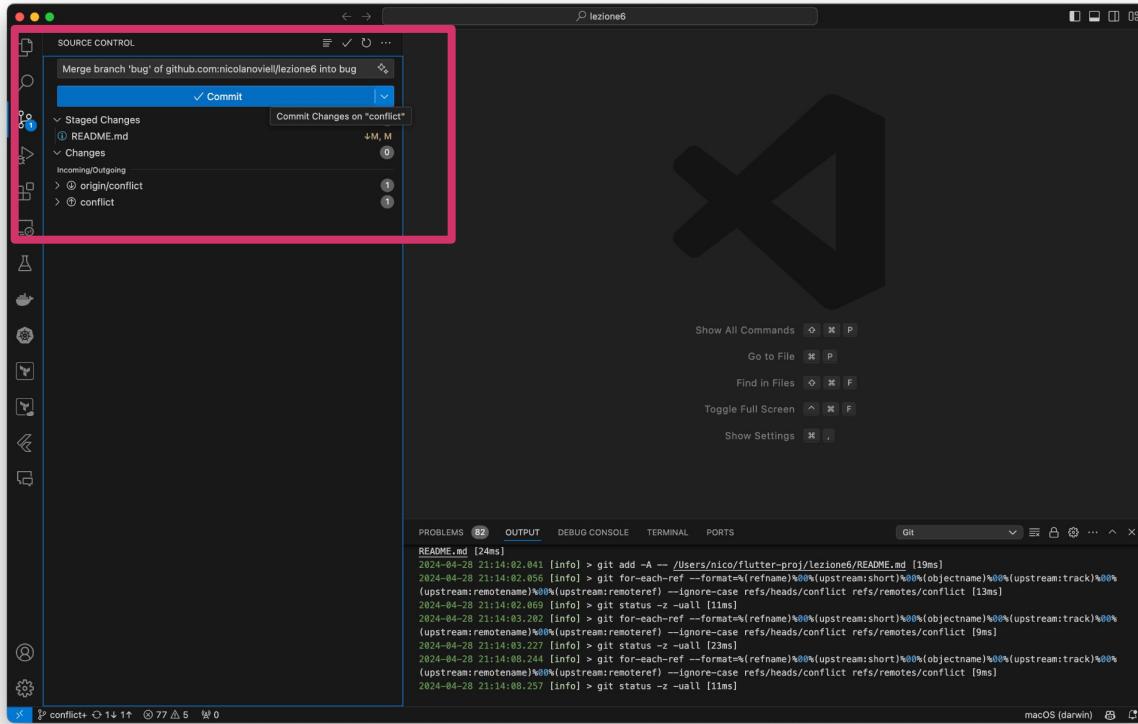
# Gestisco conflitti con l'IDE



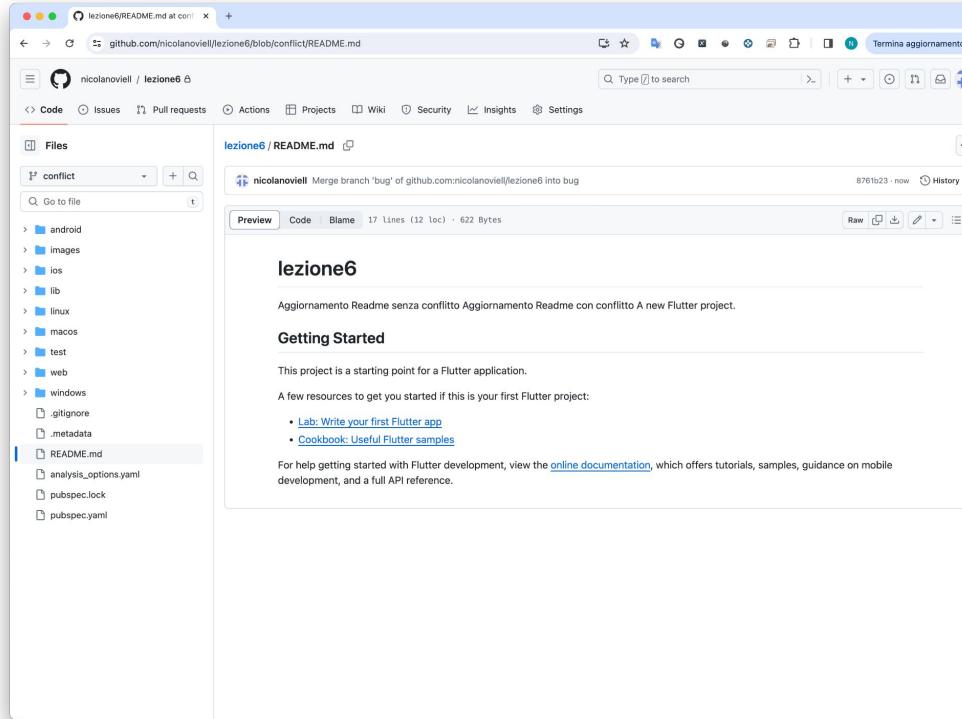
# Gestisco conflitti con l'IDE



# Gestisco conflitti con l'IDE



# Gestisco conflitti con l'IDE



# Esercizio

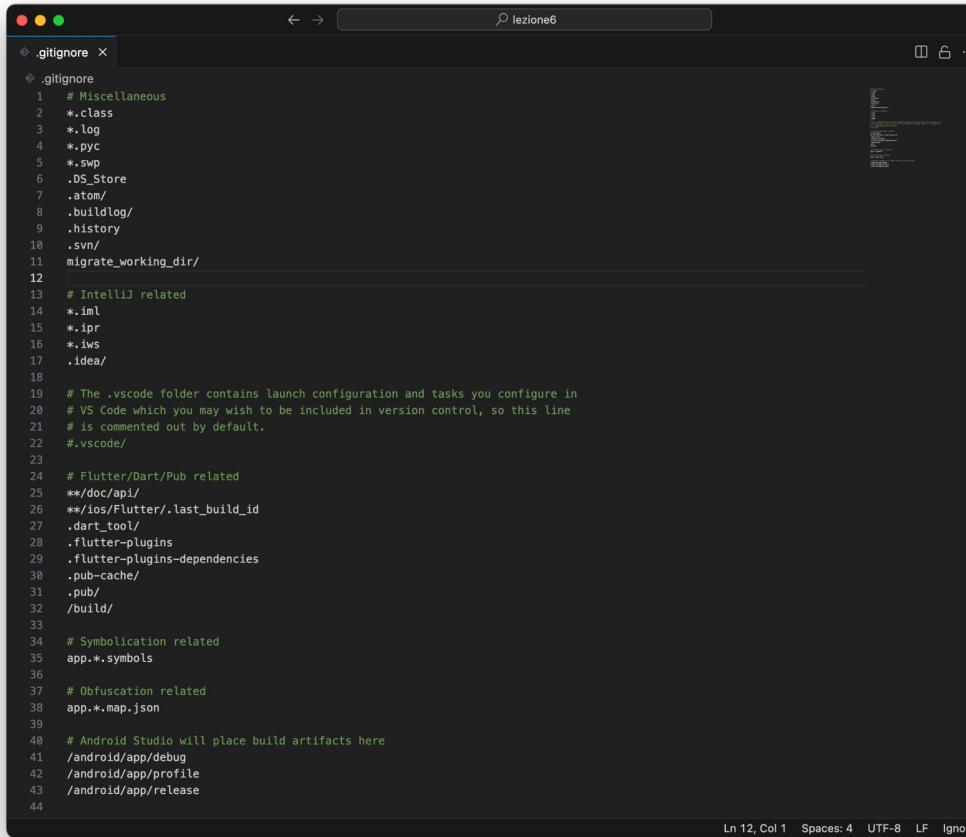
Aggiungete una o un collega al vostro progetto e provate a lavorare sullo stesso branch e sullo stesso file. Gestite in autonomia i conflitti evitando ove possibile l'uso della violenza verbale e non.

# .gitignore

# .gitignore

- **Cos'è il file .gitignore?**: un file speciale utilizzato in un repository Git per specificare i file e le directory che Git dovrebbe ignorare durante le operazioni di versioning;
- **Evita il tracciamento di file non desiderati**: previene il tracciamento di file temporanei, di build, di configurazione locale e di altre risorse non rilevanti per il repository;
- **Pulizia del repository**: si garantisce un repository pulito e ordinato, concentrando l'attenzione sui file e sul codice rilevanti;
- **Come lo fa**: specifica pattern di nomi di file o directory che Git deve ignorare. I pattern possono includere wildcard per identificare gruppi di file (ad esempio \*.log per ignorare tutti i file con estensione .log);
- **Vantaggi**: migliora la gestione dei file facilitando il lavoro con file temporanei o generati automaticamente e aiuta a evitare conflitti dovuti a file non rilevanti durante la collaborazione con altri sviluppatori.

# esempio di .gitignore



```
.gitignore
. .gitignore
# Miscellaneous
*.class
*.log
*.pyc
*.swp
.DS_Store
.atom/
.buildlog/
.history
.svn/
migrate_working_dir/
# IntelliJ related
*.iml
*.ipr
*.iws
.idea/
# The .vscode folder contains launch configuration and tasks you configure in
# VS Code which you may wish to be included in version control, so this line
# is commented out by default.
#.vscode/
# Flutter/Dart/Pub related
**/doc/api/
**/ios/Flutter/.last_build_id
.dart_tool/
.flutter-plugins
.flutter-plugins-dependencies
.pub-cache/
.pub/
/build/
# Symbolication related
app.*.symbols
# Obfuscation related
app.*.map.json
# Android Studio will place build artifacts here
.android/app/debug
.android/app/profile
.android/app/release
Ln 12, Col 1  Spaces: 4  UTF-8  LF  Ignore
```

# Cosa succede se aggiungo al .gitignore qualcosa che ho già pushato sul remote repository?

```
images/
# Miscellaneous
*.class
*.log
*.pyc
*.swp
.DS_Store
.atom/
.buildlog/
.history
.svn/
migrate_working_dir/

# IntelliJ related
*.iml
*.ipr
*.iws
.idea/

# The .vscode folder contains launch configuration and tasks you configure in
-- INSERT --
```

Aggiungo al .gitignore la directory images che era già stata precedentemente pushata sul repository

# Cosa succede se aggiungo al .gitignore qualcosa che ho già pushato sul remote repository?

```
nico@Mac-mini-di-Nico lezione6 % vim .gitignore
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "aggiunta la dir images su .gitignore"
[conflict 384fa3c] aggiunta la dir images su .gitignore
 1 file changed, 1 insertion(+)
nico@Mac-mini-di-Nico lezione6 % git pull
Already up to date.
nico@Mac-mini-di-Nico lezione6 % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 333 bytes | 333.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:nicolanoviell/lezione6.git
 8761b23..384fa3c conflict -> conflict
nico@Mac-mini-di-Nico lezione6 %
```

Sembra aver  
modificato  
solo il  
.gitignore



# Salvataggio della working directory con git stash

# git stash - casistiche di utilizzo

- Ci troviamo a lavorare su un branch ma ci accorgiamo che le modifiche devono essere effettuate su un altro branch;
- Quando stiamo lavorando su un branch e ci rendiamo conto che qualcosa è andato storto e il nostro codice rompe il funzionamento;
- Se dobbiamo nascondere temporaneamente il nostro codice per fare - ad esempio - dei test sul nostro branch “pulito”;

**git stash** serve a questo, “nasconde” il lavoro della nostra working directory. Il lavoro può essere ripristinato con il comando **git stash pop**

# git stash

```
nico@Mac-mini-di-Nico lezione6 % git status
On branch bug
Your branch is up to date with 'origin/bug'.

nothing to commit, working tree clean
nico@Mac-mini-di-Nico lezione6 % rm README.md
nico@Mac-mini-di-Nico lezione6 % git status
On branch bug
Your branch is up to date with 'origin/bug'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
nico@Mac-mini-di-Nico lezione6 % git stash ←
Saved working directory and index state WIP on bug: 219c00c Update README.md
nico@Mac-mini-di-Nico lezione6 %
```

# git stash pop

```
nico@Mac-mini-di-Nico lezione6 % git checkout conflict
Switched to branch 'conflict'
Your branch is up to date with 'origin/conflict'.
nico@Mac-mini-di-Nico lezione6 % git status
On branch conflict
Your branch is up to date with 'origin/conflict'.

nothing to commit, working tree clean
nico@Mac-mini-di-Nico lezione6 % git stash pop ←
CONFLICT (modify/delete): README.md deleted in Stashed changes and modified in Updated upstream. Version Updated upstream of README.md left in tree.
On branch conflict
Your branch is up to date with 'origin/conflict'.

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add/rm <file>..." as appropriate to mark resolution)
    deleted by them: README.md

no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.
```



Muoversi all'interno  
della history

# git log

```
nico@Mac-mini-di-Nico lezione6 % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
nico@Mac-mini-di-Nico lezione6 % git log
commit c28da760f747dbd869ed38729b2db63d5eb76026 (HEAD -> main, origin/test1, origin/test, origin/main, test1, test)
Merge: 19ef67a d4541a0
Author: nicolanoviell <166721952+nicolanoviell@users.noreply.github.com>
Date:   Sun Apr 28 18:46:26 2024 +0200

    Merge pull request #1 from nicolanoviell/bugfix/readmesbagliato

    Readme aggiornato
```

```
commit d4541a03fbe7c5d2f46be4c6275ac71e1ddcdf34
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 18:22:41 2024 +0200
```

```
    Readme aggiornato
```

Vedo la storia dei commit di un branch



# git checkout <hash>

```
nico@Mac-mini-di-Nico lezione6 % git checkout 19ef67a676c17889ceba...  
Note: switching to '19ef67a676c17889ceba...'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

```
HEAD is now at 19ef67a progetto iniziale  
nico@Mac-mini-di-Nico lezione6 % git status  
HEAD detached at 19ef67a
```

# Torniamo al branch principale

```
If you want to create a new branch to retain commits you create, you may  
do so (now or later) by using -c with the switch command. Example:
```

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

```
HEAD is now at 19ef67a progetto iniziale  
[nico@Mac-mini-di-Nico lezione6 % git status  
HEAD detached at 19ef67a  
nothing to commit, working tree clean  
[nico@Mac-mini-di-Nico lezione6 % git checkout main  
Previous HEAD position was 19ef67a progetto iniziale  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.  
nico@Mac-mini-di-Nico lezione6 %
```

# Cancellare o annullare i commit



git reset --hard

# Creo un nuovo commit

```
nico@Mac-mini-di-Nico lezione6 % vi README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "modifica al readme da annullare"
[conflict 06ae67a] modifica al readme da annullare
 1 file changed, 2 deletions(-)
nico@Mac-mini-di-Nico lezione6 % git log
commit 06ae67a6804d684686fabf8a02014cff92fdcaa1e (HEAD -> conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Mon Apr 29 08:57:02 2024 +0200

    modifica al readme da annullare

commit f20d38494011737aecf5cc7a6eed9624cb261161 (origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200
```

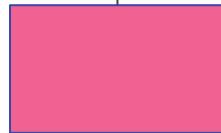
# Situazione attuale del Local Repository

**HEAD ->**

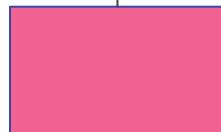


modifica al readme da annullare

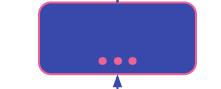
**remote**



rimossa dalla cache dir images



aggiunta la dir images su .gitignore

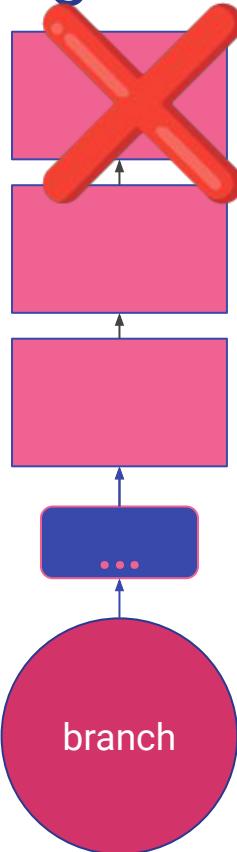


branch



# Cosa vogliamo fare

**HEAD ->**



modifica al readme da annullare

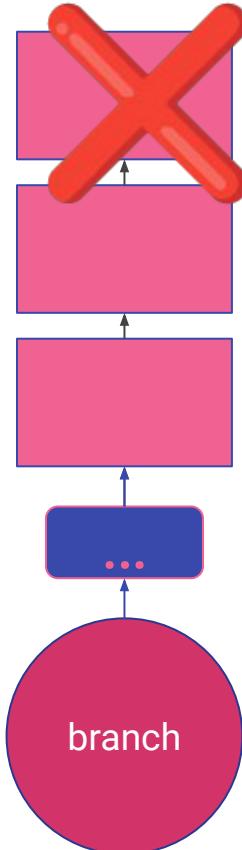
**remote**

rimossa dalla cache dir images

aggiunta la dir images su .gitignore

# Come lo facciamo?

**HEAD ->**



**remote**

modifica al readme da annullare

rimossa dalla cache dir images

aggiunta la dir images su .gitignore

`git reset --hard HEAD~1`

# git reset –hard

```
nico@Mac-mini-di-Nico lezione6 % git reset --hard HEAD~1
HEAD is now at f20d384 rimossa dalla cache dir images
nico@Mac-mini-di-Nico lezione6 % git log
commit f20d38494011737aecf5cc7a6eed9624cb261161 (HEAD → conflict, origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200

    aggiunta la dir images su .gitignore

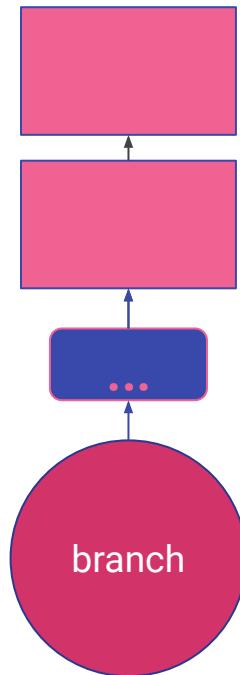
commit 384fa3cc520b442c43f901c431af5e916d96434f
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:06:45 2024 +0200

    aggiunta la dir images su .gitignore
```

l'opzione –hard di fatto “distrugge” il commit e le sue modifiche. Il codice ritorna alla versione precedente

# Situazione attuale del Local Repository

remote  
HEAD



rimossa dalla cache dir images

aggiunta la dir images su .gitignore



git reset --soft

# Creo un nuovo commit

```
nico@Mac-mini-di-Nico lezione6 % vi README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "modifica al readme da annullare - soft"
[conflict fdcd64d] modifica al readme da annullare - soft
 1 file changed, 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git log
commit fdcd64da28c6ca14ff6caf954b4e4d4b5c5b37d2 (HEAD -> conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Mon Apr 29 11:32:12 2024 +0200

    modifica al readme da annullare - soft

commit f20d38494011737aecf5cc7a6eed9624cb261161 (origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200
```

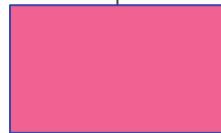
# Situazione attuale del Local Repository

**HEAD ->**

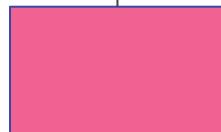


modifica al readme da annullare - soft

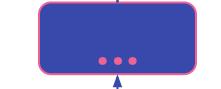
**remote**



rimossa dalla cache dir images



aggiunta la dir images su .gitignore



branch



# git reset –soft

```
nico@Mac-mini-di-Nico lezione6 % git reset --soft HEAD~1
nico@Mac-mini-di-Nico lezione6 % git log
commit f20d38494011737aecf5cc7a6eed9624cb261161 (HEAD -> conflict, origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200

    aggiunta la dir images su .gitignore

commit 384fa3cc520b442c43f901c431af5e916d96434f
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:06:45 2024 +0200

    aggiunta la dir images su .gitignore

commit 8761b230c967db206e453bbecd3012752f335cc2
```

l'opzione –soft di  
fatto “distrugge” il  
commit e le sue  
modifiche ma...

# git reset –soft

```
nico@Mac-mini-di-Nico lezione6 % git status
On branch conflict
Your branch is up to date with 'origin/conflict'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

nico@Mac-mini-di-Nico lezione6 % cat README.md
# lezione6
Aggiornamento Readme con conflitto
A new Flutter project.

## Getting Started

This project is a starting point for a Flutter application.

A few resources to get you started if this is your first Flutter project:

- [Lab: Write your first Flutter app](https://docs.flutter.dev/get-started/codelab)
- [Cookbook: Useful Flutter samples](https://docs.flutter.dev/cookbook)
```

... il file modificato  
rimane staged



git commit –amend

# Creo un nuovo commit

```
nico@Mac-mini-di-Nico lezione6 % vi README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "provo ad unire due commit"
[conflict 8e3987a] provo ad unire due commit
  1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git log
commit 8e3987af6aab1401e67b247b954f3b231da5984b (HEAD -> conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Mon Apr 29 11:46:13 2024 +0200

    provo ad unire due commit

commit f20d38494011737aacf5cc7a6eed9624cb261161 (origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200
```

# ...ma ricordo di aver dimenticato qualcosa

```
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit --amend
[CONFLICT 9518482] provo ad unire due commit
  Date: Mon Apr 29 11:46:13 2024 +0200
  1 file changed, 1 insertion(+), 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git log
commit 9518482bfbbdc8f500301a7b995c708d893455bc (HEAD -> conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Mon Apr 29 11:46:13 2024 +0200

  provo ad unire due commit

commit f20d38494011737aecf5cc7a6eed9624cb261161 (origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

  rimossa dalla cache dir images

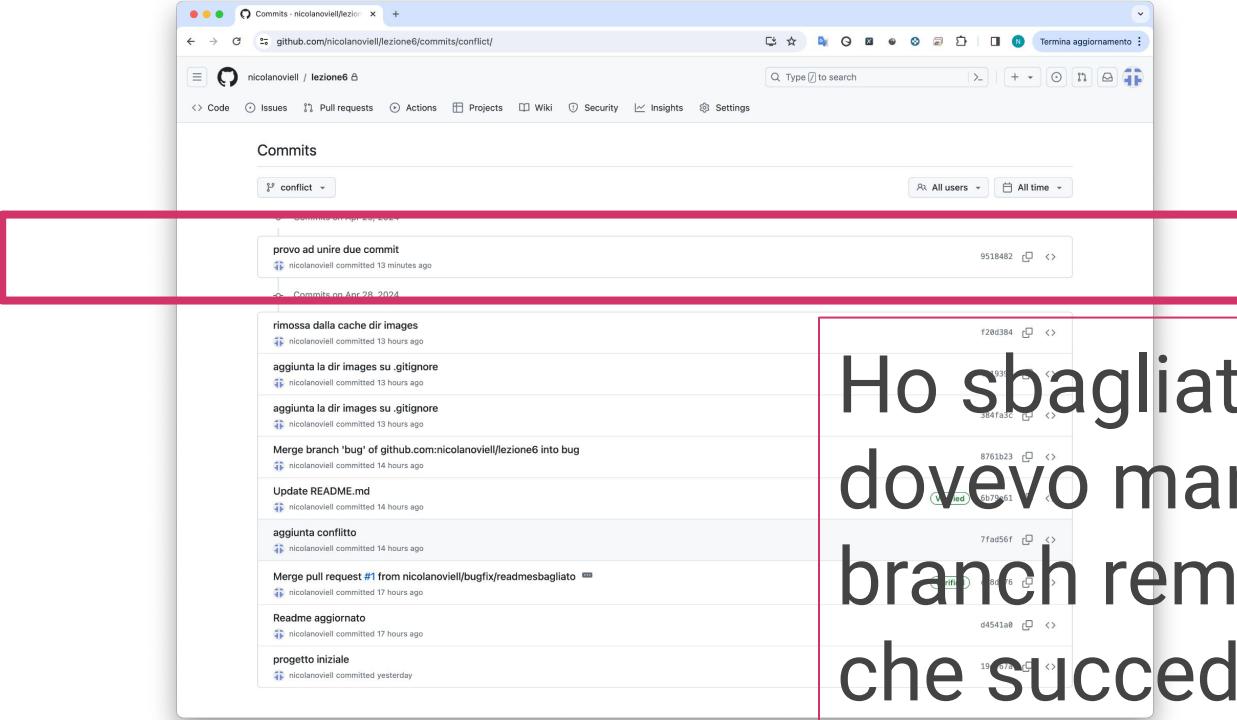
commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200
```

utilizzo –amend per  
unire il nuovo  
commit a quello  
precedente



# Reset su branch remoto

# Ho effettuato il push della precedente modifica



The screenshot shows a GitHub commit history for the repository `nicolanovali/lezione6`. The commits are listed in chronological order from top to bottom:

- provò ad unire due commit** (nicolanovali committed 13 minutes ago)
- rimossa dalla cache dir images** (nicolanovali committed 13 hours ago)
- aggiunta la dir images su .gitignore** (nicolanovali committed 13 hours ago)
- aggiunta la dir images su .gitignore** (nicolanovali committed 13 hours ago)
- Merge branch 'bug' of github.com:nicolanovali/lezione6 into bug** (nicolanovali committed 14 hours ago)
- Update README.md** (nicolanovali committed 14 hours ago)
- aggiunta conflitto** (nicolanovali committed 14 hours ago)
- Merge pull request #1 from nicolanovali/bugfix/readmesbagliato** (nicolanovali committed 17 hours ago)
- Readme aggiornato** (nicolanovali committed 17 hours ago)
- progetto iniziale** (nicolanovali committed yesterday)

Ho sbagliato, non  
dovevo mandarlo al  
branch remoto, ora  
che succede?

# git reset –hard

```
nico@Mac-mini-di-Nico lezione6 % git reset --hard HEAD~1
HEAD is now at f20d384 rimossa dalla cache dir images
nico@Mac-mini-di-Nico lezione6 % git push
To github.com:nicolanoviell/lezione6.git
 ! [rejected]      conflict -> conflict (non-fast-forward)
error: failed to push some refs to 'github.com:nicolanoviell/lezione6.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
nico@Mac-mini-di-Nico lezione6 %
```

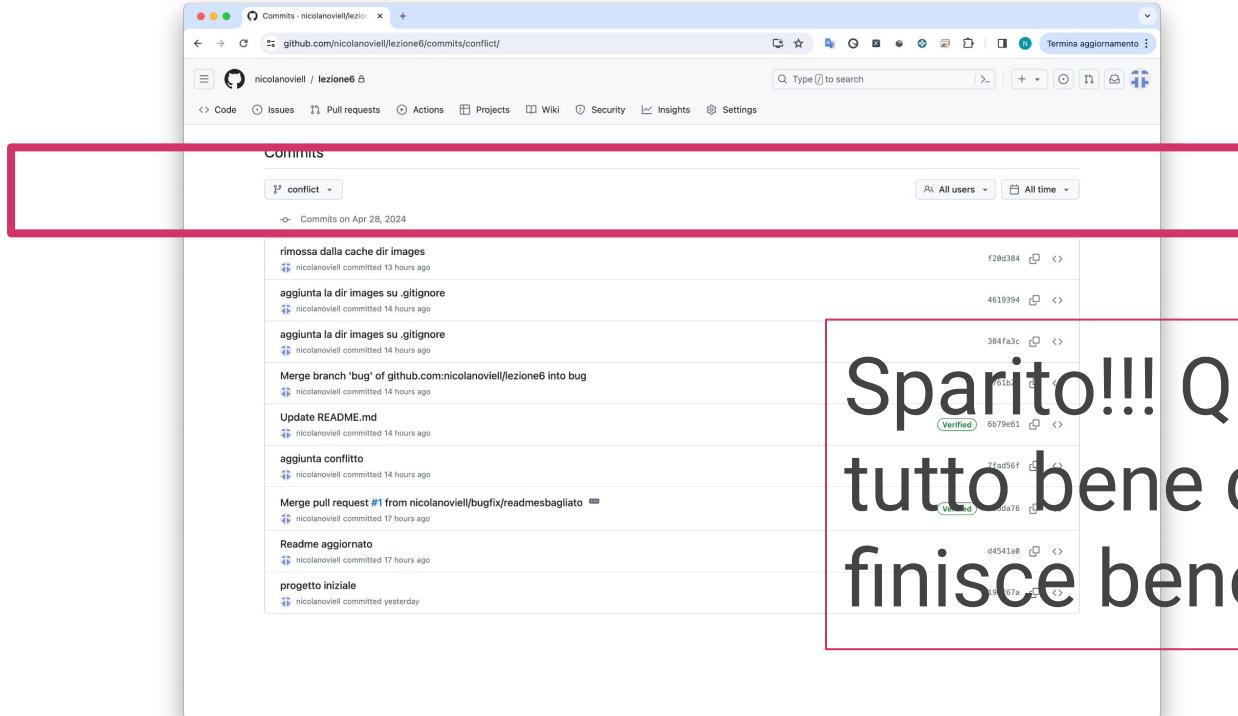
Non funziona! E ora  
che si fa?

# git push -force

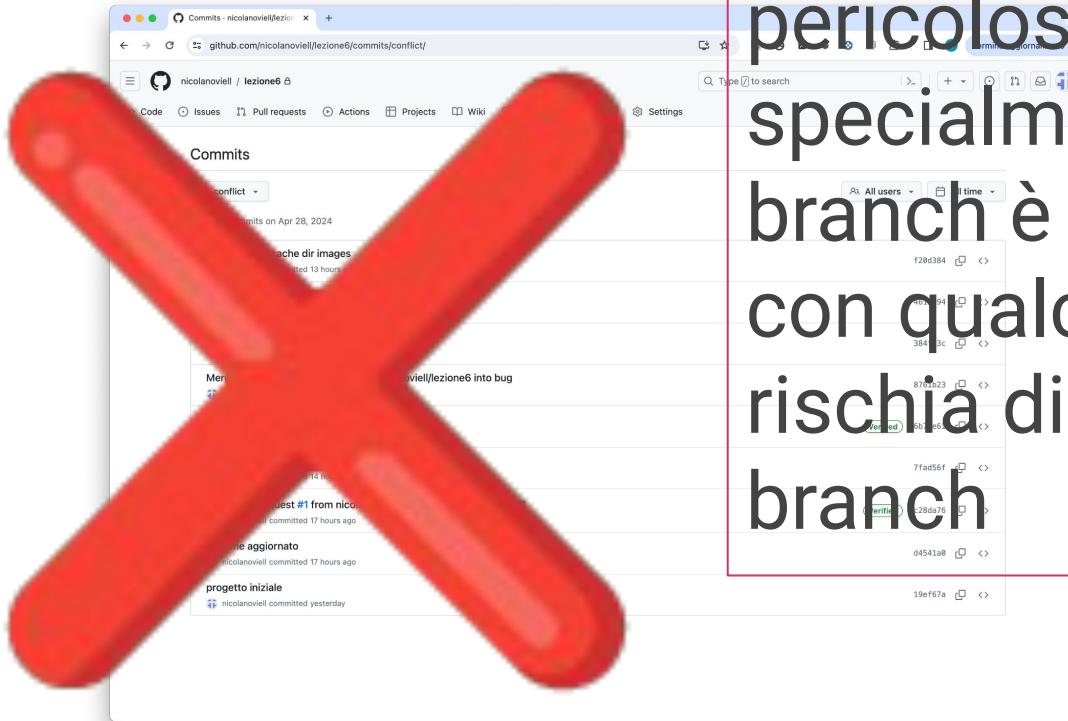
```
nico@Mac-mini-di-Nico lezione6 % git push --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:nicolanoviell/lezione6.git
 + 9518482...f20d384 conflict -> conflict (forced update)
nico@Mac-mini-di-Nico lezione6 %
```

Ha funzionato...

# Ho effettuato il push –force



# Ho effettuato il push –force



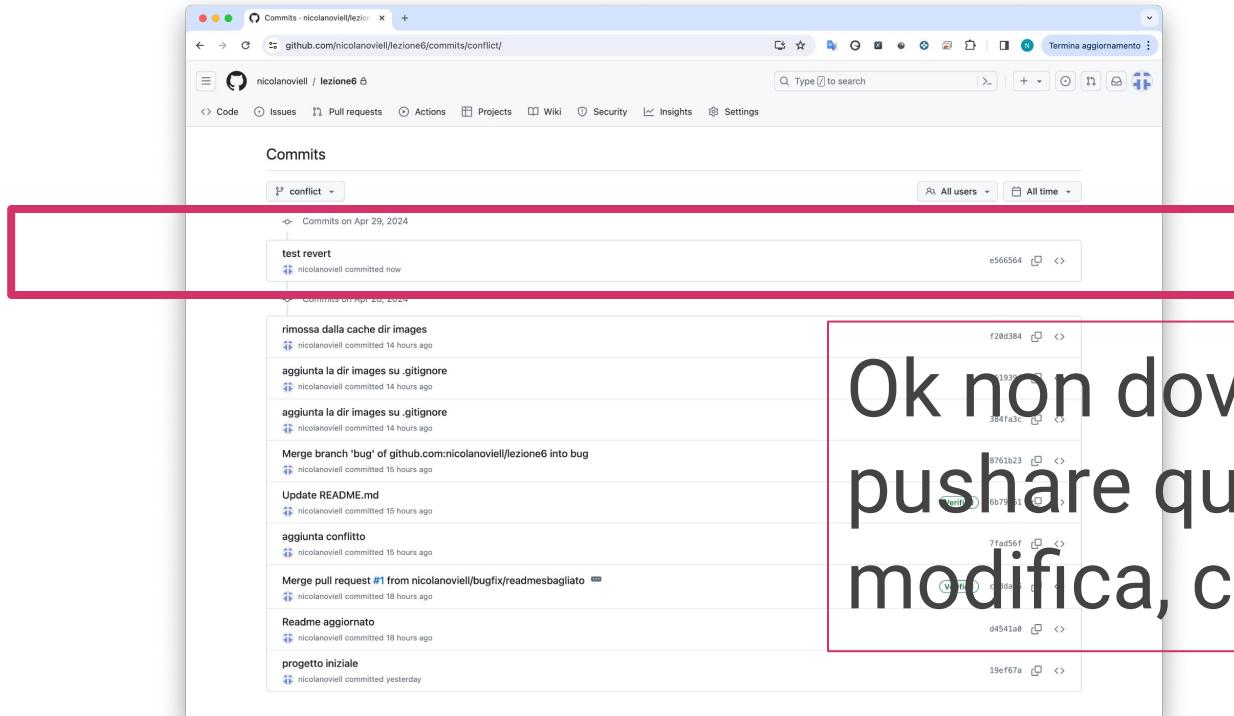
È estremamente pericoloso usarlo, specialmente se il branch è condiviso con qualcuno. si rischia di “rompere” il branch

# Revert su branch remoto

# Creo un nuovo commit

```
nico@Mac-mini-di-Nico lezione6 % vi README.md
nico@Mac-mini-di-Nico lezione6 % git add .
nico@Mac-mini-di-Nico lezione6 % git commit -m "test revert"
[conflict e566564] test revert
 1 file changed, 1 insertion(+)
nico@Mac-mini-di-Nico lezione6 % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:nicolanoviell/lezione6.git
  f20d384..e566564 conflict -> conflict
nico@Mac-mini-di-Nico lezione6 %
```

# Visualizzo sul GitHub il commit



# Acquisisco l'hash del commit con git log

```
lezione6 -- less - git log - 94x21
commit e56656436584f9ec10105e85f599c118843eb653 (HEAD -> conflict, origin/conflict)
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Mon Apr 29 12:41:46 2024 +0200

    test revert

commit f20d38494011737aecf5cc7a6eed9624cb261161
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:11:51 2024 +0200

    rimossa dalla cache dir images

commit 461939453e066b23493dbe0ec73ca88002929624
Author: Nicola Noviello <nicola.noviello@unimol.it>
Date:   Sun Apr 28 22:08:30 2024 +0200

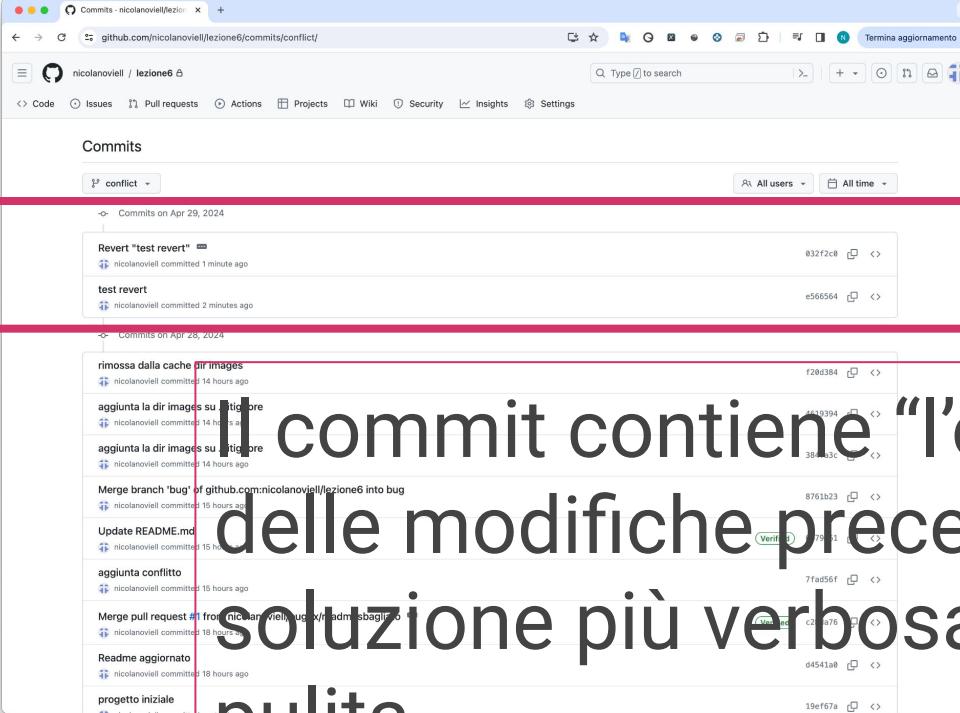
    aggiunta la dir images su .gitignore

commit 384fa3cc520b442c43f901c431af5e916d96434f
Author: Nicola Noviello <nicola.noviello@unimol.it>
:|
```

# Effettuo il revert e il push

```
nico@Mac-mini-di-Nico lezione6 % git revert e56656436584f9ec10105e85f599c118843eb653
[conflict 032f2c0] Revert "test revert"
 1 file changed, 1 deletion(-)
nico@Mac-mini-di-Nico lezione6 % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:nicolanoviell/lezione6.git
  e566564..032f2c0  conflict -> conflict
nico@Mac-mini-di-Nico lezione6 %
```

# Visualizzo sul GitHub il commit



The screenshot shows a GitHub commits page for the repository "nicolanolivelli/lezione6". A red box highlights the first commit in the list, which is a "Revert" commit. This commit contains the message "test revert" and is associated with two pushes: one from "nicolanolivelli" and another from "test revert". Below this, another commit is visible with the message "rimossa dalla cache la dir images". The main text area contains the following Italian text:

Il commit contiene “l’opposto” delle modifiche precedenti. È una soluzione più verbosa ma più pulita

# Esercizio

Provate le diverse soluzioni mostrare a lezione su dei progetti di TEST ma poi provate ad impostare su GitHub il progetto che porterete per l'esame. Potete anche aggiungermi sul progetto con il mio account GitHub @nicolanoviell