

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 `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:

```
<h2>Cursos de Cloud:</h2>
<dl>
  <dt>Amazon EC2</dt>
  <dd>Faça um deploy da sua webapp com alta disponibilidade e escalabilidade.</dd>

  <dt>Amazon S3</dt>
  <dd>Manipule e armazene objetos na nuvem</dd>
</dl>
```

8. Execute `git add index.html` e `git commit -m "Cursos de EC2 e S3 adicionados"` para `commitar` esta alteração;
9. Adicione a esta lista os dois seguintes cursos:

```
<dt>Amazon Elastic Beanstalk Parte 1</dt>
<dd>Container Docker</dd>

<dt>Amazon ECS</dt>
<dd>Gerencie Docker na nuvem da AWS</dd>
```

10. Execute `git add index.html` e `git commit -m "Cursos de Beanstalk e ECS adicionados"` para `commitar` esta alteração;
11. Execute `git checkout development` para voltar para a `branch` de desenvolvimento;
12. Execute `git checkout -b feature/lista-cursos-linux` para passar a trabalhar em um nova `branch`, chamada `feature/lista-cursos-linux`;
13. Adicione o seguinte conteúdo após a lista de cursos de DevOps:

```
<h2>Cursos de Linux:</h2>
<dl>
  <dt>Linux I</dt>
  <dd>Conhecendo e utilizando o terminal</dd>

  <dt>Linux II</dt>
```

<dd>Programas, processos e pacotes**</dd>**
</dl>

14. Execute `git add index.html` e `git commit -m "Cursos de Linux I e II adicionados"` para *commitar* esta alteração;
15. Execute `git checkout development` para voltar para a *branch* de desenvolvimento;
16. Execute o comando `git branch` para conferir quantas e quais *branches* você têm criadas atualmente;
17. Execute o comando `git merge feature/lista-cursos-linux` para trazer os cursos de Linux para a *branch* de desenvolvimento;
18. Execute o comando `git checkout feature/lista-cursos-cloud` para voltar à *branch* `feature/lista-cursos-cloud`;
19. Adicione, à lista de cursos de Cloud, o seguinte curso:

<dt>Google Cloud**</dt>**
<dd>Deploy de uma aplicação em Spring MVC**</dd>**

20. Execute `git add index.html` e `git commit -m "Curso de Google Cloud adicionado"` para *commitar* esta alteração;
21. Execute `git checkout development` para voltar para a *branch* de desenvolvimento;
22. Execute o comando `git merge feature/lista-cursos-cloud` para trazer os cursos de Cloud para a *branch* de desenvolvimento;
23. Resolva o conflito mantendo as duas listas novas de cursos;
24. Execute `git add index.html` e `git commit` para continuar o processo de *merge*;
25. Volte para a *branch* `master` com o comando `git checkout master`;
26. Trabalhe em um nova *branch*, chamada `hotfix/v0.1.1`, utilizando o comando `git checkout -b hotfix/v0.1.1`;
27. Altere o título da página (na tag `<title>`) para "Lista de cursos";
28. Execute `git add index.html` e `git commit -m "Removendo o nome da Alura do título"` para *commitar* esta alteração;
29. Volte para a *branch* `master`, com o comando `git checkout master`;
30. Una o trabalho do *hotfix*, com o comando `git merge hotfix/v0.1.1`;
31. Crie uma nova *tag* que representa uma nova versão, utilizando o comando `git tag -a v0.1.1 -m "Versão com correção no título"`;
32. Vá para a *branch* de desenvolvimento com `git checkout development`;
33. Traga a correção do *bug* com `git merge hotfix/v0.1.1`;
34. Crie um nova *branch*, chamada de `release/v0.2.0`, com o comando `git checkout -b release/v0.2.0`;
35. Adicione os dois seguintes cursos na lista de cursos de Linux:

<dt>Shell Script I**</dt>**
<dd>Começando seus scripts de automação de tarefas**</dd>**

<dt>Shell Script II**</dt>**
<dd>Fazendo monitoramento, agendando tarefas e backup**</dd>**

36. Execute `git add index.html` e `git commit -m "Corrigindo bug: Cursos de Shell faltantes"` para *commitar* esta alteração;
37. Volte para a *branch* `master`, com o comando `git checkout master`;
38. Una todo o trabalho referente à nova *release*, com `git merge release/v0.2.0`;

39. Crie uma nova *tag* que representa uma nova versão, utilizando o comando `git tag -a v0.2.0 -m "Novas listas adicionadas"`;

40. Execute o comando `git branch` e confira todos as *branches* criadas neste processo;