

## Consolidando o seu conhecimento

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

---

1. Acesse a pasta do seu repositório no computador;
2. Execute `git checkout master` para trabalhar na *branch* `master`;
3. Execute `git pull origin master` e `git pull local master` para garantir que você está com o código atualizado e sincronizado com os dois repositórios remotos;
4. Execute `git checkout -b novo-release` para passar a trabalhar em uma nova *branch*, chamada `novo-release` ;
5. Substitua as tags `<h1>` por `<h2>` no código do arquivo **`index.html`**;
6. Confira o *hash* do *commit* que foi mostrado na saída do comando. Copie ele;
7. Execute `git add index.html` e `git commit -m "h1 -> h2"` para *commitar* esta alteração;
8. Altere o título do arquivo (na *tag* `<title>` ) para "Lista de cursos";
9. Execute `git add index.html` e `git commit -m "Simplificando o título"` para *commitar* esta alteração;
10. Execute `git checkout master` para voltar a trabalhar na *branch* `master` (pois há um *bug* que você deve corrigir antes de continuar o trabalho do seu novo *release*);
11. Execute o comando `git cherry-pick {hash}` para trazer um *commit* para a *branch* `master` . Substitua `{hash}` pelo *hash* que foi copiado no passo 6;
12. Confira que apenas a alteração desejada foi aplicada no `master` , e não toda a *branch* `novo-release` ;
13. Execute o comando `git rebase novo-release` e confira que o Git executa apenas um trabalho, embora hajam dois *commits* na nova *branch*, pois um já foi trazido para o `master` ;
14. Altere o título da página (na *tag* `<title>` ) para "Lista de cursos de DevOps";
15. Execute `git add index.html` e `git commit -m "Adicionando DevOps no título"` para *commitar* esta alteração;
16. Altere o título da página (na *tag* `<title>` ) para "Lista de cursos da Alura";
17. Execute `git add index.html` e `git commit -m "Mudando título"` para *commitar* esta alteração;
18. Altere o título da página (na *tag* `<title>` ) para "Cursos da Alura";
19. Execute `git add index.html` e `git commit -m "Mexendo no título"` para *commitar* esta alteração;
20. Altere o título do arquivo (na *tag* `<title>` ) para "Lista de cursos";
21. Execute `git add index.html` e `git commit -m "Título alterado"` para *commitar* esta alteração;
22. Execute o comando `git bisect start` para informar ao Git que você vai iniciar uma busca por determinada alteração;
23. Execute o comando `git bisect bad HEAD` para informar que o estado atual do código está "ruim", ou seja, o título não está no estado que você quer;
24. Executando `git log --oneline` , copie o *hash* do *commit* com a mensagem "Simplificando o título";
25. Execute o comando `git bisect good {hash}` , substituindo `{hash}` pelo *hash* copiado no passo anterior, para informar que o estado atual do código está "bom", ou seja, o título está no estado que você quer;
26. Confira no código que agora o título está como você quer;
27. Execute `git bisect good` para informar ao Git que neste *commit* o código ainda está como você quer;
28. Confira no código que agora o título não está mais como você quer;
29. Execute `git bisect bad` para informar ao Git que neste *commit* o código não está mais como você quer;
30. Note que o Git encontrou o exato *commit* onde o título deixou de estar no estado em que você quer;
31. Execute `git bisect reset` para indicar que você finalizou a busca;

32. Execute o comando `git show {hash}`, substituindo `{hash}` pelo *hash* conferido no passo 30. Veja que é exatamente a alteração que você estava buscando;
33. Para reverter esta alteração, execute `git revert {hash}`, substituindo `{hash}` pelo *hash* conferido no passo 30. Resolva quaisquer conflitos que possam ter sido encontrados;
34. Execute `git blame index.html` e confira o responsável por cada linha do arquivo.