

Git Detached

Como vimos na última aula, o git permite ao usuário ir para um *commit* anterior sem estar em nenhum *branch*. Esse estado chama-se *detached*, onde se está em um *commit* que não está em *branch* algum.

Tarefas

1. Crie um repositório git numa pasta de trabalho à sua escolha.
2. Nesse repositório, crie um arquivo texto e armazene alguma informação nele. Faça um *commit* inicial.
3. Altere o arquivo e faça novo *commit*.
4. Novamente, altere o arquivo e faça um terceiro *commit*.
5. Execute o comando *git checkout <código do segundo commit>*. Isso fará com que o git retorne ao commit 2, mas mantendo o commit 3, diferentemente de quando executamos um *git reset* no qual o *commit 3* é perdido. Exemplo:

```
viniciusduarte (master) aulaDia_2016-03_02 $ git checkout
b3f2ffc2dce215359440dd6300e10269485fcc67
Note: checking out 'b3f2ffc2dce215359440dd6300e10269485fcc67'.
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 performing another
checkout.
If you want to create a new branch to retain commits you create,
you may
do so (now or later) by using -b with the checkout command again.
Example:
git checkout -b <new-branch-name>
HEAD is now at b3f2ffc... c2
```

6. Observe que o git deu uma mensagem, dizendo que ele está no *detached HEAD state*. Também informa que você pode fazer mudanças experimentais e pode fazer *commit* delas, sem impactar nenhum *branch* criado anteriormente (no nosso caso, sem impactar o *master*). Veja qual é o estado que está o arquivo que você criou, ele deve estar no estado do commit 2. Altere o arquivo e veja o status do *detached state*. Para isso, use o *git status*. Adicione o arquivo. Use o *git status*, para verificar. Faça um *commit*.
7. Para criar um novo branch, sigamos a instrução do próprio git (dada no exemplo do item 5). Usemos o comando que o git informou, o *git checkout -b <new-branch-name>*, para criar um *branch* a partir desse. O git automaticamente já vai para esse *branch* criado.
8. Use o *git log* para verificar os *commits*. Devemos ter 3 *commits*.
9. Vá para o *master* e faça um *merge* no master, com o novo *commit* que você criou através do *detached state*. Resolva os possíveis conflitos, neste caso, crie um novo *commit* que os resolve.

GitHub

Essa parte do roteiro depende muito da conexão com a internet do laboratório. Caso esteja muito ruim, faça em casa.

1. Acesse <https://github.com> para fazer o seu cadastro no site, caso você não tenha um ainda.
2. Crie um repositório no GitHub, clicando em *New Repository*.
3. Coloque um nome para o repositório. Crie-o.
4. O GitHub mostrará uma página de configuração do repositório.

5. Execute o comando de push. Deverá ser: *git remote add origin <endereço>*
6. Tente fazer o comando indicado abaixo do comando acima, *git push -u origin master*. Deverá aparecer um erro, indicando que você não tem permissão para fazer *push* no projeto.
7. É necessário gerar uma chave e adicioná-la no seu GitHub, para autenticação. Siga os passos descritos nos site do GitHub, qualquer problema, informe o professor. Site: <https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>
8. Após completar os passos, você será capaz de concluir com sucesso o comando *git push origin master*.