**Create Branch**

git branch <branch name>

**Lista de ramificações**

git branch

**Criar e fazer check-out de uma ramificação ao mesmo tempo**

git checkout -b <branch name>

**Checkout**

git checkout <branch name>

**Merge basic**

1 - git checkout master

2 - git merge <branch name>

**Delete branhc local - Não pode estar na branch que será excluida - faça chechout em outra**

git checkout <branch>

**Delete branch remoto**

git push <remote> --delete <branch>

**Quando houver problemas para excluir branch local, efetuar os passos abaixo:**

git checkout -b outra-branch (criando um nova branch que aponta para o commit que o HEAD está apontando)

git checkout master (voltando para a branch master)

git merge --no-ff outra-branch (trazendo as alterações da outra branch para o master)

git branch -d outra-branch (se a branch não for mais necessária, pode ser removida)

18

É essencialmente quase isso mesmo se você quer excluir o *commit* atual. Provavelmente o que deseja é voltar para o *commit* anterior, então precisa pegar o atual - 1:

git reset --hard HEAD~1

Se quer voltar para algum outro *commit* específico deve usar o *hash* dele. Ou em alguns casos pode fazer

git reset --hard HEAD^

Tente com --soft antes e veja se resolve o que precisa. Ele é menos radical e mais seguro.

Se o *commit* já tinha sido enviado terá que forçar no remoto:

git push origin HEAD --force

Isto fará perder todas as alterações locais. Então se não pode perdê-las, sugiro fazer um *stash* com todas elas antes.

Se der errado pode reverter assim:

git reset HEAD@{1}

Se está com medo de fazer algo errado, faça um *backup* antes e pode voltar ao original se não sair como esperava (em geral isso não deveria ser necessário porque o objetivo dessa ferramenta é justamente este, mas sabemos como é...).

Se quiser fazer no remoto:

git push origin +HEAD^:master #ou outro nome aqui