

Guarda as diferentes versões de um ficheiro como snapshots. Se não forem alterados, guarda um link para o snapshot anterior.

Tudo no git é guardado como um SHA-1.

Cada versão (commit) tem um snapshot da versão dos ficheiros (ou links caso não haja alterações). Todos os objetos (ficheiros, commits...) têm uma hash identificadora.

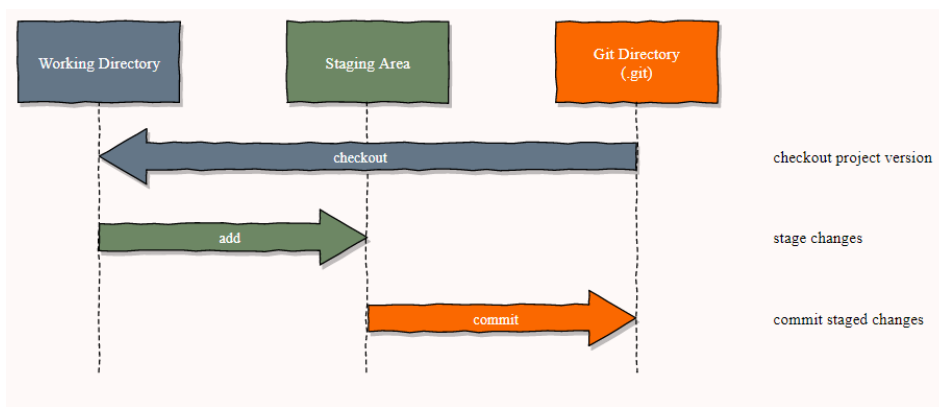


Figura 1: Git Areas

`git init`: cria um subdiretório `.git/` escondido

`git add ...`

- Dá track e stage a um ficheiro untracked pelo Git
- Dá stage a um ficheiro que foi modificado

`git commit`: grava um novo snapshot do repositório

`git status`: mostra o estado dos ficheiros

`git log`: mostra o histórico de commits do repositório

Um branch é um apontador para um commit. O branch inicial é o master. HEAD é um apontador especial que aponta sempre para o branch atual.

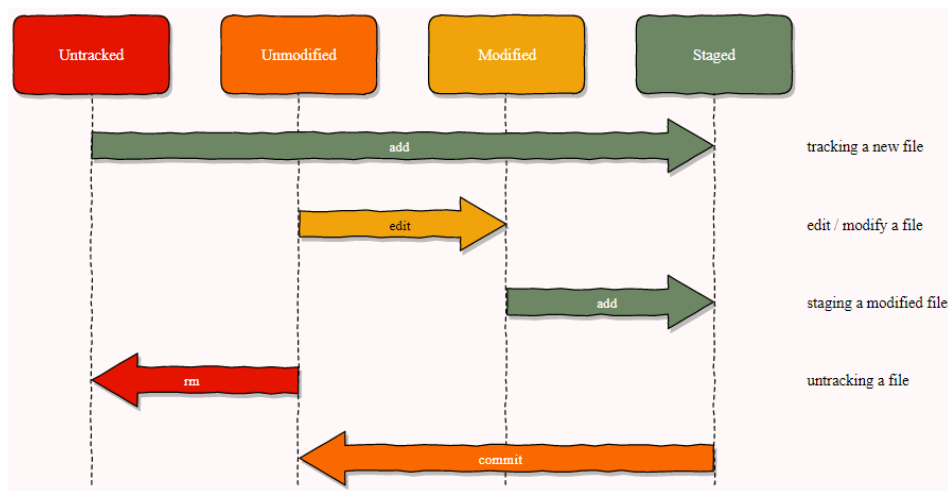


Figura 2: Estados dos Ficheiros

`git branch test`: cria um branch chamado *test*

`git branch`: mostra os branches atuais

`git checkout test`: alternar para o branch *test*

`git merge test`: junta ao branch atual as modificações feitas no branch *test*

Existem duas estratégias de merge:

- Fast-forward merge
 - Quando não existe trabalho divergente
 - Move-se o apontador para a frente
- Three-way merge
 - Quando existe trabalho divergente
 - Utiliza-se os snapshots dos dois branches e o antepassado comum e cria-se um novo commit

`git branch -d test`: apaga o branch *test*

Ficheiros *.gitignore* especificam ficheiros untracked que o git deve ignorar.

`git clone url`: clonar/criar repositório remoto (adiciona o origin)

- `git remote -v`: lista todos os repositórios remotos
- `git remote add nome url`: adicionar outro remoto

`git fetch`: download dos ficheiros do remoto

`git pull`: `git fetch` + `git merge`

`git push`: upload das modificações realizadas

`git reset`: reverter modificações

- `git revert <commit_id>`: caso já tenha sido realizado um push

`COMMIT^`: commit anterior a `COMMIT`

`COMMIT~2`: refere-se a 2 commits atrás de `COMMIT`, ou seja, ao commit anterior a `COMMIT^`