

**Explicação do problema da última aula, quando não conseguimos mudar de branch: quando o git deixa fazer checkout de branch e quando não deixa.**

O git não deixa fazer checkout de branch quando mudou-se um arquivo no diretório de trabalho (working directory) e não foi feito commit do arquivo antes da mudança. O git, quando faz checkout de branch, sempre atualiza o diretório de trabalho (working directory) para o estado do último commit do branch. Portanto, se tentamos mudar de branch com um arquivo alterado e não "comitado", mesmo que tenha sido adicionado na staging area, o git não deixa porque depois ele não saberá qual é o estado que o seu arquivo estava quando vc fez o checkout de branch, visto que ele sempre volta pro branch carregando as informações do último commit do branch que ele vai.

Lista de Comandos vistos até agora:

- git init
- git add arquivo1 arquivo2 ... arquivoN
- git add . (adiciona todos os arquivos do diretório atual)
- git commit -m "Mensagem"
- git log
- git show HEAD
- git show código-do-commit-
- git diff (Mostra a diferença entre o diretório de trabalho e o último commit)
- git checkout HEAD arquivo (Faz arquivo no diretório de trabalho voltar ao estado do último commit)
- git reset HEAD arquivo (Retira o arquivo da Staging Area, WD fica igual)
- git reset commit (Volta ao commit x, resetando tudo)
- git branch (Mostra os branches e marca o branch atual)
- git branch novobranch (Cria novo branch de nome "novobranch")
- git checkout novobranch (Muda para o branch "novobranch")

Novos comandos dessa aula:

- git commit -am "mensagem" (Adiciona na staging area e comita)
- git merge nome\_do\_branch (Faz o merge do branch que você está com o branch que você quer, "nome\_do\_branch")
- git branch -d nome\_do\_branch (Deleta o branch "nome\_do\_branch")

Faça:

1. Crie uma pasta para trabalharmos nessa aula.
2. Dentro dessa pasta, digite o comando `git clone https://gitlab.com/viniciusduarte/det476-roterio2.git`  
O git irá clonar um projeto criado no GitLab. Caso o código acima não funcione, tente executar esse: `git clone git@gitlab.com:viniciusduarte/det476-roterio2.git`
3. Verifique a situação do seu repositório local, que foi clonado, através do comando `git status`. O git deve informar que está no branch master e não há modificações do diretório de trabalho (working directory) e o último commit do seu repositório.
4. Abra o NetBeans.
5. Dentro do NetBeans, mande abrir o projeto que você baixou através do comando clone do repositório (ou seja, no repositório tinha um código que era um projeto Java do NetBeans). Vá até a pasta na qual você fez o git clone, estará lá o projeto.
6. Execute o código e veja o que o programa faz.

7. No terminal (Git Bash), crie um novo branch. Você fará uma nova funcionalidade no programa.
8. Mude para esse novo branch.
9. Adicione uma função que faça o programa não aceitar números negativos para a idade. O programa deve esperar até que o usuário digite um número negativo. Toda vez que o usuário digitar um número negativo, o programa avisa e pede novamente um número.
10. Após realizar a funcionalidade, adicione os arquivos na staging area e faça um commit, com uma boa mensagem de commit.
11. Mude para o branch master e faça o merge com o branch que você criou.
12. Apague o branch que você criou, após o merge ter sido feito com sucesso.
13. Após finalizar chame o professor, assine a chamada e pode ir embora =D