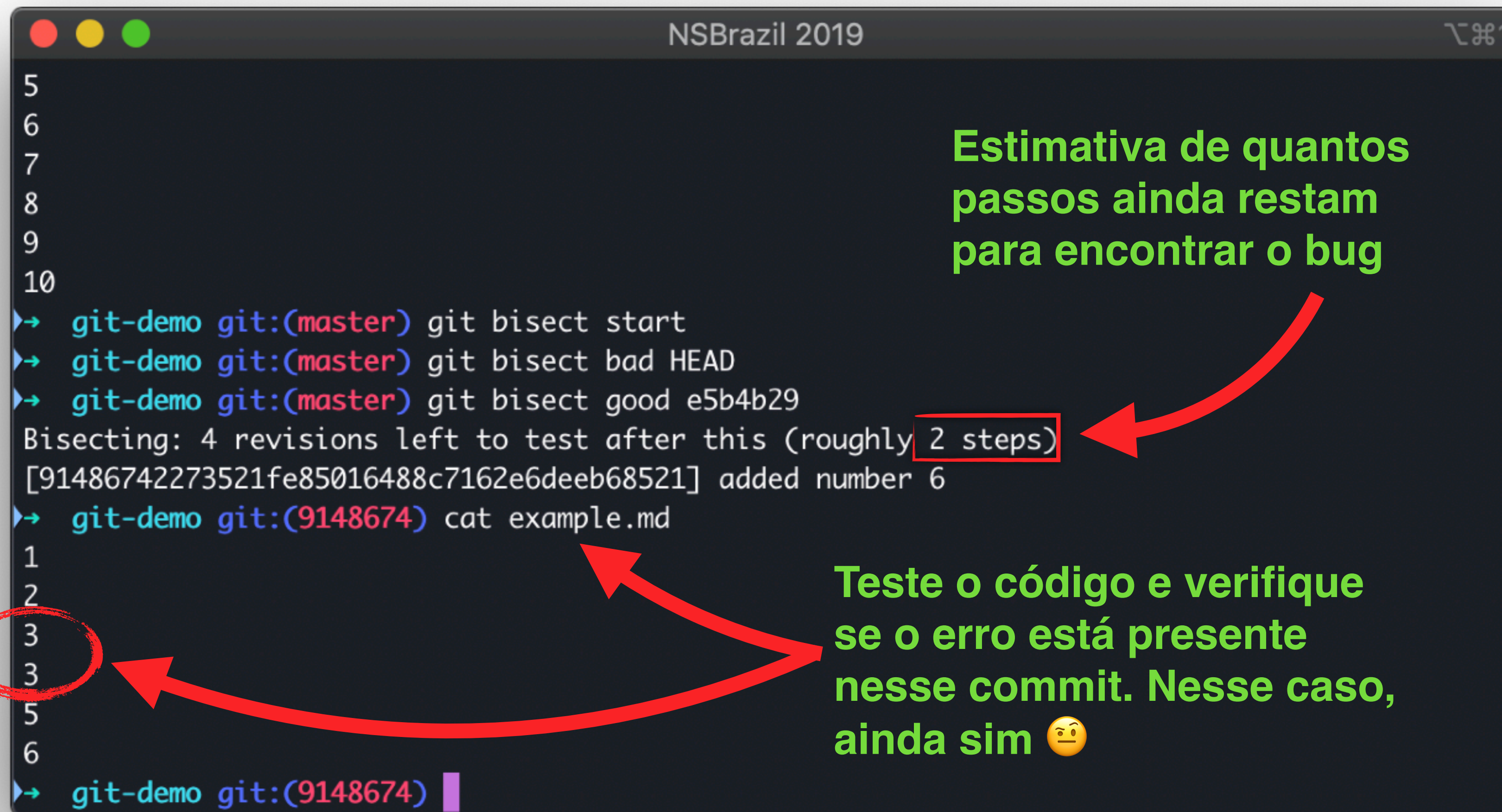


# Git bisect

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) |
```

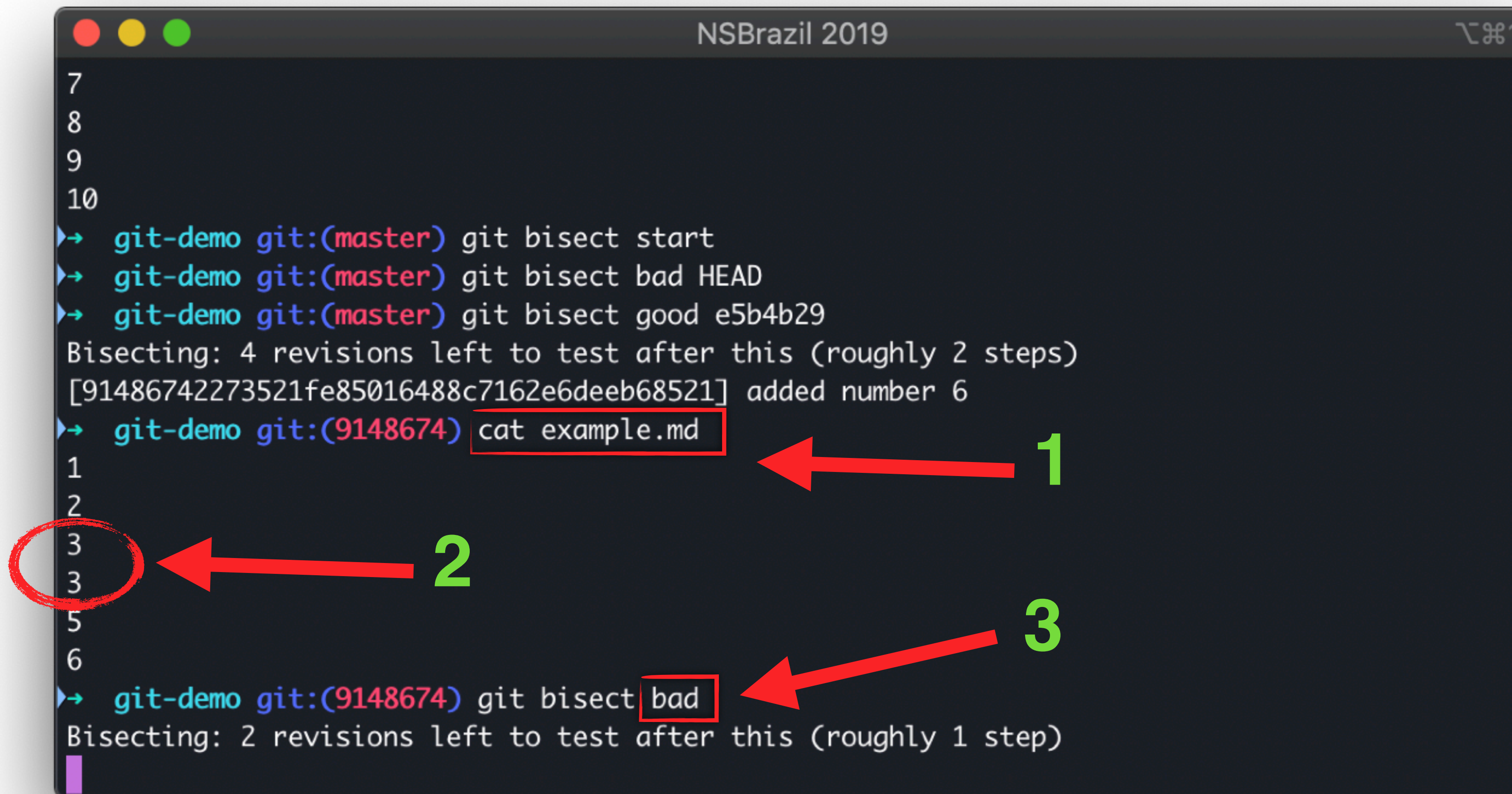
Annotations in green text with red arrows:

- Estimativa de quantos passos ainda restam para encontrar o bug**: Points to the text "(roughly 2 steps)" which is also enclosed in a red box.
- Teste o código e verifique se o erro está presente nesse commit. Nesse caso, ainda sim 🤔**: Points to the output "3" on the line following the command "cat example.md". This "3" is circled in red.



# Git bisect

Testamos o commit (1) verificamos que o erro existe (2), marcamos esse passo como **bad** (3).



```
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) git bisect bad
Bisecting: 2 revisions left to test after this (roughly 1 step)
```

The terminal window shows the steps of a Git bisect search. Three annotations are present: a green '1' with an arrow pointing to the `cat example.md` command; a green '2' with an arrow pointing to the second '3' in the file content list; and a green '3' with an arrow pointing to the `git bisect bad` command. The window title is 'NSBrazil 2019'.