

1. Use git branch para ver os dois ramos que são relevantes para este exercício

```
C:\Users\luisg\git-katas\meu__remoto>git branch -r
origin/HEAD -> origin/main
origin/main
```

2. Em que ramo você está?

```
C:\Users\luisg\git-katas\meu__remoto>git branch
* main
```

3. Use git branch mybranch para criar uma nova ramificação chamada mybranch

```
C:\Users\luisg\git-katas\meu__remoto>git branch mybranch
```

4. Use git branch novamente para ver a nova ramificação criada.

```
C:\Users\luisg\git-katas\meu__remoto>git branch
* main
mybranch
```

5. Use git switch mybranch para alternar para sua nova filial.

```
C:\Users\luisg\git-katas\meu__remoto>git switch mybranch
Switched to branch 'mybranch'
```

6. Como a saída git status muda quando você alterna entre o master e o novo branch que você criou?

A saída do git status muda ao alternar entre branches porque reflete o estado atual do diretório de trabalho em relação ao branch para o qual você está trabalhando. As alterações feitas em um branch não afetam automaticamente outros branches, a

menos que você faça um merge ou use outros comandos para sincronizar as alterações entre eles.

7. Como o espaço de trabalho muda quando você alterna entre os dois ramos?

Ao alternar entre branches, o Git atualiza o seu espaço de trabalho para refletir o estado do branch para o qual você está mudando. Isso permite que você trabalhe em diferentes versões do seu projeto simultaneamente, mantendo cada uma organizada em sua própria branch.

8. Certifique-se de que você está no seu branch mybranch antes de continuar.

```
C:\Users\luisg\git-katas\meu__remoto>git switch mybranch
Already on 'mybranch'
```

9. Crie um arquivo chamado file1.txt com seu nome.

```
C:\Users\luisg\git-katas\meu__remoto>echo "Luís Gustavo" > file1.txt
```

10. Add o arquivo e commit com essa alteração.

```
C:\Users\luisg\git-katas\meu__remoto>echo "Luís Gustavo" > file1.txt
C:\Users\luisg\git-katas\meu__remoto>git add file1.txt
C:\Users\luisg\git-katas\meu__remoto>git commit -m "Adiciona arquivo file1.txt"
[mybranch eb84ab1] Adiciona arquivo file1.txt
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
```

11. Use git log --oneline --graph para ver sua ramificação apontando para o novo commit.

```
C:\Users\luisg\git-katas\meu__remoto>git commit -m "Adiciona arquivo file1.txt"
[mybranch 03803fb] Adiciona arquivo file1.txt
1 file changed, 1 deletion(-)
delete mode 100644 file1.txt
C:\Users\luisg\git-katas\meu__remoto>git log --oneline --graph
* 03803fb (HEAD -> mybranch) Adiciona arquivo file1.txt
* eb84ab1 Adiciona arquivo file1.txt
* 59bab61 (main) Initial commit
```

12. Retorne para o branch chamado master .

```
C:\Users\luisg\git-katas\meu_remoto>git switch master
fatal: invalid reference: master
```

13. Use git log --oneline --graph observe como o commit que você fez no branch mybranch está faltando no branch master .

```
C:\Users\luisg\git-katas\meu_remoto>git log --oneline --graph
* 03803fb (HEAD -> mybranch) Adiciona arquivo file1.txt
* eb84ab1 Adiciona arquivo file1.txt
* 59bab61 (main) Initial commit
```

14. Crie um novo arquivo chamado file2.txt e confirme esse arquivo.

```
C:\Users\luisg\git-katas\meu_remoto>echo "Conteúdo para o arquivo" > file2.txt
C:\Users\luisg\git-katas\meu_remoto>git add file2.txt
C:\Users\luisg\git-katas\meu_remoto>git commit -m "Adiciona arquivo file2.txt"
[mybranch ce3360a] Adiciona arquivo file2.txt
1 file changed, 1 insertion(+)
create mode 100644 file2.txt
```

15. Use git log --oneline --graph --all para ver sua ramificação apontando para o novo commit e que as duas ramificações agora têm commits diferentes.

```
C:\Users\luisg\git-katas\meu_remoto>git log --oneline --graph --all
* ce3360a (HEAD -> mybranch) Adiciona arquivo file2.txt
* 03803fb Adiciona arquivo file1.txt
* eb84ab1 Adiciona arquivo file1.txt
| * 9245ac9 (origin/main, origin/HEAD) Update README.md
|/
* 59bab61 (main) Initial commit
```

16. Mude para sua filial mybranch .

```
C:\Users\luisg\git-katas\meu_remoto>git switch mybranch
Already on 'mybranch'
```

17. O que aconteceu com seu diretório de trabalho? Você consegue ver seu file2.txt?

Quando você muda de branch no Git, o diretório de trabalho é atualizado para refletir o estado do branch para o qual você está mudando. Isso significa que os arquivos presentes no novo branch serão visíveis no seu diretório de trabalho, enquanto os arquivos que existem apenas no branch anterior podem desaparecer ou serem substituídos por versões diferentes. No meu caso ele não está mais visível.

18. Use `git diff mybranch master` para ver a diferença entre os dois ramos.

```
C:\Users\luisg\git-katas\meu_remoto>git diff mybranch master
fatal: ambiguous argument 'master': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...']
```