



# COMANDO BRANCH

Jhenifer Teixeira - 6 SI Tópicos especiais

# O QUE É O BRANCH?

Os branches são ramificações que podem ser usadas para trabalhar paralelamente no mesmo projeto. Pode ser usado para : criar, listar e excluir ramificações.

Não trabalhar apenas na **main**, ajuda a evitar problemas e ter um projeto mais flexível.



# BRANCH LOCAL X BRANCH REMOTO



O branch local são as ramificações do código que apenas um desenvolvedor tem acesso, ficando disponível **localmente em seu computador**.

O branch remoto, **coloca (hospeda) as ramificações em um local** que pode ser acessado por outros desenvolvedores. E com isso, é possível trabalhar em conjunto no mesmo projeto para facilitar o desenvolvimento.



## QUANDO USÁ-LO?

---

É útil para adicionar um novo recurso ou corrigir um erro, gerando uma nova ramificação sem interferir os arquivos do projeto principal.

Depois de concluir a atualização dos códigos da ramificação, você pode mesclar a ramificação com a principal, geralmente chamada de **master**.

# VANTAGENS DO SEU USO



É possível desenvolver novos recursos para nossa aplicação sem impedir o desenvolvimento no branch principal.



É possível criar diferentes branches de desenvolvimento que podem ter o mesmo repositório.

# MUDAR DE RAMIFICAÇÃO

O comando `checkout` pode ser utilizado para trocar de uma ramificação para outra.

```
$ git checkout nome_da_branch
```

```
Jhenifer Teixeira@LAPTOP-LO55A4CL MINGW64 ~/exemplos (okbranch)
$ git checkout branch_checkout
Switched to branch 'branch_checkout'
M      texto.txt
```

E com **um único comando** é possível criar uma nova branch e já entrar dentro dela , para isso após o checkout é preciso usar o **-b**

```
$ git checkout -b novo_nome_da_branch
```

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (branch_checkout)
$ git checkout -b novabranh
Switched to a new branch 'novabranh'

Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (novabranh)
$ git branch
  branch_checkout
  master
* novabranh
  okbranch
```

# LISTAR BRANCH

`$ git branch` (lista as ramificações presentes no repositório do seu computador)

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (okbranch)
$ git branch
  branch_checkout
  master
* okbranch
```

`$ git branch -a` (lista também as branches que estão no repositório remoto)

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (okbranch)
$ git branch -a
  branch_checkout
  master
* okbranch
```



# EXCLUIR BRANCH

`$ git branch -d` ou `--delete <nome_da_branch>`  
(deleta somente se já tiver feito merge ou enviado as alterações para o repositório remoto, evitando perda de código)

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (okbranch)
$ git branch -d testbranch
Deleted branch testbranch (was 0396dc4).
```

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (okbranch)
$ git branch
branch_checkout
master
* okbranch
```

# EXCLUIR BRANCH

`$ git branch -D <nome_da_branch>` (ignora o estado da sua branch, forçando a sua remoção)

```
Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (testbranch)
$ git checkout anovabranh
Switched to branch 'anovabranh'
M      texto.txt

Jhenifer Teixeira@LAPTOP-L055A4CL MINGW64 ~/exemplos (anovabranh)
$ git branch -D testbranch
Deleted branch testbranch (was 0396dc4).
```

# RENOMEAR BRANCH

`$ git branch -m <novo-nome-da-branch>`  
(Renomeia a branch local, através do comando `-m`)

```
Jhenifer Teixeira@LAPTOP-LO55A4CL MINGW64 ~/exemplos (novabranh)
$ git branch -m anovabranh

Jhenifer Teixeira@LAPTOP-LO55A4CL MINGW64 ~/exemplos (anovabranh)
$ |
```

# CONCLUSÃO

Usar os branches é importante para desenvolver uma aplicação em um jeito ordenado, **facilitando os desenvolvedores a testar novas funções isoladamente**, mantendo o programa estável.

