

ATIVIDADE: FF-Merge

1. Trabalho no Branch 'greeting'

```
git switch -c greeting  
echo "Olá, mundo!" > greeting.txt  
git add greeting.txt  
git commit -m "Atualiza saudação em greeting.txt"
```

2. Trabalho no Branch 'master'

```
git switch -c greeting  
echo "Este é um repositório para o kata 3-way merge" > README.md  
git add README.md  
git commit -m "Adiciona README.md"
```

3. A Divergência (1ª Pergunta)

Agora, os dois branches têm trabalhos únicos que o outro não tem.

- **Comando:**

```
git log --oneline --graph --all
```

- **What is the output?:** O log mostrará um histórico **divergente**. Os branches main e greeting se separaram de um commit pai em comum.

Gráfico:

```
* e8ab81b (HEAD -> main) Adiciona LER.md na main  
| * f8a022b (greeting) Adiciona README.md  
|/  
* 50e3a26 Primeiro Commit
```

4. O Merge (3-Way Merge)

Aqui é onde o Git une os dois históricos divergentes.

- **Comando (Diff):**

```
git diff master greeting
```

```
diff --git a/LER.md b/README.md  
similarity index 100%  
rename from LER.md  
rename to README.md
```

- **Comando (Merge):**

```
git merge greeting
```

ATIVIDADE: FF-Merge

5. O Resultado (2ª Pergunta)

- **Comando:**

`git log --oneline --graph --all`

- **(What is the output now?:** O log mostrará a **convergência**. Você verá o novo commit de merge no topo do main, unindo os dois históricos. Exatamente como o exercício mencionou, o commit "Merge branch 'greeting'".

Seu gráfico será algo assim:

```
* 7c95c66 (HEAD -> main) Merge branch 'greeting'
|\
| * fb9d478 (greeting) Commit geral em greeting
* | e7d1623 Commit geral
|/
| * f8a022b Adiciona README.md
* | e8ab81b Adiciona LER.md na main
|/
* 50e3a26 Primeiro Commit
```