

# Encontrando branches - git fetch

terça-feira, 27 de junho de 2023

17:12

- As branches novas não são mapeadas pelo seu git porém com o comando **git fetch** consegue atualizar você das outras branches dos outros devs
- O comando é útil para atualizar a branch de outros devs de sua equipe
- Após usar o comando **git fetch** devemos usar o comando **git checkout nomebranch** para poder entrar e adicionar a branch ao nosso clone do projeto
- Outro dev entrar na nossa branch faz as modificações e envia, a partir disso é apenas usar o comando **git pull** para nos atualizar do que ele fez na nossa branch

## Encontrando a branch criada por outro dev

```
To https://github.com/Geronimonetto/repo_teste.git
* [new branch]      novabbranch -> novabbranch
branch 'novabbranch' set up to track 'origin/novabbranch'.

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste (novabbranch)
$ cd ..

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste (master)
$ cd repo_teste2

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ git branch
* main

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ git fecth
git: 'fecth' is not a git command. See 'git --help'.

The most similar command is
    fetch

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ git fetch
From https://github.com/Geronimonetto/repo_teste
* [new branch]      novabbranch -> origin/novabbranch
```

A branch foi criada por outro dev clonado do projeto, então devemos usar o comando git fetch para encontrar lá no repositório remoto essa nova branch

# Recebendo atualizações - git pull

quarta-feira, 28 de junho de 2023

14:47

- Deve-se usar o comando **git pull**
- Serve para receber atualizações de uma branch master ou de uma branch que outro dev modificou sua

# Enviando alterações - git push

quarta-feira, 28 de junho de 2023 14:49

- Usado para enviar modificações para o projeto
- Geralmente seguido do **git commit** ou **git add + git commit**
- Deve-se usar o comando **git push**

# Utilizando o remote - git remote

quarta-feira, 28 de junho de 2023

14:52

- Pode ser utilizado tanto para adicionar um repo remoto como remover também
- Quando mudamos o projeto de servidor, por exemplo do github para o bitbucket usamos o comando git remote para remover a origin e adicionar a nova url da origin em outro servidor
- Adicionando uma nova repo remota **git remote add origin <link>**
- Para remover basta usar **git remote rm origin**
- Para verificar qual a origem **git remote -v**

# Utilizando Submódulos - git submodules

quarta-feira, 28 de junho de 2023 15:05

- Deve-se usar o comando **git submodules add <link>**
- Serve como um submódulo de um projeto "Pai"
- Funciona como um repositório de um repositório
- Pode funcionar como diversas equipes utilizando os submódulos do projeto

```
Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ git submodule add https://github.com/Geronimonetto/submodulo.git
Cloning into 'D:/github_teste/repo_teste2/submodulo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
warning: in the working copy of '.gitmodules', LF will be replaced by CRLF the next time Git touches it

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ git submodule
070e9da898454adb6e26c94488da807a19b1899b submodulo (heads/main)

Geronimo Morais@DESKTOP-HQ9L9N0 MINGW64 /d/github_teste/repo_teste2 (main)
$ |
```

# Atualizando submodulos - `git push --recurse-submodules=on-demand`

quarta-feira, 28 de junho de 2023

15:10

- Devemos usar como se fosse um repositório normal, utilizando commits
- O comando a se utilizar para atualizar o submodulo é **`git push --recurse-submodules=on-demand`**