

Fábrica de Software

Documento: fs-mobile-2

Git (exercícios adicionais)

1. Siga atentamente os *slides* disponíveis [AQUI](#). Experimente os comandos, ambiente-se com eles. Esclareça suas dúvidas com o professor. Faça uso de repositórios locais e, para boa parte dos comandos, terá que interagir com o GitHub (serviço sugerido na disciplina). Faça isto antes de prosseguir com os demais itens.

2. Qual o comando para que as “marcas” ou *tags* sejam enviadas para o repositório remoto? (um simples **git push** não produz este efeito)

`git push origin <tag_name>` para uma tag específica

ou

`git push --tags` para todas as tags

3. Qual o nome do *branch* padrão do Git?

Master

4. O que o comando **git branch <branchname>** realiza?

Cria uma nova branch com o nome especificado em <branchname>

5. Como criar um *branch* a partir de um *commit* específico?

`git branch branchname <sha1-of-commit>`

6. Em um repositório, qual o efeito do comando **git branch erro1234**?

É criada uma branch com o nome erro1234

7. Qual o comando para se alternar para um *branch* de nome **experimento2**?

`git checkout experimento2`

8. Em um repositório com dois *branches*, **b1** e **b2**, onde **b1** é o corrente, qual o efeito do comando **git branch**?

Lista todos os branches de um repositório e mostra o branch corrente de maneira destacada.

9. O que o comando **git checkout -b novobranch** faz?

Cria e faz o checkout do novo branch

10. Qual a função do comando **git branch -d teste**?

Deleta um branch específico, no caso do exemplo, um branch chamado "teste".

Este é o modo seguro pra excluir um branch, ele previne que você exclua o branch se houver modificações unmerged.

11. Durante o desenvolvimento de um software é comum, por exemplo, utilizar um novo recurso por meio de experimentação. Talvez uma nova tecnologia, uma nova biblioteca que pode ser útil ao que está em desenvolvimento, ou até mesmo uma nova versão de um produto já empregado. Para que o uso deste novo recurso não interfira com o que é considerado pronto, um *branch* pode ser criado para a experimentação. Código que for criado para a experimentação existirá apenas no *branch* criado. Se eventualmente o experimento demonstrar um resultado satisfatório, as alterações realizadas no *branch* poderão ser incorporadas no que é considerado pronto, ou seja, no *branch* principal (*master*). Esta última ação é conhecida por *merge*. Neste item, apresente uma sequência de comandos que simula um caso simples de criação e uso seguido de *merge* empregando um *branch* para ilustrar uma experimentação conforme acima. A sequência deve incluir, obrigatoriamente: (a) criação de um ou mais *branches*; (b) chaveamento para pelo menos dois *branches* e (c) *merge*. Para simular alteração em um arquivo, basta simplesmente fornecer algo como **Arquivo <nome> é alterado**. O que foi fornecido em negrito representa uma ação que altera um arquivo cujo nome é fornecido entre o sinal de menor e o de maior.

```
git branch novo_branch
```

```
git checkout master
```

```
git merge novo_branch
```