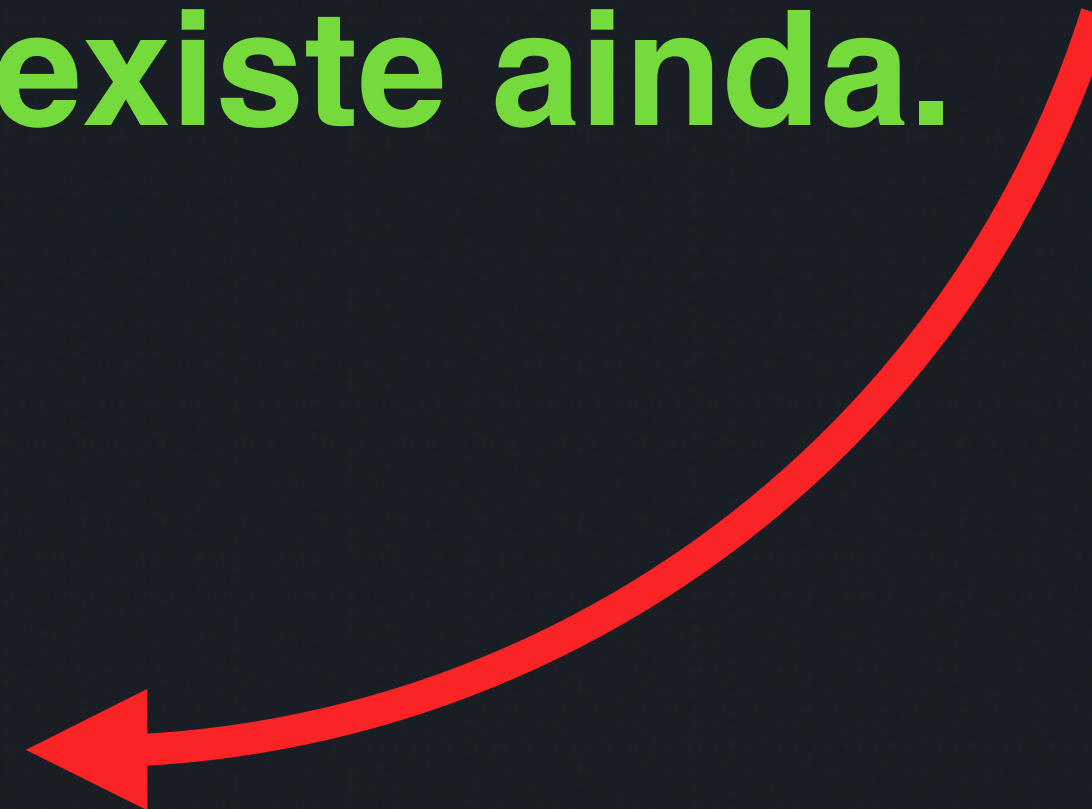


Git bisect

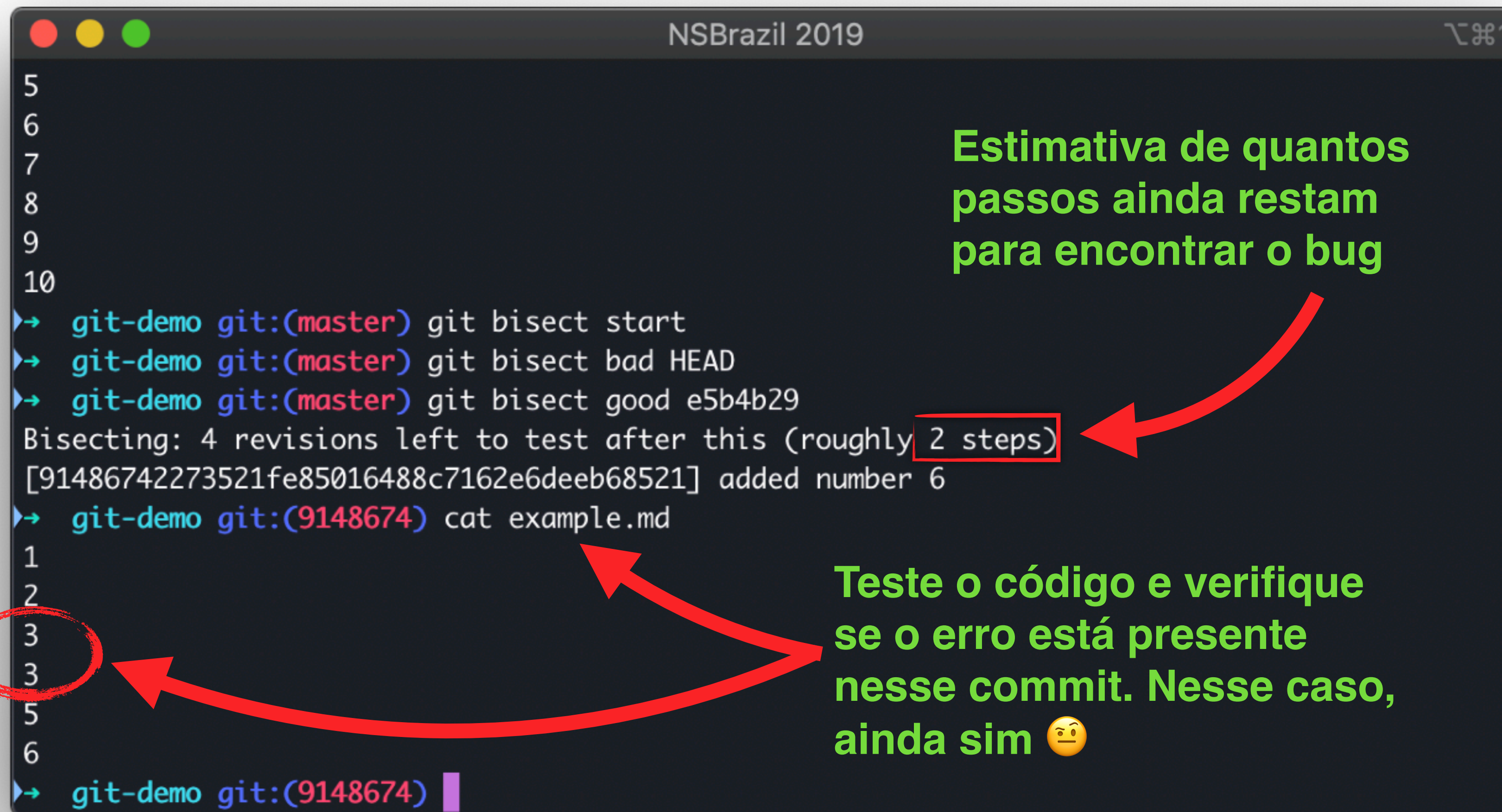
```
NSBrazil 2019
b4bef0c added number 2
e5b4b29 added number 1
efad026 initial commit
➤ git-demo git:(master) cat example.md
1
2
3
3
5
6
7
8
9
10
➤ git-demo git:(master) git bisect start
➤ git-demo git:(master) git bisect bad HEAD
➤ git-demo git:(master) git bisect good e5b4b29
Bisecting: 4 revisions left to test after this (roughly 2 steps)
```

Passo 3:
Dizer pro git,
onde o erro não
existe ainda.



Git bisect

A partir daí, o git vai te mover na timeline de commits, buscando o commit assassino!



The image shows a terminal window titled "NSBrazil 2019" with a dark background. The terminal output is as follows:

```
5  
6  
7  
8  
9  
10  
➤ git-demo git:(master) git bisect start  
➤ git-demo git:(master) git bisect bad HEAD  
➤ git-demo git:(master) git bisect good e5b4b29  
Bisecting: 4 revisions left to test after this (roughly 2 steps)  
[91486742273521fe85016488c7162e6deeb68521] added number 6  
➤ git-demo git:(9148674) cat example.md  
1  
2  
3  
3  
5  
6  
➤ git-demo git:(9148674) |
```

Annotations in green text with red arrows:

- Estimativa de quantos passos ainda restam para encontrar o bug**: An arrow points to the text "(roughly 2 steps)" in the terminal output, which is also enclosed in a red box.
- Teste o código e verifique se o erro está presente nesse commit. Nesse caso, ainda sim 🤔**: An arrow points to the terminal prompt "git-demo git:(9148674) cat example.md". The number "3" on the line immediately following the prompt is circled in red.