

## 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) Execute o comando `git branch` e veja que apenas a `branch master` existe no seu repositório;
- 2) Execute o comando `git branch titulo` e logo após execute o comando `git branch`. Veja que uma nova `branch` foi criada;
- 3) Agora, para começar a trabalhar nesta `branch`, digite `git checkout titulo`;
- 4) Execute novamente `git branch` e confira que agora você está na `branch` chamado `titulo`;
- 5) Altere o título da página `index.html` para "Cursos de DevOps da Alura";
- 6) Adicione as alterações com `git add index.html`;
- 7) Faça o `commit`, com `git commit -m "Alterando título da página"`;
- 8) Execute o comando `git log` e confira o novo `commit`;
- 9) Altere o título da página para "Lista de cursos de DevOps da Alura";
- 10) Repita os passos 6 e 7, para adicionar um novo `commit`, alterando a mensagem;
- 11) Repita o passo 8 para conferir o novo `commit`;
- 12) Execute o comando `git checkout master` para voltar à linha de desenvolvimento `master`;
- 13) Execute `git log` para conferir que os últimos dois `commits` não estão lá. Confira se o conteúdo do seu arquivo também voltou ao seu estado original;
- 14) Na pasta criada para representar o trabalho de outra pessoa na aula anterior:
  - Execute `git checkout -b lista` para criar uma nova `branch`, chamada `lista` e passar a trabalhar nela;
  - Adicione o curso de "Kubernetes" na lista;
  - Repita os passos 6 e 7 para adicionar um novo `commit`, alterando a mensagem;
  - Execute o comando `git checkout master` para voltar à linha de desenvolvimento `master`;
- 15) Volte para a pasta que representa o seu próprio trabalho;
- 16) Altere o nome do curso de Docker para "Docker: Criando containers sem dor de cabeça";
- 17) Repita os passos 6 e 7 para adicionar um novo `commit`, alterando a mensagem;
- 18) Execute o comando `git merge titulo` para trazer o trabalho feito na `branch titulo` para a `branch master`;
- 19) Execute o comando `git log --graph` para ver as linhas de desenvolvimento (`branches`);
- 20) Execute `git checkout titulo` para trabalhar na `branch` chamada `titulo`;

- 21) Altere o título para ter a palavra "Cursos" com letra maiúscula;
- 22) Repita os passos 6 e 7 para adicionar um novo *commit*, alterando a mensagem;
- 23) Execute o comando `git checkout master` para voltar à linha de desenvolvimento `master`;
- 24) Execute o comando `git rebase titulo`;
- 25) Execute o comando `git log` e confira que o *commit* foi adicionado antes do *commit* realizado diretamente na *branch* `master`;
- 26) Execute o comando `git push local master` para enviar suas alterações para o repositório remoto que criamos na última aula;
- 27) Na pasta criada para representar o trabalho de outra pessoa na aula anterior:
  - Execute o comando `git pull local master` para baixar as alterações que você já realizou;
  - Execute o comando `git checkout lista` para continuar trabalhando na lista de cursos;
  - Altere o nome do curso de Docker para "Curso de Docker: Criando containers sem dor de cabeça";
  - Repita os passos 6 e 7 para adicionar um novo *commit*, alterando a mensagem;
  - Execute o comando `git checkout master` para voltar à linha de desenvolvimento `master`;
  - Tente juntar seu trabalho com `git merge lista`;
  - Veja que há conflitos. Corrija-os, deixando apenas a linha com o nome correto do curso;
  - Execute o comando `git add index.html` para informar que os conflitos neste arquivo foram corrigidos;
  - Execute o comando `git commit` para que o Git finalize o *merge*;
  - Execute o comando `git push local master` para enviar as suas alterações;
- 28) Volte para a pasta que representa o seu próprio trabalho;
- 29) Altere o nome do curso de Vagrant para "Vagrant: Gerenciando máquinas virtuais";
- 30) Repita os passos 6 e 7 para adicionar um novo *commit*, alterando a mensagem;
- 31) Tente executar o comando `git push local master .` Veja a falha;
- 32) Execute o comando `git pull local master` para trazer as alterações da outra pessoa;
- 33) Agora sim, execute o comando `git push local master` para enviar as alterações.