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

1. No repositório do GitHub que você criou utilizando o seu usuário no curso anterior, clique na aba **Issues**;
2. Clique em **New issue** para criar uma nova *issue* para o seu projeto;
3. Defina o título como "Adicionar título na página" e a descrição como "A página contém apenas a lista dos cursos. Senti falta de um título";
4. Clique em **Submit new issue** para criar a *issue* em questão;
5. Crie um novo usuário no GitHub, que servirá como um colaborador fictítcio (como o exemplo da **analura**);
6. Com este novo usuário, vá até o seu repositório original e clique em **Fork**, para criar uma cópia dele;
7. Em seu computador, acesse a pasta do projeto como Ana, e execute git pull origin master. Confira que não há este repositório remoto;
8. Adicione o repositório *fork* ao projeto da Ana, utilizando git remote add origin {url\_do\_repositorio};
9. Execute git pull origin master para trazer todas as alterações para o projeto local da Ana;
10. Logo após a abertura da tag <body> no HTML, adicione <h1>Cursos de DevOps</h1>;
11. Execute o comando git config --local user.email "{email}", substituindo {email} pelo e-mail do usuário fictício que você criou no GitHub, para que o autor dos *commits* esteja correto;
12. Execute git add index.html e depois git commit -m "Título adicionado", para realizar o *commit* desta alteração;
13. Adicione uma quebra de linha logo após o título recém adicionado;
14. Execute git add index.html e depois git commit -m "Quebra de linha" para realizar o *commit* desta alteração;
15. Execute git push origin master para enviar as alterações para o GitHub do usuário fictício;

15.1. Se estiver utilizando Windows, limpe as credenciais do GitHub

15.2. Vá em **Painel de Controle --> Contas de Usuário --> Gerenciador de Credenciais --> Credenciais do Windows**;

15.3. Encontre a credencial do GitHub e clique em **Remover**, para excluí-la;

15.4. Execute git push origin master para, agora sim, enviar as alterações para o GitHub do usuário fictício;

1. Acesse o repositório cópia do seu usuário fictício;
2. Confira que há alterações que não estão presentes no repositório original;
3. Clique em **New pull request** para criar o pedido de envio de alterações do código original;
4. Analise as alterações, veja a lista de *commits*, e depois clique em **Create pull request** para confirmar a criação;
5. Defina "Adicionando título na página" como título do *pull request* e "Adicionando um título na página que antes não tinha." como descrição;
6. Clique em **Create pull request** para confirmar todos os dados;
7. Entre novamente como o seu usuário original do GitHub e confira a aba **Pull requests** em seu repositório;
8. Clique em **Merge pull request** e depois em **Confirm merge** para trazer este *pull request* para o seu repositório original;
9. Confira no GitHub que o código no repositório original foi atualizado com as alterações do usuário fictício;
10. Acesse, na aba **Issues**, a *issue* que você criou para adicionar título;
11. Digite na caixa de comentários "Fechado pelo PR #". Note que abrirá uma caixa mostrando o título do *pull request*. Clique nela;
12. Clique no botão **Close and comment**;
13. Acesse novamente o GitHub, utilizando o usuário fictício criado no passo 5;
14. No repositório do seu usuário original, crie uma nova *issue* com o título "Trocar ul por dl";
15. Em seu computador, no repositório da Ana, altere as tags <ul> por <dl> e as tags <li> por <dd>;
16. Execute git add index.html e depois git commit -m "Trocando UL por DL" para realizar o *commit* desta alteração;
17. Separe os títulos das descrições dos cursos, deixando o conteúdo da seguinte forma:
18. <dl>
19. <dt>Vagrant</dt>
20. <dd>Gerenciando máquinas virtuais</dd>
21. <dt>Docker</dt>
22. <dd>Criando containers sem dor de cabeça</dd>
23. <dt>Ansible</dt>
24. <dd>Sua infraestrutura como código</dd>
25. <dt>Integração Contínua</dt>
26. <dd>Maturidade e Produtividade no Desenvolvimento de Software</dd>
27. <dt>Kubernetes</dt>
28. <dd>Introdução a orquestração de containers</dd>

</dl>COPIAR CÓDIGO

1. Execute git add index.html e depois git commit -m "Separando os títulos", para realizar o *commit* desta alteração;
2. Adicione uma quebra de linha antes de cada título (<dt>);
3. Execute git add index.html e depois git commit -m "Quebras de linha" para realizar o *commit* desta alteração;
4. Execute git log --oneline para verificar que há três novos *commits* para a última *issue* criada;
5. Execute git rebase -i HEAD~3 para analisar quais dos três últimos *commits* serão mantidos ou unidos ao anterior;
6. Troque a palavra **pick**, nas linhas 2 e 3, por um **s** (de *squash*) para que ambos sejam unidos ao *commit* anterior, da primeira linha;

38.1. Se o editor de texto que for aberto for o **Vim**, para excluir uma palavra, digite dw. Para começar a digitar, aperte i, e então você pode adicionar o s na linha. Para salvar, aperte Esc e depois :x e Enter;

1. Na nova tela que abrir, altere a mensagem que será atribuida ao novo *commit*, para algo como "Trocando UL por DL e separando os títulos";
2. Execute git log --oneline novamente e confira que agora os três *commits* viraram apenas 1;
3. Repita os passos do 15 ao 27, trocando as mensagens e a *issue* pelo que fizer sentido para seu caso.