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

1. Execute git log --graph e confira que, embora haja várias linhas de desenvolvimento, tudo começa e termina na linha da *branch* master;
2. Execute o comando git branch -d titulo para remover a *branch* titulo;
3. Execute o comando git branch -D novo-release para remover a *branch* novo-release, mesmo que ela tenha *commits* à frente do master;
4. Execute o comando git branch development para criar um nova *branch* de desenvolvimento chamado development;
5. Execute o comando git checkout development para passar a trabalhar na *branch* development;
6. Execute o comando git checkout -b feature/lista-cursos-cloud para criar um nova *branch*, chamada feature/lista-cursos-cloud;
7. Adicione o seguinte conteúdo após a lista de cursos de DevOps:
8. <h2>Cursos de Cloud:</h2>
9. <dl>
10. <dt>Amazon EC2</dt>
11. <dd>Faça um deploy da sua webapp com alta disponibilidade e escalabilidade.</dd>
12. <dt>Amazon S3</dt>
13. <dd>Manipule e armazene objetos na nuvem</dd>

</dl>COPIAR CÓDIGO

1. Execute git add index.html e git commit -m "Cursos de EC2 e S3 adicionados" para *commitar* esta alteração;
2. Adicione a esta lista os dois seguintes cursos:
3. <dt>Amazon Elastic Beanstalk Parte 1</dt>
4. <dd>Container Docker</dd>
5. <dt>Amazon ECS</dt>

<dd>Gerencie Docker na nuvem da AWS</dd>COPIAR CÓDIGO

1. Execute git add index.html e git commit -m "Cursos de Beanstalk e ECS adicionados" para *commitar* esta alteração;
2. Execute git checkout development para voltar para a *branch* de desenvolvimento;
3. Execute git checkout -b feature/lista-cursos-linux para passar a trabalhar em um nova *branch*, chamada feature/lista-cursos-linux;
4. Adicione o seguinte conteúdo após a lista de cursos de DevOps:
5. <h2>Cursos de Linux:</h2>
6. <dl>
7. <dt>Linux I</dt>
8. <dd>Conhecendo e utilizando o terminal</dd>
9. <dt>Linux II</dt>
10. <dd>Programas, processos e pacotes</dd>

</dl>COPIAR CÓDIGO

1. Execute git add index.html e git commit -m "Cursos de Linux I e II adicionados" para *commitar* esta alteração;
2. Execute git checkout development para voltar para a *branch* de desenvolvimento;
3. Execute o comando git branch para conferir quantas e quais *branches* você têm criadas atualmente;
4. Execute o comando git merge feature/lista-cursos-linux para trazer os cursos de Linux para a *branch* de desenvolvimento;
5. Execute o comando git checkout feature/lista-cursos-cloud para voltar à *branch* feature/lista-cursos-cloud;
6. Adicione, à lista de cursos de Cloud, o seguinte curso:
7. <dt>Google Cloud</dt>

<dd>Deploy de uma aplicação em Spring MVC</dd>COPIAR CÓDIGO

1. Execute git add index.html e git commit -m "Curso de Google Cloud adicionado" para *commitar* esta alteração;
2. Execute git checkout development para voltar para a *branch* de desenvolvimento;
3. Execute o comando git merge feature/lista-cursos-cloud para trazer os cursos de Cloud para a *branch* de desenvolvimento;
4. Resolva o conflito mantendo as duas listas novas de cursos;
5. Execute git add index.html e git commit para continuar o processo de *merge*;
6. Volte para a *branch* master com o comando git checkout master;
7. Trabalhe em um nova *branch*, chamada hotfix/v0.1.1, utilizando o comando git checkout -b hotfix/v0.1.1;
8. Altere o título da página (na *tag* <title>) para "Lista de cursos";
9. Execute git add index.html e git commit -m "Removendo o nome da Alura do título" para *commitar* esta alteração;
10. Volte para a *branch* master, com o comando git checkout master;
11. Una o trabalho do *hotfix*, com o comando git merge hotfix/v0.1.1;
12. Crie uma nova *tag* que representa uma nova versão, utilizando o comando git tag -a v0.1.1 -m "Versão com correção no título";
13. Vá para a *branch* de desenvolvimento com git checkout development;
14. Traga a correção do *bug* com git merge hotfix/v0.1.1;
15. Crie um nova *branch*, chamada de release/v0.2.0, com o comando git checkout -b release/v0.2.0;
16. Adicione os dois seguintes cursos na lista de cursos de Linux:
17. <dt>Shell Script I</dt>

<dd>Começando seus scripts de automação de tarefas</dd>COPIAR CÓDIGO

<dt>Shell Script II</dt>

<dd>Fazendo monitoramento, agendando tarefas e backup</dd>COPIAR CÓDIGO

1. Execute git add index.html e git commit -m "Corrigindo bug: Cursos de Shell faltantes" para *commitar* esta alteração;
2. Volte para a *branch* master, com o comando git checkout master;
3. Una todo o trabalho referente à nova *release*, com git merge release/v0.2.0;
4. Crie uma nova *tag* que representa uma nova versão, utilizando o comando git tag -a v0.2.0 -m "Novas listas adicionadas";
5. Execute o comando git branch e confira todos as *branches* criadas neste processo;